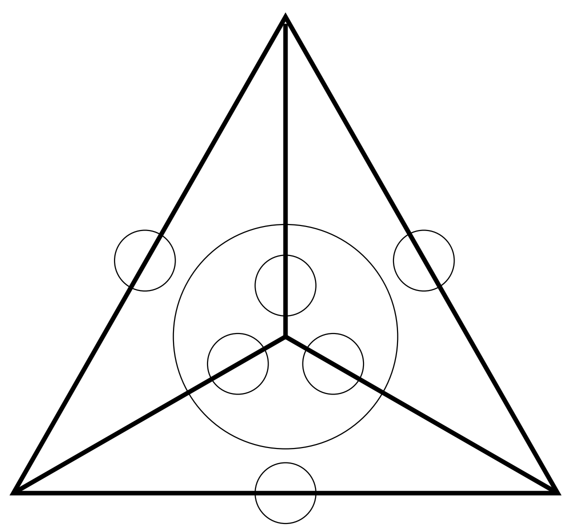
Mathematical Logic with Diagrams

Based on the Existential Graphs of Peirce

Frithjof Dau, TU Dresden, Germany



Disclaimer: This is (nearly) the final version of this treatise. There will be no more content added. It is only subject of a further proof-reading. For this reason, if you find any misspellings, gaps, flaws, etc., please contact me (`dau@dr-dau.net`). Similarly, do not hesitate to contact me if you have any questions.

Frithjof Dau, September 13, 2006

Come on, my Reader, and let us construct a diagram to illustrate the general course of thought; I mean a System of diagrammatization by means of which any course of thought can be represented with exactitude.

Peirce, Prolegomena to an Apology For Pragmaticism, 1906

Contents

Start

1 Introduction 1

1.1 The Purpose and the Structure of this Treatise 3

2 Short Introduction to Existential Graphs 7

2.1 Alpha 8

2.2 Beta 11

2.3 Gamma 14

3 Theory of Signs 17

3.1 Diagrams 18

3.2 Icons, Indices, Symbols 18

3.3 Types and Tokens, Signs and Replicas 23

4 The Role of Existential Graphs in Peirce's Philosophy 25

4.1 Foundations of Knowledge and Reasoning 26

4.2 Existential Graphs 32

4.3 Conclusion 37

5 Formalizing Diagrams 39

5.1 Problems with Existential Graphs Replicas 40

5.2 The First Approach to Diagrams 45

5.3 Linear Representations of Logic 48

5.4 The Second Approach to Diagrams 52

6 Some Remarks to the Books of Zeman, Roberts, and Shin . 55

Alpha

7 Syntax for Alpha Graphs 61

8 Semantics and Calculus for Formal Alpha Graphs 75
8.1 Semantics for Formal Alpha Graphs 75
8.2 Some Remarks to the Calculus 76
8.3 Calculus for Alpha Graphs 78
8.4 Some Simple Theorems 80

9 Soundness and Completeness 83
9.1 Soundness 83
9.2 Completeness 86

10 Translation to Propositional Logic 91

Beta

11 Getting Closer to Syntax and Semantics of Beta 97
11.1 Lines of Identities and Ligatures 98
11.2 Predicates 107
11.3 Cuts 111
11.4 Border cases: LoI touching or crossing on a Cut 117

12 Syntax for Existential Graphs 123
12.1 Relational Graphs with Cuts 123
12.2 Existential Graph Instances 126
12.3 Further Notations for Existential Graph Instances 133
12.4 Formal Existential Graphs 137

13 Semantics for Existential Graphs 141
13.1 Semantics for Existential Graph Instances 141
13.2 Semantics for Existential Graphs 145

14 Getting Closer to the Calculus for Beta 149
14.1 Erasure and Insertion 150
14.2 Iteration and Deiteration 152
14.3 Double Cuts 159
14.4 Inserting and Deleting a Heavy Dot 161

15 Calculus for Formal Existential Graphs 163

16 Improving the Handling of Ligatures 169
16.1 Derived Rules For Ligatures 169
16.2 Improving the Reading of Ligatures 178

17 Soundness of the Calculus 189

18 First Order Logic 197
18.1 Syntax 197
18.2 Semantics 201
18.3 Calculus 202

19 Syntactical Translations 205
19.1 Definition of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ 205
19.2 Semantical Equivalence between Graphs and Formulas 211

20 Syntactical Equivalence to $\mathcal{FO}$ and Completeness 217
20.1 Ψ Respects $\vdash$ 217
20.2 Identity of $\mathfrak{G}$ and $\Psi(\Phi(\mathfrak{G}))$ and Completeness of $\vdash$ 227

21 Working with Diagrams of Peirce's Graphs 229

Extending the System

22 Overview 243

23 Adding Constants and Functions 245
23.1 General Logical Background 247
23.2 Extending the Calculus 249

23.3 Examples for EGIs with Constants and Functions 253

24 Vertices with Constants 259

24.1 Syntax and Semantics 259

24.2 Correspondence between vertex-based EGIs and EGIs 262

24.3 Calculus for vertex-based EGIs 265

24.4 Ligatures in vertex-based EGIs 274

25 Relation Graphs 281

25.1 Semi Relation Graph Instances 285

25.2 Relation Graph Instances 289

26 Peirce's Reduction Thesis 297

26.1 Introduction 297

26.2 Peircean Algebraic Logic 298

26.3 Graphs for Peircean Algebraic Logic 299

26.4 Peirce's Reduction Thesis for Relation Graphs 309

26.5 The Contributions of Herzberger and Burch 314

Appendix

Referenced Authors 319

Symbols 321

Index 323

References 329

Start

1

Introduction

The research field of *diagrammatic reasoning* investigates all forms of human reasoning and argumentation wherever diagrams are involved. This research area is constituted from multiple disciplines, including cognitive science and psychology as well as computer science, artificial intelligence, logic and mathematics. But it should not be overlooked that there has been until today a long-standing prejudice against non-symbolic representation in mathematics and logic. Without doubt diagrams are often used in mathematical reasoning, but usually only as illustrations or thought aids. Diagrams, many mathematicians say, are not rigorous enough to be used in a proof, or may even mislead us in a proof. This attitude is captured by the quotation below:

> [The diagram] is only a heuristic to prompt certain trains of inference; ... it is dispensable as a proof-theoretic device; indeed ... it has no proper place in a proof as such. For the proof is a syntactic object consisting only of sentences arranged in a finite and inspectable area.
>
> Neil Tennant 1991, quotation adopted from [Bar93]

Nonetheless, there exist some diagrammatic systems which were designed for mathematical reasoning. Well-known examples are Euler circles and Venn diagrams. More important to us, at the dawn of modern logic, two diagrammatic systems had been invented in order to formalize logic. The first system is Frege's Begriffsschrift, where Frege attempted to provide a formal universal language. The other one is the systems of existential graphs (EGs) by Charles Sanders Peirce, which he used to study and describe logical argumentation. But none of these systems is used in contemporary mathematical logic. In contrast: For more than a century, linear *symbolic* representation systems (i.e., formal languages which are composed of signs which are a priori meaningless, and which are therefore manipulated by means of purely formal rules) have been the exclusive subject for formal logic. There are only a few logicians who have done research on formal, but non-symbolic logic. The most important ones are without doubt Barwise and Etchemendy. They say that

> there is no principle distinction between inference formalisms that use text and those that use diagrams. One can have rigorous, logically sound (and complete) formal systems based on diagrams.
>
> Barwise and Etchemendy 1994, quotation adopted from [Shi02a]

This treatise advocates this view that rigor formal logic can be carried out by means of manipulating diagrams. In order to do this, the systems of existential graphs is elaborated in a manner which suits the needs and rigor of contemporary mathematics.

There are good reasons for choosing Peirce's EGs for the purpose of this treatise. Peirce had been a philosopher and mathematician who devoted his life to the investigation of reasoning and the growth of knowledge. He was particularly interested in the exploration of mathematical reasoning, and EGs are designed as an instrument for the investigation of such reasoning.

Before he invented EGs at the end of the 19th century, Peirce contributed much to the development of the symbolic approach to mathematical logic.[1] Thus, Peirce was very familiar with both approaches –the diagrammatic and the symbolic– to logic. As he was interested in an *instrument for the investigation* of logic (which has to be distinguished from the investigation and development of logic as such), he discussed the differences, the advantages and disadvantages, of these two approaches to a large extent. Particularly, he elaborated a comprehensive theory of what he already called *diagrammatic reasoning*, and he considered his diagrammatic system of EGs to be far more perfect for the investigation of mathematical reasoning than the symbolic approach he developed as well. His philosophical considerations, his arguments for his preference of the diagrammatic approach to logic, will give us valuable insights to how logic with diagrams can be done.

The system of EGs is divided into three parts which are called *Alpha*, *Beta* and *Gamma*. These parts presuppose and are built upon each other, i.e. Beta builds upon Alpha, and Gamma builds upon Alpha and Beta. As EGs are an instrument for the investigation of mathematical reasoning, it is not surprising that the different parts of EGs correspond to specific fragments of mathematical logic. It is well accepted that Alpha corresponds to propositional logic, and Beta corresponds to first-order predicate logic.[2] The part Gamma

[1] For example, he invented, independently from Frege, together with his student O. H. Mitchell a notation for existential and universal quantification. According to Putnam [Put82], Frege discovered the quantifier four years before Mitchell, but 'Frege did "discover" the quantifier in the sense of having the rightful claim to priority; but Peirce and his students discovered it in the effective sense.'

[2] Later, it will be discussed in more detail how far the arguments by these authors can be understood as strict, mathematical proofs.

is more complicated: It contains features of higher order and modal logic, the possibilty to express self-reference, and other features. Due to its complexity, it was not completed by Peirce. The majority of works which deal with Gamma deal only with the fragment of Gamma which corresponds to modal logic.

The formal mathematical logic we use nowadays emerged at the beginning of the 20th century. Russell's and Whitehead's landmark work *Principia Mathematica*, probably the most influential book on modern logic, had been published in the years 1910–1912. It is obvious that Peirce's works can by no means satisfy the needs and criteria of present mathematical logic. His contributions to symbolic logic found their place in the development of modern formal logic, but his system of EGs received no attention during this process. Thus, in order to prove mathematically that Alpha and Beta correspond to propositional and first order predicate logic, respectively, the system of EGs has first to be be reworked and reformulated as a precise theory of mathematical logic. Then the correspondence to the symbolic logic we use nowadays can be mathematically formulated and proven.

Several authors like Zeman, Roberts, Sowa, Burch or Shin have explored the system of EGs. Most of them work out a correspondence of Alpha and Beta to propositional and first order predicate logic, but it will be discussed later in detail how far their arguments can be considered to be mathematical proofs. Moreover, these authors usually fail to implement EGs as a logic system on its own without a need for translations to other formal, usually symbolic logics, that is, they fail to provide a dedicated, extensional semantics for the graphs. The attempt of this treatise is to amend this gap. EGs will be developed as a formal, but diagrammatic, mathematical logic, including a well-defined syntax, an extensional semantics, and a sound and complete calculus. Translations from and to symbolic logic are provided as additional elements to work out the correspondence between diagrammatic and symbolic logic in a mathematical fashion. The methodology of developing a formal, diagrammatic logic is carried out on EGs, but it can be transferred to the development of different forms of diagrammatic logic as well.

1.1 The Purpose and the Structure of this Treatise

The overall purpose of this treatise has already been explicated: It is to develop a general framework and methodology for a diagrammatic approach to mathematical logic. In Chpt. 3, a small part of Peirce's extensively developed semiotics, i.e., theory of signs, is presented. This part is helpful to elaborate the specific differences between symbolic and diagrammatic representations of logic. Moreover, it gives us a first hint on how diagrams can be mathematically formalized. This will be more thoroughly discussed in Chpt. 5. In this chapter, the use of representations in mathematical logic is investigated, and two different, possible approaches for a formalization of diagrams are discussed and compared. From the results of this discussion, we obtain the methodology for the formalization of diagrams which is be used in this treatise.

In the frame of this general purpose, Peirce's EGs serve as a case-study. But understanding EGs as a 'mere' case-study is much too narrow. I have already argued why it is convenient not to implement an 'arbitrary' diagrammatic system, but to consider especially Peirce's EGs. Although they are not completely independent from each other, there are two main lines in the elaboration of Peirce's EGs.

First of all, this treatise aims to describe Peirce's deeper understanding of his systems of EGs (this is similar to Robert's approach in [Rob73]. See also Chpt. 6). Due to this aim, in Chpt. 4 it is discussed which role Peirce's systems of EGs in his whole philosophy has (this chapter relies on Peirce's semiotics which is described in Chpt. 3, but it is a separate chapter in the sense that the remaining treatise hardly refers to it), and Peirce's philosophical intention in the design of the syntax and the transformation rules of EGs is discussed. For Peirce's Beta graphs, in Chpt. 11, Peirce's deeper understanding on the form and meaning of his graphs is investigated, and in Chpt. 14, the same is done for Peirce's transformation rules. These four chapters offer a so-to-speak 'historical reconstruction' of Peirce's graphs. Chaps. 11 and 14 are also needed for the second goal of this treatise: To rework Peirce's graphs as a system which fulfills the standards of our contemporary mathematical logic. This is done first for Peirce's Alpha graphs, then for his Beta graphs.

The Alpha-part of EGs is mathematically elaborated in Chaps. 8–10. The syntax of these graphs is presented in Chpt. 7, the semantics and calculus is presented in Chpt. 8. In Chpt. 9, it is directly shown that the calculus is sound and complete. Propositional logic is encompassed by first order order logic; analogously, the system of Alpha graphs is encompassed by the system of Beta graphs. Thus, from a mathematical point of view, the separate elaboration of Alpha graphs is not needed. But propositional logic and first order logic are the most fundamental kinds of mathematical logic, thus, in most introductions to mathematical logic, both kinds are separately described. Moreover, this treatise aims to formalize EGs, and Peirce separated EGs into the systems Alpha, Beta and Gamma as well. For this reason, Alpha graphs are separately treated in this treatise, too. Moreover, the Alpha part can be seen as a preparation to the Beta part. Due to this reason, the formalization of Alpha graphs is geared to the formalization of Beta graphs. In fact, the formalization of Alpha graphs is somewhat a little bit too clumsy and technical. If one aims to develop solely the Alpha graphs in a mathematical manner, their formalization could be simplified, but in the light of understanding Alpha as a preparation for Beta, the herein presented formalization is more convenient. Finally, in Chpt. 10, translations between Alpha graphs and formulas of propositional logic are provided. It will be shown that these translations are meaning-preserving, thus we have indeed a correspondence between the system of Alpha graphs and propositional logic.

First order logic is much more complex than propositional logic, henceforth, the Beta part of this treatise is much more extensive than the Alpha part. Moreover, Alpha graphs, more precisely: their diagrammatic representations, and the transformation rules are somewhat easy to understand and hard to misinterpret.

Obtaining a precise understanding of the diagrams of Beta graphs, as well as a precise understanding of the transformation rules, turns out to be much harder. This is partly due to the fact that Alpha graphs are discrete structures, whereas Beta graphs (more precisely: the networks of heavily drawn lines in Beta graphs) are a priori non-discrete structures. For this reason, in Chpt. 11, the diagrams of Peirce's Beta graphs are first investigated to a large degree, before their syntax and semantics are formalized in Chaps. 12 and 13. It turns out that EGs should be formalized as *classes* of discrete structures. Then, the transformation rules for Peirce's Beta graphs are first discussed separately in Chpt. 14, before their formalization is provided in Chpt. 15. The soundness of these rules can be shown similar to the soundness of their counterparts in Alpha. This is done in Chpt. 17. Similar to Alpha, I will provide translations between Beta graphs and formulas of first order logic ($\mathcal{FO}$). In Chpt. 18, the style of $\mathcal{FO}$ which is used for this purpose is presented. In Chpt. 19, the translations between the system of Beta graphs and $\mathcal{FO}$ are provided, and it is shown that these translations are meaning-preserving. It remains to show that the calculus for Beta graphs is complete (the completeness cannot be obtained from the facts that the translations are meaning-preserving). Proving that a logic system with the expressiveness of first order logic is somewhat extensive. For this reason, in contrast to Alpha, the completeness of Beta will not be shown directly. Instead, the well-known completeness of a calculus for symbolic first order logic will be transferred to Beta graphs. In Chpt. 20, it will be shown that the translation from formulas to graphs respects the syntactical derivability-relation as well, from which the completeness of the calculus for Beta graphs is concluded. Finally in Chpt. 21, the results of the preceeding chapters are transferred to the diagrammatic representations of EGs. Thus, this chapter concludes the program of formalizing Peirce's EGs.

Peirce's EGs allow to represent propositions about relations. In Chpts. 22–26, some extensions of EGs which extend their expressiveness are investigated. First, an overview of them is provided in Chpt. 22. In Chpts. 23 and 24, the graphs are augmented with constants and functions. In Chpts. 25 and 26, a new syntactical device which correspond to free variables are added to the graphs. These resulting graphss evaluate to relations instead of propositions and are therefore termed *relation graphs*. In Chpts. 23–Chpts. 25, the syntax, semantics, and the calculus of the Beta part are appropiately extended to cover the new elements. Instead of the *logic* of the extended graphs, Chpt. 26 focuses on *operations* on relations and on how these operations are reflected by relation graphs. Then a mathematical version of Peirce's famous reduction thesis is proven for relation graphs.

The aim and the structure of this treatise should be clear now. In the remainder of this section, some unusual features of treatise are explained.

First of all, this treatise contains a few definitions, lemmata and theorems which cannot be considered to be mathematical. For example, this concerns discussions of the relationship between mathematical structures and their representations. A ‘definition’ how a mathematical structure is represented fixes a relation between these mathematical structures and their representations, but as the representations are non-mathematical entities, this definition is not a definition in a rigid mathematical sense. To distinguish strict mathematical definitions for mathematical entities and definitions where non-mathematical entities are involved, the latter will be termed *Informal Definition*. Examples can be found in Def. 5.1 or Def. 7.8.

Secondly, there are some parts of the text providing provide further discussions or expositions which are not needed for the understanding of the text, but which may be of interest for some readers. These parts can be considered to be ‘big footnotes’, but, due to their size, they are not provided as footnotes, but embedded into the continuous text. To indicate them clearly, they start with the word ‘Comment’ and are printed in footnote size. An example can be found below.

Finally, the main source of Peirce’s writings are the collected papers [HB35]. The collected papers are -as the name says- a thematically sorted collection of his writings. They consist of eight books, and in each book, the paragraphs are indexed by three-digit numbers. I adopt this index without explicitely mentioning the collected papers. For example, a passage in this treatise like ’in 4.476, Peirce writes [. . .]’ refers to [HB35], book 4, paragraph 476.

Comment: Unfortunately, the collected papers are by no means a complete collection of Peirce’s manuscripts: More than 100.000 pages, archived in the Houghton Library at Harvard, remain unpublished. Moreover, due to the attempt of the editors to provide the writings in a thematically sorted manner, they divided his manuscripts, placed some parts of them in different places of the collected papers, while other parts are dismissed. Moreover, they failed to indicate which part of the collected papers is obtained from which source, and sometimes it is even impossible to realize whether a chapter or section in the collected papers is obtained from exactly one source or it is assembled from different sources. As Mary Keeler writes in [Kee]: ‘*The misnamed Collected Papers [. . .] contains about 150 selections from his unpublished manuscripts, and only one-fifth of them are complete: parts of some manuscripts appear in up to three volumes and at least one series of papers has been scattered throughout seven.*’

2

Short Introduction to Existential Graphs

Modern formal logic is presented in a symbolic and linear fashion. That is, the signs which are used in formal logic are *symbols*, i.e. signs which are a priori meaningless and gain their meaning by conventions or interpretations (in Chpt. 3, the term 'symbol' is discussed in detail). The logical propositions, usually called *formulas* or *sentences*, are composed of symbols by writing them -like text- linearly side by side (in contrast to a spatial arrangement of signs in diagrams). In fact, nowadays *formal* logic seems to dismiss any non-symbolic approach (see the discussion at the beginning of Chpt. 5), thus *formal* logic is identified with *symbolic* logic.[1]

In contrast to the situation we have nowadays, the formal logic of the nineteenth century was not limited to symbolic logic only. At the end of that century, two important diagrammatic systems for mathematical logic have been developed. One of them is Frege's Begriffsschrift. The ideas behind the Begriffsschrift had an influence on mathematics which can hardly be underestimated, but the system itself had never been used in practice.[2] The other diagrammatic system are Peirce's existential graphs, which are the topic of this treatise. But before Peirce developed his diagrammatic form of logic, he contributed to the development of symbolic logic to a large extent. He invented the algebraic notation for predicate logic, namely the quantifiers (see for example [Rob73]) for a historical survey of Peirce's contributions to logic). Although Peirce invented the algebraic notation, he was not satisfied with this form of logic. As Roberts says in [Rob73]: '*It is true that Peirce considered algebraic formulas to be diagrams of a sort; but it is also true that these formulas, unlike other diagrams, are not 'iconic' — that is, they do not resemble the objects or relationships they represent. Peirce took this for a defect.*' Unfortunately, Peirce discovered his system of existential graphs at the very end

[1] A much more comprehensive discussion of this topic can be found in [Shi02a].

[2] The common explanation for this is that Frege's diagrams had been to complicated to be printed.

of the nineteenth century (in a manuscript of 1906, he says that he invented this system in 1896. see [Rob73]), when symbolic logic already had taken the vast precedence in formal logic. For this reason, although Peirce was convinced that EGs are a much better approach to formal logic than any symbolic notation of logic, EGs did not succeed against symbolic logic. It is somewhat ironic that existential graphs have been ruled out by symbolic formal logic, a kind of logic which was developed on the basis of Peirce's algebraic notation he introduced about 10 years before.

This treatise attempts to show that rigor formal logic can be carried out with the non-symbolic existential graphs. Before we start with the mathematical elaboration of existential graphs, in this chapter a first, informal introduction to existential graph is provided.

The system of existential graphs is a highly elegant system of logic which covers propositional logic, first order logic and even some aspects of higher-order logic and modal logic. It is divided into three parts: Alpha, Beta and Gamma.[3] These parts presuppose and are built upon each other, i.e. Beta builds upon Alpha, and Gamma builds upon Alpha and Beta. In this chapter, Alpha and Beta are introduced, but we will only take a short glance at Gamma.

2.1 Alpha

We start with the description of Alpha. The EGs of Alpha consist only of two different syntactical entities: (atomar) propositions, and so-called *cuts* (Peirce often used the term 'sep'[4] instead of 'cut', too) which are represented by fine-drawn, closed, doublepoint-free curves.[5] Atomar propositions can be considered as predicate names of arity 0. Peirce called them MEDADS.

Medads can be written down on an area (the term Peirce uses instead of 'writing' is *'scribing'*). The area where the proposition is scribed on is what Peirce called the *sheet of assertion*. It may be a sheet of paper, a blackboard, a computer screen or any other surface. Writing down a proposition is to assert it (an asserted proposition is called *judgement*). Thus,

it rains

is an EG with the meaning 'it rains', i.e. it asserts that it rains.

[3] In [Pie04], Pietarinen writes that Peirce mentions in MS 500: 2-3, 1911, that he even projected a fourth part Delta. However, Pietarinen writes that he found no further reference to it. And, to the best of my knowledge, no other authors besides Pietarinen have mentioned or even discussed Delta so far.

[4] According to Zeman [Zem64], the term 'sep' is inspired from the latin term *saepes*, which means 'fence'. Before Fig. 2.3, I provide a passage from Peirce where he writes that some cut is used to fence off a proposition from the sheet of assertion.

[5] Double-point free means that the line must not cross itself.

We can scribe several propositions onto the sheet of assertion, usually side by side (this operation is called a *juxtaposition*). This operation asserts the truth of each proposition, i.e. the juxtaposition corresponds to the conjunction of the juxtaposed propositions. For example, scribing the propositions ‘it rains’, ‘it is stormy’ and ‘it is cold‘ side by side yields the graph

it rains it is stormy it is cold

which means ‘it rains, it is stormy and it is cold’. The propositions do not have to be scribed or read off from left to right, thus

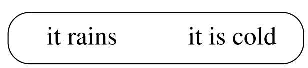
it is cold

it rains it is stormy

is another possibility to arrange the same propositions onto onto the sheet of assertion, and this graph still has the meaning ‘it rains, it is stormy and it is cold’.

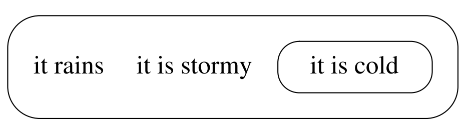
Encircling a graph by a cut is to negate it. For example, the graph

has the meaning ‘it is not true that it rains’, i.e. ‘it does not rain’. The graph

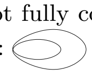
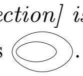


has the meaning ‘it is not true that it rains and that it is cold’, i.e. ‘it does not rain or it is not cold’ (the part ‘it is not true’ of this statement refers to the whole remainder of the statement, that is, the whole proposition ‘it rains and it is cold’ is denied.)

The space within a cut is called its *close* or *area*. Cuts may not overlap, intersect, or touch[6], but they may be nested. The next graph has two nested cuts.



Its reading starts on the sheet of assertion, then proceeding inwardly. This way of reading is called *endoporeutic method* by Peirce. Due to endoporeutic

[6] This is not fully correct: Peirce often drew scrolls with one point of intersection as follows:  . But in 4.474 he informs us that the ‘*node [the point of intersection] is of no particular significance*’, and a scroll may equally well be drawn as .

reading, this graph has the meaning ‘it is not true that it rains and it is stormy and that it is not cold’, i.e. ‘if it rains and if its stormy, then it is cold’. It has three distinct areas: The area of the sheet of assertion contains the outer cut, the area of the outer cut contains the propositions ‘it rains’ and ’it is stormy’ and the inner cut, and the inner cut contains the proposition ‘it is cold’. An area is *oddly enclosed* if it is enclosed by an odd number of cuts, and it is *evenly enclosed* if it is enclosed by an even number of cuts. Thus, the sheet of assertion is evenly enclosed, the area of the outer cut is oddly enclosed, and the area of the inner cut is evenly enclosed. Moreover, for the items on the area of a cut (or the area of the sheet of assertion), we will say that these items are *directly enclosed* by the cut. Items which are deeper nested are said to be *indirectly enclosed* by the cut. For example, the proposition ‘it is cold’ is directly enclosed by the inner cut and indirectly enclosed by the outer cut.

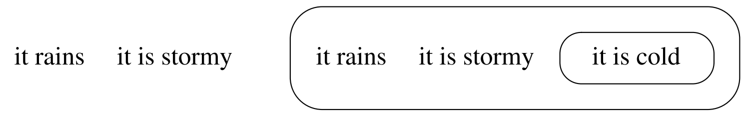
The device of two nested cuts is called a *scroll*. From the last example we learn that a scroll can be read as an implication. A scroll with nothing on its outer area is called *double cut*. Obviously, it corresponds to a double negation.

As we have the possibility to express conjunction and negation of propositions, we see that Alpha has the expressiveness of propositional logic. Peirce also provided a calculus for existential graphs (due to philosophical reasons, Peirce would object against the term ‘calculus’. This will be elaborated in Chpt. 4). This calculus has a set of five rules, which are named *erasure, insertion, iteration, deiteration, and double cut*, and only one axiom, namely the empty sheet of assertion. Each rule acts on a single graph. For Alpha, these rules can be formulated as follows:

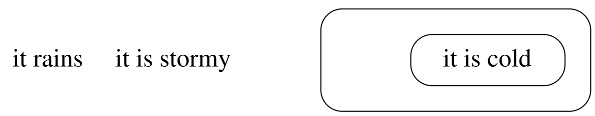
- Erasure: Any evenly enclosed subgraph[7] may be erased.
- Insertion: Any graph may be scribed on any oddly enclosed area.
- Iteration: If a subgraph $\mathfrak{G}$ occurs on the sheet of assertion or in a cut, then a copy of the graph may be scribed on the same or any nested area which does not belong to $\mathfrak{G}$.
- Deiteration: Any subgraph whose occurrence could be the result of iteration may be erased.
- Double Cut: Any double cut may be inserted around or removed from any area.

We will prove in this treatise that this set of rules is sound and complete. In the following, a simple example of a proof (which is an instantiation of modus ponens in EGs) is provided. Let us start with the following graph:

[7] The technical term ‘subgraph’ will be precisely elaborated in Chpt. 7.



It has the meaning 'it rains, and if it rains, then it is cold'. Now we see that the inner subgraph it rains it is stormy may be considered to be a copy of the outer subgraph subgraph it rains it is stormy. Hence we can erase the inner subgraph using the deiteration-rule. This yields:



This graph contains a double cut, which now may be removed. We get:

it rains it is stormy it is cold

Finally we erase the subgraph it rains it is stormy with the erasure-rule and get:

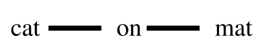
it is cold

So the graph with the meaning 'it rains and it is stormy, and if it rains and it is stormy, then it is cold' implies the graph with the meaning 'it is cold'.

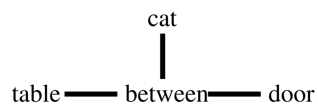
2.2 Beta

The step from the Alpha part of EGs to the Beta part corresponds to the step from propositional logic to first order logic. First of all, a new symbol, the *line of identity*, is introduced. Lines of identity are used to denote both the existence of objects and the identity between objects. They are represented as heavily drawn lines. Secondly, instead of only considering medads, i.e. predicate names of arity 0, now predicate names of arbitrary arity may be used.

Consider the following graph:

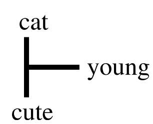


It contains two lines of identity, hence it denotes two (not necessarily different) objects. The first line of identity is attached to the unary predicate 'cat', hence the first object denotes a cat. Analogously the second line of identity denotes a mat. Both lines are attached to the dyadic predicate 'on', i.e. the first object (the cat) stands in the relation 'on' to the second object (the mat). The meaning of the graph is therefore ‘there are a cat and a mat such that the cat is on the mat’, or in short: A cat is on a mat. Analogously,



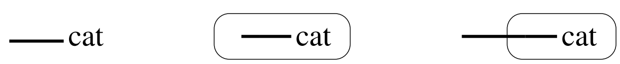
means ‘there is a cat between a table and a door’.

Lines of identity may be composed to networks. Such a network of lines of identity is called *ligature*. For example, in



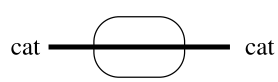
we have a ligature composed of three lines of identity, which meet in a so-called *branching point*. Still this ligature denotes one object: The meaning of the graph is ‘there is an object which is a cat, young and cute’, or ‘ there is a cute and young cat’ for short.

Ligatures may cross cuts (it will become clear in Chpt. 11 why I use the term ‘ligature’ in these examples, i.e., why I do not write that lines of identity may cross cuts). Consider the following graphs:



The meaning of the first graph is clear: it is ‘there is a cat’. The second graph is built from the first graph by drawing a cut around it, i.e. the first graph is denied. Hence the meaning of the second graph is ‘it is not true that there is a cat’, i.e. ‘there is no cat’. In the third graph, the ligature starts on the sheet of assertion. Hence the existence of the object is asserted and not denied. For this reason the meaning of the third graph is ‘there is something which is not a cat’.

A heavily drawn line which traverses a cut denotes the non-identity of the extremities of that line (again this will be discussed in Chpt. 11). For example, the graph



has the meaning ‘there is an object o_1 which is a cat, there is an object o_2 which is a cat, and o_1 and o_2 are not identical’, that is, there are at least two cats.

Now we have the possibility to express existential quantification, predicates of arbitrary arities, conjunction and negation. Hence we see that the Beta part of existential graphs corresponds to first order predicate logic (that is first order logic with identity and predicate names, but without object names and without function names).

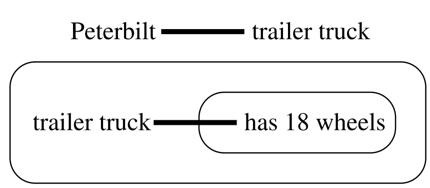
Essentially, the rules for Beta are extensions of the five rules for Alpha such that the Beta rules cover the properties of the lines of identity. The Beta rules are as follows:

- Erasure: Any evenly enclosed subgraph and any evenly enclosed portion of a line of identity may be erased.
- Insertion: Any graph may be scribed on any oddly enclosed area, and two portions of two lines of identity which are oddly enclosed on the same area may be joined.
- Iteration: For a subgraph $\mathfrak{G}$ on the sheet of assertion or in a cut, a copy of this subgraph may be scribed on the same or any nested area which does not belong to $\mathfrak{G}$. In this operation, it is allowed to connect any line of identity of $\mathfrak{G}$, which is not scribed on the area of any cut of $\mathfrak{G}$, with its iterated copy. Consequently, it is allowed to add new branches to a ligature, or to extend any line of identity inwards through cuts.
- Deiteration: Any subgraph whose occurrence could be the result of an iteration may be erased.
- Double Cut: Any double cut may be inserted around or removed from any area. This transformation is still allowed if we have ligatures which start outside the outer cut and pass through the area of the outer cut to the are of the inner cut.

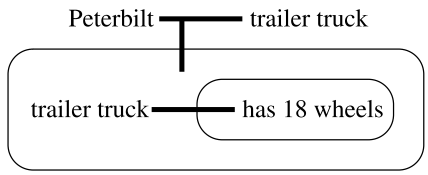
The precise understanding of these rules will be unfolded in Chpt. 14. In this chapter, they will be illustrated with an example which is taken from [Sow97a]. This example is a proof of the following syllogism of type Darii:

Every trailer truck has 18 wheels. Some Peterbilt is a trailer truck. Therefore, some Peterbilt has 18 wheels.

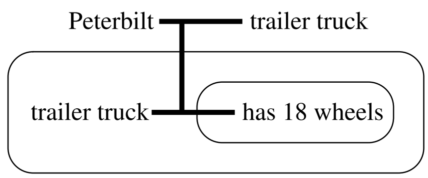
We start with an existential graph which encodes our premises:



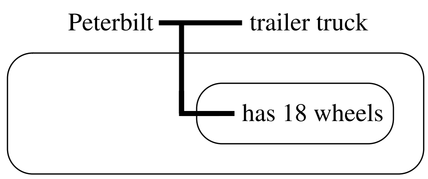
We use the rule of iteration to extend the outer line of identity into the cut:



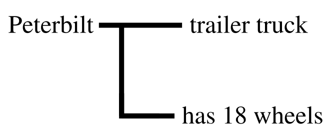
As the area of this cut is oddly enclosed, the insertion-rule allows us to join the loose end of the line of identity we have just iterated with the other line of identity:



Now we can remove the inner instance of 'is a trailer truck' with the deiteration-rule:



Next we are allowed to remove the double cut (the space between the inner and the outer cut is not empty, but what is written on this area is a ligature which entirely passes through it, thus the application of the double-cut-rule is still possible):



Finally we erase the remaining instance of 'is a trailer truck' with the erasure-rule and obtain:

This is a graph with the meaning 'some Peterbilt has 18 wheels', which is the conclusion of the syllogism.

2.3 Gamma

The Gamma part of EGs shall not be described here: I will only pick out some peculiar aspects of Gamma. The Gamma system was never completed (in 4.576, Peirce comments Gamma as follows: '*I was as yet able to gain mere glimpses, sufficient only to show me its reality, and to rouse my intense curiosity, without giving me any real insight into it.*'), and it is difficult to be sure about Peirce's intention. Roughly speaking, it encompasses higher order and modal logic and the possibilty to express self-reference. The probably best-known new device of Gamma is the so-called *broken cut*. Consider the following two graphs of 4.516 (the letter 'g' is used by Peirce to denote a graph):

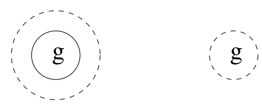
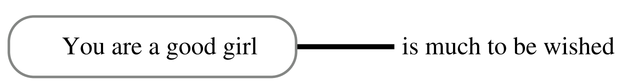


Fig. 2.1. Figs. 182 and 186 in 4.516

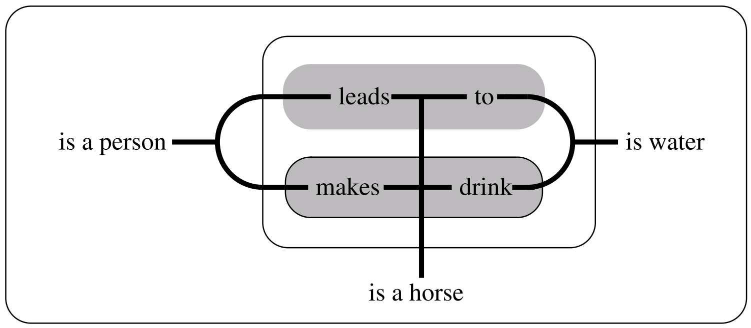
Peirce describes these graphs as follows: '*Of a certain graph g let us suppose that I am in such a state of information that it* may be true *and* may be false*; that is I can scribe on the sheet of assertion Figs. 182 and 186.*' We see that encircling a graph $\mathfrak{E}$ by a broken cuts is interpreted as 'it is possibly not the case that $\mathfrak{E}$ holds', thus, the broken cut corresponds to the syntactical device '$\Diamond\neg$' of modal logic.

Another important aspect of Gamma is the possibility to express meta-level propositions, i.e. propositions about propositions. As Peirce says: A main idea of Gamma is that a graph '*is applicable instead of merely applying it*' (quotation from [Rob73]). In other words: Graphs, which have been used to speak *about* objects so far, can now in Gamma be treated like *objects themselves* such that other graphs speak about them (this is a kind of abstraction which Peirce called 'hypostatic abstraction'). A simple example for this idea can be found in [Pei92], where Peirce says: '*When we wish to assert something about a proposition without asserting the proposition itself, we will enclose it in a lightly drawn oval, which is supposed to fence it off from the field of assertions.*' Peirce provides the following graph to illustrate his explanation:



The meaning of this graph is: *'You are a good girl' is much to be wished*. Peirce generalized the notation of a cut. The lightly drawn oval is not used to negate the enclosed graph, it is merely used to '*fence it off from the field of assertions*' and to provide a graphical possibility to speak about it.

Peirce extended this approach further. He started to use colors or tinctures to distinguish different kind of contexts. Peirce said himself: '*The nature of the* *universe or universes of discourse (for several may be referred to in a single assertion) in the rather unusual cases in which such precision is required, is denoted either by using modifications of the heraldic tinctures, marked in something like the usual manner in pale ink upon the surface, or by scribing the graphs in colored inks.*' (quotation taken from [Sow]). For example, in the next graph he used the color red to indicate possibility (which is –due to the fact that this treatise is printed in black and white– replaced by gray in the next example).



In this example we have two red (gray) ovals. One is purely red; it says that the enclosed graph is possible. The other one is a cut which is red inside, hence it says that the enclosed graph is *im*possible. As the three lines of identity start in the area of a scroll, they can be understood as *universally* quantified objects. Hence the meaning of the graph is: For all persons, horses and water, it is possible for the person to lead the horse to the water, but is impossible to make the horse drink. Or, for short: You can lead a horse to water, but you can't make him drink.

It is important to note that Peirce did not consider the tinctures to be logical operators, but to be meta-level operators. That is, they are part of a meta-language which can be used to describe how logic applies to the universe of discourse.

3

Theory of Signs

Peirce invented two types of notations for logic: An linear, algebraic notation and the spatial, graphical notation of EGs. In Peirce's writings, we find lot of passages where he emphasizes that all mathematical reasoning is based on diagrams. For example, in [Eis76], 4.314, he writes that '*all mathematical reasoning is diagrammatic*', or in his Cambridge Lectures, he says that '*all necessary reasoning is strictly speaking mathematic reasoning, that is to say, it is performed by* observing *something equivalent to a mathematical diagram.*' At a first glance, these quotations seem to be a motivation for Peirce's invention of the diagrammatic notation of EGs. But, Peirce invented a non-diagrammatic notation for logic as well. Moreover, the quotations give no hint why EGs should be preferred, or even what the main difference between the algebraic and the diagrammatic notations for logic are. Moreover, without doubt Peirce would agree that working with the algebraic notation for logic is mathematical reasoning as well, and as a mathematician he was of course very aware of the evident fact that a huge amount of mathematics is carried out with text, formulas, etc, but not with diagrams. In the light of this consideration, the quotations appear pretty puzzling, and two questions naturally arise: What qualities distinguish diagrammatic notations from non-diagrammatic notations? And what does Peirce mean when he claims that all mathematical is diagrammatic?

It is well known that Peirce developed a comprehensive (maybe the most comprehensive) and notoriously complicated theory of signs, i.e., *semiotics* (or 'semeiotics', as Peirce often called his theory). In this chapter, a few aspects of this semiotics which are essential for the elaboration of his understanding of diagrammatic reasoning will be presented. Particularly, these aspects are helpful to answer the questions above.

3.1 Diagrams

First of all, it has to stressed that Peirce's conception of the term 'diagram' is much broader than the understanding we have today. In fact, in the starting passage of this chapter I sticked to the understanding of the term 'diagram' we have nowadays, for example, when I wrote about the differences between algebraic and diagrammatic notations for logic. There seems to be a contrast or even conflict between these two approaches, but certainly, Peirce would not agree to that, as for him *all* mathematical reasoning is diagrammatic. In 1.418, he says that '*a diagram has got to be either auditory or visual, the parts being separated in the one case in time, in the other in space.*' Of course, in our understanding of diagrams, they are not auditory.[1] But even Peirce's conception of visual diagrams is much broader than ours. Any collection of spatially separated signs, i.e., any kind of visual representation system, has to be considered a diagram in Peirce's sense. Roughly speaking, nowadays, diagrammatic notations are understood to be graphical notations, thus they are distinguished from algebraic and symbolic notations like algebraic equations of formulas in mathematical logic. But for Peirce, these equations and formulas are diagrams as well. For example, in 3.419 he says that '*algebra is but a sort of diagram*', and in his Prolegomena to an Apology For Pragmaticism, he discusses in 4.530 the formula

$$\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f_0}$$

stating that '*this equation is a diagram*', and calls it later on '*algebraic diagram.*'

3.2 Icons, Indices, Symbols

Peirce classifies the signs into several trichotomies, which are often again subdivided into other trichotomies, etc.[2] For the purpose of this treatise, it is helpful and sufficient to consider one of Peirce's trichotomies, namely the division of signs into *icons, indices, and symbols*, a division of which Peirce says that this is the most fundamental (see 2.275).

In contrast to the development of existential graphs which were invented not until 1896 (Peirce was 57 years old), his investigations of signs started decades

[1] But I guess, due to the development of computers and presentation programs with multi-media features, where text, graphics, animations and sounds can be mixed, it is nowadays easier to follow Peirce's conception of diagrams.

[2] The use of trichotomies is based on Peirce's three fundamental categories called *firstness*, *secondness*, and *thirdness*. But a discussion of these categories goes far beyond the scope of this treatise.

earlier. Thus it is no wonder that we find a huge amount of definitions of the term ‘sign’ (or ‘representamen’, which is synonymous) in his writings. In [ML], 76 definitions are collected, ranging from 1865 to 1911 (and the ongoing definitions of ‘sign’ can be found in this source). Though these definitions differ in details, the overall conception of a sign does not change.

Already in 1873, in a text titled ‘Of logic as a study of signs’, MS 380, he states that ‘*a sign is something which stands for another thing to a mind.*’ 30 years later, we find a very similar definition in his Lowell Lectures (see 1.346), where he says: ‘*Now a sign is something, A, which denotes some fact or object, B, to some interpretant thought, C.*’ Thus, an entity is established as a sign when it is interpreted by a mind to represent another object. Thus we have three things to deal with: First of all, we have the sign, secondly, we have the object which is represented by the sign, and thirdly, we have a mind in which the representation causes a subjective idea of the represented object. This subjective idea is what Peirce calls ‘interpretant thought’, or, in many places, simply ‘interpretant’. This conception of signs leads to the so-called meaning triangle which is constituted by the sign, its object and the interpretant (= subjective idea). According to Peirce, the sign is ‘*mediating between an object and an interpreting thought*’ (1.480). For him, the interpreting thought is essential: 1902 he writes in MS 599 that ‘*a Sign does not function as a sign unless it be understood as a sign.*’

A sign can represent its object in three different ways, which leads to the division of signs into icons, indices, and symbols. These different types of signs are perceived in different ways, that is, the interpretant thought emerges differently. For this reason this division is interesting for the investigation of diagrams, and in fact, in Peirce’s writings on existential graphs, we find discussions on icons, indices and symbols in several places. In the following, these types will be defined and discussed. But before I do so, I have to stress that a sign rarely belongs to one of these classes exclusively. Often, a sign possesses iconical, indexical and symbolic qualities. Thus, the definitions which will be given should be understood in a prototypic manner.

A *symbol* is a sign which gains its meaning by mere convention. The best example for symbols are words. In 4.447 Peirce’s writes: ‘*Take, for example, the word "man." These three letters are not in the least like a man; nor is the sound with which they are associated*’ (we will come back to this example in the next section). There is no necessary reason whatsoever that the word ‘man’ is interpreted as it is. A symbol is a sign if ‘*is constituted a sign merely or mainly by the fact that it is used and understood as such*’, Peirce says in 2.307. The meaning of a symbol is based on laws, habits or conventions in a community. In order to understand the meaning of a symbol, we have to learn what the symbol denotes. Of course, there are other examples of symbols which are no words. For example, flags are symbols for countries, or colors are often used as symbols (for example, in Europe, the color black is a symbol for mourning, in China it is the color white).

In contrast to symbols, indices and icons have a direct connection to the represented objects. Roughly speaking, for indices it is a physical causality, for icons it is resemblance. In 2.448, Peirce writes that an '*An Index is a sign which refers to the object that it denotes by virtue of being really affected by that object*', or, more explicit in 2.299: '*The index is physically connected with its object; they make an organic pair, but the interpreting mind has nothing to do with this connection, except remarking it, after it is established.*' This induces an important difference to symbols: Symbols may refer to ideas, or abstract concepts (like 'man', which refers not to a specific man, but to the concept of man), but an index always refers to an individual, singular entity. An example Peirce gives is that smoke is an index of fire, more precisely, of *the* fire which causes the smoke. Other examples of Peirce are weathercocks which indicate the direction of the wind or low barometers which indicate moist air and coming rain; a pointing finger is an index, too (all examples can be found in 2.286). Due to the direct connection between an index and the represented object, an index does not necessarily need an interpretant to be a sign, but it need the physical existence of the denoted object. Peirce explicates this point in 2.304 with a somewhat drastic example as follows: '*An index is a sign which would, at once, lose the character which makes it a sign if its object were removed, but would not lose that character if there were no interpretant. Such, for instance, is a piece of mould with a bullet-hole in it as sign of a shot; for without the shot there would have been no hole; but there is a hole there, whether anybody has the sense to attribute it to a shot or not.*'

The most important type for diagrams is the class of *icons*. The essential feature of an icon is a similarity between the sign and the denoted object, i.e. that the '*relation between the sign and its object [...] consists in a mere resemblance between them*' (3.362). A simple example is a painting, let's say, of the Eiffel tower. A person who knows the Eiffel tower and sees the painting will immediately know that the painting refers to the Eiffel tower in Paris. No convention is needed for this, thus, the painting is not a symbol. Moreover, there is no direct, physical connection between the tower and its painting[3], thus, the painting is not an index. This becomes even more evident if we considered the painting of an object which does not exist, for example, an unicorn. Other, popular examples for icons can be found in public buildings, like signs for restrooms, busses, trains, elevators etc in an international airport.

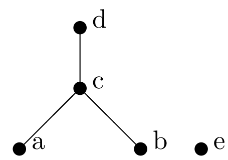
Iconic representations are much easier to comprehend than symbolic representations. Iconicity is an essential feature which occurs in diagrams. Peirce says in 4.333 that diagrams '*ought to be as iconic as possible.*' Let us consider an example in mathematics, where diagrams are used. A well-known type of diagrams are are the spatial representations of graphs, a slightly more sophis-

[3] This would be different if we considered a *photograph* of the Eiffel tower. In fact, photographs are for Peirce a mixture of indices and icons.

ticated example are Hasse-diagrams which are used to represent ordered sets.[4] Let us consider the following ordered set:

$$(\{a,b,c,d,e\},\{(a,a),(b,b),(c,c),(d,d),(e,e),(a,c),(b,c),(a,d),(b,d),(c,d)\})$$

The following is a representation of this ordered set as a Hasse-diagram:



An element of the ordered set is *above* another element, if and only if there is a path from the dot representing the first element *upwards* to the dot representing the second element. In this sense, the order $\leq$ is represented iconically. Without doubt, the order is easier to comprehend from the Hasse-diagram than from the symbolic notation given before. Moreover, in the Hasse-diagram, more facts about the represented order can immediately be read off. For example, we easily see that a, b and e are the minimal elements of the order. This advantage of iconic representation is described by Peirce in 2.279, where her writes that '*a great distinguishing property of the icon is that by the direct observation of it other truths concerning its object can be discovered than those which suffice to determine its construction.*' [5]

It is crucial to note that a diagram *has* iconic features, but usually, it *is* not an icon. Peirce writes in 2.282 that '*many diagrams resemble their objects not at all in looks; it is only in respect to the relations of their parts that their likeness consists.*' For Peirce, even algebraical equations are not purely symbolic, but they have some iconic qualities. In 2.281, he writes that '*every algebraical equation is an icon, in so far as it exhibits, by means of the algebraical signs (which are not themselves icons), the relations of the quantities concerned.*' For this reason, Peirce considered even algebraical equations to be diagrams. But without doubt, other diagrams bear much more iconicity than algebraic equations. Diagrams are representations, representing some facts, and they '*ought to be as iconic as possible.*' In 4.418 he states more precisely that '*A diagram is a representamen [sign] which is predominantly an icon of relations and is aided to be so by conventions.*' Nonetheless, there is no sharp distinction between diagrammatic and non-diagrammatic representations, but the discussion so far gives rise to a possible fixation of the term 'diagram', which, I suspect, is close to Peirce's understanding of diagrams, and which might serve as a definition which suits contemporary needs as well:

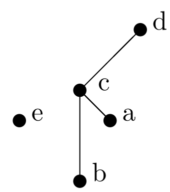
[4] A *graph* is a pair (V,E) with $E \subseteq V \times V$. The elements of V and E are called *vertices* and *edges*, respectively. An ordered set $(X,\leq)$ is a set X with a relation $\leq\,\subseteq\; X \times X$ such that $\leq$ is reflexive, anti-symmetric and transitive, i.e. for all $x,y,z \in X$ we have $x \leq x$, $\;x \leq y \leq x \Rightarrow x = y$ and $x \leq y \leq z \Rightarrow x \leq z$.

[5] In the modern research field of diagrammatic reasoning, Shimojima coined the term *free ride* for this property of diagrams.

Diagram: Given a representation of some facts, the higher the amount of iconically represented facts is (among all represented facts), the more the representation is a *diagram*.

(This is closely related to Hammer's characterization of 'good diagrams'. In Chpt. 1 of his book citeHa95, where he states that '*the resemblance between diagrams and what they represent is more a feature of good diagrams than of diagrams generally.* '.)

Concerning our example, the representation of the order $\leq$ is partly iconic, but it is partly symbolic as well. For example, it is a convention that not all pairs $(x, y) \in \leq$ are represented with a line; this is only done if x and y are neighbors, i.e., there is no third element z with $x \leq z \leq y$. A student who gets for the first time in contact with Hasse-diagrams has to learn this convention. Thus she has to learn which facts of the structure are represented in the diagram, and how this is done. Vice versa, she has to learn which properties in the diagram are indeed representing facts. It is sometimes puzzling for novices that the positions of the dots may vary to some extent, and that the length of a line is of no significance. For example, the next (badly designed) Hasse-diagram represents the same order, although it looks quite different:



The example does not only show in Hasse-diagrams are partly symbolic, moreover, it shows that it is of great importance to make clear which graphical properties in a diagram are of significance, i.e. they represent some facts, and which not. We have already seen that an advantage of diagrams is that some new facts are easily read off the diagram. But if it is not clear which graphical properties represent facts, there may be accidental, non-representing properties which seem to represent something, while they do not. Thus they may mislead a reader to draw wrong conclusion from the diagram. Shin investigates in [Shi02a] the use of icons in diagrams to a large extent, and one result of her scrutiny is the following statement: '*Now we may reach the following general conclusion about the representation of icons: When we adopt an icon to represent something in general (as a symbol does), it is important to classify the observable properties of the particular icon into two different categories; properties that are representing facts and properties that are not.[...] The existence of accidental properties is directly related to the traditional complaint against the use of diagrams in a proof.*'

3.3 Types and Tokens, Signs and Replicas

In the last section, Peirce was quoted where he exemplifies his definition of a symbol with the word "man". The whole passage is:

> Take, for example, the word "man." These three letters are not in the least like a man; nor is the sound with which they are associated. Neither is the word existentially connected with any man as an index. It cannot be so, since the word is not an existence at all. The word does not consist of three films of ink. If the word "man" occurs hundreds of times in a book of which myriads of copies are printed, all those millions of triplets of patches of ink are embodiments of one and the same word. I call each of those embodiments a replica of the symbol.

(a similar consideration with this example can be found in 2.292). This is a consideration which does only apply to symbols. For example, if we have smoke which acts as an index for a fire, then the physical object smoke *is* the sign. The sign is physically determined by the object. For symbols, the relation between object and sign is established by the interpreting mind, and the symbol is not a concrete physical object. But of course there are physical objects which represent the symbol: For the word "man", that are the three films of ink Peirce spoke about. Each of these physical occurrences are replicas of the abstract symbol "man". To put it short: '*Symbols act through replicas,*' as Peirce says. This will turn out to be important in Chpt. 5, where it is discussed how EGs can be formalized.

Although the *type-token* issue, as it is known from philosophy, is far from being settled, the distinction between graphs and graph replicas is obviously very similar to the distinction between types (graphs) and tokens (graph replicas). In fact, in his Prolegomena to an Apology For Pragmaticism, Peirce elaborates the meaning of the terms 'token' and 'type' exactly like the meaning of the term 'symbol' and 'replica'. But unfortunately, in other passages uses the term 'token' as a synonym for the term 'symbol', thus one has to be careful about Peirce use of the term 'token'.

4

The Role of Existential Graphs in Peirce's Philosophy

> Diagrammatic reasoning is the only really fertile reasoning. If logicians would only embrace this method, we should no longer see attempts to base their science on the fragile foundations of metaphysics or a psychology not based on logical theory; and there would soon be such an advance in logic that every science would feel the benefit of it.
>
> Peirce, Prolegomena to an Apology For Pragmaticism, 1906

Among philosophers, Peirce is in the first place recognized as the founder of 'pragmatism' (or 'pragmaticism', as Peirce later called his theory), and as a scientist who has elaborated the probably most extensive theory of signs, i.e., semiotics. But the system of existential graphs is neither in philosophy, nor in mathematics or logic, very much acknowledged or even appreciated. Interestingly, in contrast to the contemporary estimation of his work, Peirce himself considered his development of existential graphs as his 'luckiest find of my career', and he called them his 'chef d'oeuvre'. In a letter to William James, he says that EGs are the 'logic of future'. In fact, after he started working with EGs, he spent the remainder of his life with the elaboration of this system. Mary Keeler writes in [Kee] that '*he produces his most intensive theoretical work, which includes the Existential Graphs, during the last 10 years of his life (40.000 pages, or nearly half of the whole collection [100.000 unpublished pages which are achieved in the Houghton Library at Harvard]).*'

This chapter attempts to explain why Peirce places his existential graphs into the center of his philosophy, and from this elaboration we will moreover obtain good reasons why Peirce designed the existential graphs the way he did.

4.1 Foundations of Knowledge and Reasoning

The overall goal of Peirce's philosophy are the foundations of reasoning and knowledge. Hookway, who has worked extensively with Peirce's manuscripts, writes in [Hoo85]: '*Inspired by Kant, he devoted his live to providing foundations for knowledge and, in the course of doing so, he brought together a number of different philosophical doctrines*', and Mary Keeler says in [Kee] that '*generally, his life's work can be seen as a struggle to build the philosophical perspective needed to examine how intellectual growth occurs.*'

Peirce's semiotics and his theory of pragmaticism can be seen as two important facets of his theory of reasoning. Pragmaticism is not addressed by this chapter, thus I let other authors describe the relationship between pragmatism and reasoning. The editors of the collected papers summarize in the introduction of Volume 5 (Pragmatism and Pragmaticism) this relationship as follows: '*Pragmatism is conceived to be a method in logic rather than a principle of metaphysics. It provides a maxim which determines the admissibility of explanatory hypotheses.*' Similarly, Dipert writes in [Dip04] that '*the penultimate goal of thought is to have correct representations of the world, and these are ultimately grounded for the pragmatist in the goal of effective action in the world.*' I.e., as Dipert writes, pragmaticism answers the question *why* to think logically.

More important to us is the relationship between semiotics and reasoning. For Peirce, semiotics is not a mere metatheory of linguistics, he is interested in what sense signs are involved in reasoning. Already in 1868, in a publication titled 'Questions concerning certain Faculties Claimed for Man', he addresses the question whether reasoning which does not use signs is possible, and he comes to the conclusion that '*all thought, therefore, must necessarily be in signs*' (whole article: 5.213-5.263, quotation: 5.252[1]). Particularly, the main types of signs, i.e. icons, indices, and symbols (see [Dau04c, Shi02a] for a discussion), are needed in reasoning. In 5.243, Peirce claims that these '*three kinds of signs [...] are all indispensable in all reasoning.*' It is not only reasoning which has to be in signs. Pape summarizes the following fundamental principle which underlies Peirce understanding of semiotics:[2] '*All intellectual or sensory experience – no matter of which pre-linguistic or pre-conscious level it is – can be generalized in a way that it can be interpreted in a universal representation.*'

In his speculative grammar (2.105–2.444), Peirce's elaborates that the growth of knowledge is condensed in the change and growth of the meaning of signs.

[1] I adopt the usual convention to refer to the collected papers [HB35]. I.e., 5.213-5.263 refers to the fifth book of [HB35], paragraphs 213–263.

[2] The original German quotation is: 'Alle intellektuelle und sinnliche Erfahrung – gleich welcher vorsprachlichen oder vorbewußten Stufe – kann so verallgemeinert werden, daß sie in einer universalen Darstellung interpretierbar ist.'

In 2.222, he writes: '*For every symbol is a living thing, in a very strict sense that is no mere figure of speech. The body of the symbol changes slowly, but its meaning inevitably grows, incorporates new elements and throws off old ones.*' In this understanding, semiotics is more than a formal theory of signs: It is a theory of meaning as well. Moreover, to investigate the laws of reasoning is to investigate the relationships between the signs reasoning is based on. Thus a theory of reasoning and the emergence of knowledge has to be a theory of signs. In 1.444, Peirce summarizes the relationship between logic, reasoning and semiotic as follows: '*The term "logic" [...] in its broader sense, it is the science of the necessary laws of thought, or, still better (thought always taking place by means of signs), it is general semeiotic, treating not merely of truth, but also of the general conditions of signs being signs [...], also of the laws of the evolution of thought.*' Due to this broad understanding of semiotics and logic, these two research fields investigate reasoning from different perspectives, but they are essentially the same. So Peirce starts the second chapter of his speculative grammar with the conclusion that '*logic, in its general sense, is, as I believe I have shown, only another name for semiotic*' (2.227).

In the following, we will investigate Peirce's theory of logic and reasoning. I start this scrutiny with two quotations from Peirce, both taken from 'Book II: Existential graphs' of the collected papers, in which he elaborates his understanding of logic and so-called *necessary reasoning*. In 4.431, Peirce writes:

> But what are our assertions to be about? The answer must be that they are to be about an arbitrarily hypothetical universe, a creation of a mind. For it is necessary reasoning alone that we intend to study; and the necessity of such reasoning consists in this, that not only does the conclusion happen to be true of a pre-determinate universe, but will be true, so long as the premises are true, howsoever the universe may subsequently turn out to be determined. Physical necessity consists in the fact that whatever may happen will conform to a law of nature; and logical necessity, which is what we have here to deal with, consists of something being determinately true of a universe not entirely determinate as to what is true, and thus not existent.

In 4.477, we find:

> The purpose of logic is attained by any single passage from a premiss to a conclusion, as long as it does not at once happen that the premiss is true while the conclusion is false. But reasoning proceeds upon a rule, and an inference is not necessary, unless the rule be such that in every case the fact stated in the premiss and the fact stated in the conclusion are so related that either the premiss will be false or the conclusion will be true. (Or both, of course. "Either A or B" does not properly exclude "both A and B.") Even then, the reasoning may not

> be logical, because the rule may involve matter of fact, so that the reasoner cannot have sufficient ground to be absolutely certain that it will not sometimes fail. The inference is only logical if the reasoner can be mathematically certain of the excellence of his rule of reasoning; and in the case of necessary reasoning he must be mathematically certain that in every state of things whatsoever, whether now or a million years hence, whether here or in the farthest fixed star, such a premiss and such a conclusion will never be, the former true and the latter false.

The main point in both quotations is that Peirce's emphasizes to investigate *necessary* reasoning, and he elaborates his understanding of necessity in reasoning. First of all, we see that a necessary implication is an implication which can never lead from a true premise to a false conclusion. This can be expressed by different logical connectives: In the second quotation, he explicates a necessary inference like a truth-table (to adopt a term from contemporary propositional logic) with the operators 'not' and 'or'. In another place, he writes: '*A leading principle of inference which can lead from a true premiss to a false conclusion is insofar bad; but insofar as it can only lead either from a false premiss or to a true conclusion, it is satisfactory; and whether it leads from false to false, from true to true, or from false to true, it is equally satisfactory*'; thus in this quotation he provides the truth-table for the syntactical expression $a \to b$. A necessary implication corresponds to the material implication as it is understood in classical propositional logic, that is, as an implication which can be expressed in the following different ways:

$$a \to b \quad \sim \quad \neg(a \wedge \neg b) \quad \sim \quad \neg a \vee b$$

The truth of a necessary implication does not depend on the actual *facts* expressed in its premise and conclusion, but only on its *form*. An implication can be a 'physical necessity' if it is true due to physical laws, but here are still facts involved: Only if an implication is true in an '*arbitrarily hypothetical universe, a creation of a mind*', i.e. it is true in '*every state of things whatsoever*', then it is a necessary implication. Moreover, considering hypothetical universes fits very well into the contemporary tarski-style approach to logic and model-theory, where the different states of things, the different universes of discourses are mathematically captured by (usually relational) models, and an implication is true (a better word in mathematical logic would be 'valid') if it holds in every model.

Peirce had been a mathematician on its own, having a deep respect for mathematics and their kind of reasoning (in 4.235, he appraises the mathematicians as follows: '*Mathematicians alone reason with great subtlety and great precision.*' It is important to understand the role mathematics plays among the sciences in Peirce's philosophy. In 4.232, he explains his view what the '*true essence of mathematics*' is: '*For all modern mathematicians agree with Plato* *and Aristotle that mathematics deals exclusively with hypothetical states of things, and asserts no matter of fact whatever; and further, that it is thus alone that the necessity of its conclusions is to be explained.*' Dealing not with actual facts, but exclusively with hypothetical states of things is the essence of mathematics, not shared with any other science. In his Cambridge lectures ([Pei92]), lecture one, Peirce provides a classification of science which is based on their level of abstraction: A science is placed above a second one if the second science adopts the principles of the first science, but not vice versa. Mathematics is the science at the top of this classification '*for this irrefutable reason, that it is the only of the sciences which does not concern itself to inquiry what the actual facts are, but studies hypotheses exclusively.*' In this sense, even philosophy is more concrete than mathematics, as it is '*a search for the real truth*' and as '*it consequently draws upon experience for premises and not merely, like mathematics, for suggestions.*' As mathematics is the only science which does not deal with facts, but with hypothetical universes, it is clear why Peirce identifies necessary and mathematical reasoning. He explicates this very clearly in his lectures when he says that '*all necessary reasoning is strictly speaking mathematical reasoning.*'

In 4.425–4.429, Peirce makes clear that mathematical reasoning is by no means a mere application of some static inference rules. He starts his observations with an examination of the syllogisms of Aristotle from which he says that the '*ordinary treatises on logic [. . .] pretend that ordinary syllogism explains the reasoning of mathematics. But if this statement is examined, it will be found that it represents transformations of statements to be made that are not reduced to strict syllogistic form; and on examination it will be found that it is precisely in these transformations that the whole gist of the reasoning lies.*' When Peirce wrote these sentences, after an absolute dominance of syllogism which lasted for more than two thousand years, new approaches to a formal theory of logic and necessary reasoning emerged. Peirce, as a researcher in this field, was of course aware of these approaches. In these paragraphs, he mentions Schröder, Dedekind, and his own systems. In other places, he discusses (and extends) Boole's approach to a large extend. But none of these approaches are comprehensive enough or even sufficient to capture the whole realm of reasoning, that is '*that the soul of the reasoning has even here not been caught in the logical net*' (4.426). And even more explicit, in the beginning of the 4.425, he writes:

> But hitherto nobody has succeeded in giving a thoroughly satisfactory logical analysis of the reasoning of mathematics.[. . .] yet nobody has drawn up a complete list of such rules covering all mathematical inferences. It is true that mathematics has its calculus which solves problems by rules which are fully proved; but,[. . .] every considerable step in mathematics is performed in other ways.

We see that there is no comprehensive theory of mathematical reasoning. Moreover, Peirce is aware that mathematician are human beings which may make mistakes in their reasoning. In [Pei92], lecture 4, he writes: 'Theoretically*, I grant you, there is no possibility of error in necessary reasoning. But to speak thus 'theoretically', it is to use language in a Pickwickian sense. In practice and in fact, mathematics is not exempt from the liability to error that affects everything that man does*' (emphasis by Peirce). In the light of these observations, the question arises why Peirce had so much trust in the reliability and certainty of mathematical reasoning.

The clue is '*this marvellous self-correcting property of Reason*', as Peirce says in [Pei92]. Reasoning is a conscious process which in turn can be subject of inspection, criticism, or reasoning itself. This ability of self-criticism[3] is crucial to call any inference-process 'reasoning'; it distinguishes reasoning from a mere, mechanical application of inference rules to obtain conclusions from to premises. In 1.606 (a work titled 'ideals of conduct'), Peirce expresses this point: '*For reasoning is essentially thought that is under self-control. [...] You will nevertheless remark, without difficulty, that a person who draws a rational conclusion, not only thinks it to be true, but thinks that similar reasoning would be just in every analogous case. If he fails to think this, the inference is not to be called reasoning.*' The ability of self-control includes the ability of self-criticism: '*But in the case of reasoning an inference which self-criticism disapproves is always instantly annulled, because there is no difficulty in doing this*' (1.609).

The ability of self-criticism implies an important consequence. The conclusions of some train of reasoning are not simply granted for true: They are observed and verified. The verification of the truth of the conclusion may fail. In this case, the reasoning has to be corrected. The correction not only concerns the result of the reasoning: The assumptions the reasoning started with, even if they had been taken for true so far, may be corrected, too. In [Pei92], Peirce's writes: '*I can think of, namely, that reasoning tends to correct itself, and the more so the more wisely its plan is laid. Nay, it not only corrects its conclusions, it even corrects its premises.*'

Peirce distinguishes between three modes of reasoning. *Induction* concludes its conclusion from a sufficient large amount of facts; that is, the conclusion is an approximate proposition which generalizes and explains these facts. This is the mode of inquiry which occurs as main mode of reasoning in sciences which are based on experiments. Induction leads to truth in the long run of experience.

[3] Self-reference and self-criticism are based on a specific kind of abstraction, a shift of the observing level from the use of (linguistic) items to their observation. It should be noted that it is this shift of levels which underlies Peirce's already mentioned conception of *hypostatic abstraction*, where a former collection of items is considered to be a new, singular item of its own.

Deduction concludes its conclusion not from the content of the premises, but from the form of the argumentation. It may happen that the conclusion does not necessarily follow from the premises: It can only be concluded to a certain probability. In contrast to probable deduction, necessary deduction always leads from true premises to true conclusions. Thus, necessary deduction corresponds to necessary reasoning. It is worth to note that, according to Peirce, even deductive inquiry is based on experiments too, namely on mental experiments. Roughly speaking, induction is based on many experiments in the real world, and deduction is based on one experiment in the mind.[4]

Finally, besides induction and deduction, *abduction* is a creative generation of a new hypothesis and its provisional adoption. For a hypothesis which is obtained by abduction, its consequences are capable of experimental verification, and if further, new experiments contradict the hypothesis, it will be ruled out.

In induction and abduction, the conclusions are hypothetical, thus it is clear that these modes of reasoning tend to correct themselves. But this applies to deduction as well. Already at the beginning of lecture 4 in [Pei92], Peirce says that '*deductive inquiry, then, has its errors; and it corrects them, too*', and two pages later he concludes '*that inquiry of every type, fully carried out, has the vital power of self-correction and of growth.*' Now we see why Peirce was convinced that mathematical reasoning is such reliable: '*The certainity of mathematical reasoning, however, lies in this, that once an error is suspected, the whole world is speedily in accord about it.*'

The last quotation sheds a new light to another important aspect in Peirce's theory of reasoning and knowledge, namely the importance of a rational community (the 'whole world' [of mathematicians] in the quotation above). In Peirce's understanding, knowledge is a collective achievement. It grows by means of communication between human beings, where the results of reasoning are critically observed and discussed. In any moment, the community possesses certain information, obtained from previous experiences, whose results are analyzed by means of reasoning, i.e. deduction, induction, and abduction. Informations are conscious cognitions, and Peirce speaks of '*the cognitions which thus reach us by this infinite series of inductions and hypotheses*' (5.311). This process leads from specific information to more general information, and to the recognition of the reality and truth in the long run. In fact, there is no other way than just described to reach a knowledge of reality: In 5.312, Peirce continues: '*Thus, the very origin of the conception of reality shows that this conception essentially involves the notation of*

[4] Mathematical reasoning and diagrammatic reasoning are synonymous for Peirce. In [Eis76], we find an often quoted passage where the use of experiments in diagrammatic reasoning is explained as follows: '*By diagrammatic reasoning, I mean reasoning which constructs a diagram according to a precept expressed in general terms, performs experiments upon this diagram, notes their results, [...] and expresses this in general terms.*'

a COMMUNITY , without definite limits, and capable of a definite increase of knowledge' (emphasis by Peirce). We see that knowledge growths by use of rational communication in a community. It is worth to note that even reasoning carried out by a single person can be understood to be a special kind of rational communication as well. In 5.421, 'What Pragmaticism Is', 1905, Peirce says: '*A person is not absolutely an individual. His thoughts are what 'he is saying to himself', that is, is saying to that other self that is just coming into life in the flow of time*', or in 7.103 he explains: '*In reasoning, one is obliged to think to oneself. In order to recognize what is needful for doing this it is necessary to recognize, first of all, what "oneself" is. One is not twice in precisely the same mental state. One is virtually [...] a somewhat different person, to whom one's present thought has to be communicated.*'

4.2 Existential Graphs

The discussion so far show up some essential aspects of reasoning: It is self-controlled and self-critical, and it takes places in a community by means of rational communication. For this reason, we need an instrument which allows to explicate and investigate the course of reasoning as best as possible. This is the purpose of EGs, as it is clearly stated by Peirce in 4.248–4.429:

> Now a thorough understanding of mathematical reasoning must be a long stride toward enabling us to find a method of reasoning about this subject as well, very likely, as about other subjects that are not even recognized to be mathematical.
>
> This, then, is the purpose for which my logical algebras were designed but which, in my opinion, they do not sufficiently fulfill. The present system of existential graphs is far more perfect in that respect, and has already taught me much about mathematical reasoning. [...]
>
> Our purpose, then, is to study the workings of necessary inference.

This has already been realized by Roberts: He writes in [Rob73] that '*The aim [of EGs] was not to facilitate reasoning, but to facilitate the study of reasoning.*' In the beginning of this chapter, it has already been said that Peirce's life-long aim was the investigation of reasoning and knowledge. For him, his EGs turned out to be the right instrument for making necessary reasoning explicit (much better than language), thus the investigation of EGs *is* the investigation of necessary reasoning. From this point of view, the central place of EGs in Peirce's philosophy becomes plausible. Moreover, due to the discussion so far, the design of EGs can be explained as well.

The quotation I have just provided continues as follows: '*What we want, in order to do this, is a method of representing diagrammatically any possible set* *of premises, this diagram to be such that we can observe the transformation of these premises into the conclusion by a series of steps each of the utmost possible simplicity.*' We have already seen that deductive inquiries are for Peirce mental experiments. In these experiments, we are starting with some facts, and rearrange these facts to obtain new knowledge. First of all, different pieces of information are brought together, that is, they are *colligated*. Then, sometimes, informations are duplicated (or vice versa: redundant information is removed), or some other information which is not needed anymore is erased. These are for Peirce the general figures of reasoning: '*Precisely those three things are all that enter in the Experiment of any Deduction — Colligation, Iteration, Erasure. The rest of the process consists of observing the result.*' [Pei92]. It is this understanding of reasoning which underlies Peirce's permission rules, i.e. erasure and insertion, iteration and deiteration, and double cut. These rules are the patterns reasoning is composed of.[5]

The purpose of the rules is to explicate a reasoning process *a posteriori*, to explain and allow to make mental experiments on diagrams which explicate the premises of the reasoning process, but not to aid the drawing of inferences. In 4.373, he writes:

> The first requisite to understanding this matter is to recognize the purpose of a system of logical symbols. That purpose and end is simply and solely the investigation of the theory of logic, and not at all the construction of a calculus to aid the drawing of inferences. These two purposes are incompatible, for the reason that the system devised for the investigation of logic should be as analytical as possible, breaking up inferences into the greatest possible number of steps, and exhibiting them under the most general categories possible; while a calculus would aim, on the contrary, to reduce the number of processes as much as possible, and to specialize the symbols so as to adapt them to special kinds of inference.

Peirce has a very precise understanding of the term 'calculus' (probably based on Leibniz' idea of a 'calculus ratiocinator'). A calculus is not simply a set of (formal) rules acting on a system of symbols. For him, the purpose is essential, and the purpose gives a set of rules its shape. The purpose of a calculus is to support drawing inferences. Thus, the derivations in a calculus are rather short, and the inference steps are rather complicated, because it is the goal to reach the conclusion as fast as possible. A calculus is a *synthetical* tool. In contrast to that, the goal of Peirce's rules is to exhibit the steps of a reasoning process. Thus, the rules are rather simple and correspond the general patterns of reasoning, and the derivations '*dissect the operations of inference into as many distinct steps as possible*' (4.424). Peirce's rules are an *analytical* tool.

[5] In [Shi02a], Shin argues that Peirce's transformation rules are not fully developed in an iconic manner, and she poses the question why Peirce himself did not fully exploit the iconic features of EGs. This might be an answer to her.

They allow to discuss and critizise any reasoning best. For this reason, Peirce emphasizes that his system '*is not intended as a calculus, or apparatus by which conclusions can be reached and problems solved with greater facility than by more familiar systems of expression*' (4.424).

Comment: Ironically, compared to the rules of contemporary calculi for first order logic (like natural deduction), the rules for Peirce's EGs turn out to be rather complex. Moreover, nowadays it is often said that a main advantage of Peirce's rules is that they allow to draw very *short* inferences. An heuristic argument for this claum is given by Sowa in his commentary on Peirce's MS 514. Sowa provides an proof for Leibniz's Praeclarum Theorema (splendid theorem) with Peirce's rules, which needs 7 steps, and writes later on: '*In the Principia Mathematica, which Whitehead and Russell (1910) published 13 years after Peirce discovered his rules, the proof of the Praeclarum Theorema required a total of 43 steps, starting from five non-obvious axioms. One of those axioms was redundant, but the proof of its redundancy was not discovered by the authors or by any of their readers for another 16 years. All that work could have been saved if Whitehead and Russell had read Peirce's writings on existential graphs.*'

In [Dau06c], a mathematical proof that Peirce's rules allow fairly short proofs is provided. In this work, proofs for Statman's formulas are investigated. It is known that size of their proofs in (cut-free) sequent calculi increases exponentially. In contrast to that, in [Dau06c], it has been shown that for Statman's formulas, there exists proofs with with Peirce's rules (without using the erasure-rule) whose size increases polynomially. In Peirce's calculus, the erasure rule is the only rule which, similarly to the cut-rule in sequent calculi, does not satisfy the so-called finite choice property. That is, roughly speaking, Peirce's rules (without the erasure rule) are as nice as the rules of cut-free sequent calculi and as fast as sequent calculi including the cut-rule.

As we have just discussed the purpose and the design of the rules, we will now explore the form and appearance of EGs. I have already quoted 4.431, where Peirce states that necessary reasoning is about assertions in an '*arbitrarily hypothetical universe, a creation of a mind.*' Reasoning can be understood as a rational discourse, and such a discourse takes always place in a specific context, the *universe of discourse*. It is essential for the participants of a discourse to agree on this universe. This is explicated by by Peirce in Logical Tracts. No. 2. 'On Existential Graphs, Euler's Diagrams, and Logical Algebra', MS 492, where he writes: '*The logical universe is that object with which the utterer and the interpreter of any proposition must be well-aquainted and mutually understand each other to be well acquainted, and must understand that all their discourse refers to it.*' EGs are an instrument to make reasoning explicit. The universe of discourse is represented the system of EGs by the sheet of assertion. This function of the sheet of assertion is described in in 4.396 by Peirce as follows: '*It is agreed that a certain sheet, or blackboard, shall, under the name of The Sheet of Assertion, be considered as representing the universe of discourse [...].*'

Using a sheet of assertion for representing the universe of discourse is no accident, but a consequence of Peirce's purpose –making reasoning explicit– of EGs. This is explained by Peirce in 4.430 as follows:

> What we have to do, therefore, is to form a perfectly consistent method of expressing any assertion diagrammatically. The diagram must then evidently be something that we can see and contemplate. Now what we see appears spread out as upon a sheet. Consequently our diagram must be drawn upon a sheet. We must appropriate a sheet to the purpose, and the diagram drawn or written on the sheet is to express an assertion. We can, then, approximately call this sheet our sheet of assertion.

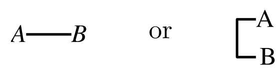
An empty sheet of assertion represents the very beginning of a discourse, when no assertions so far are made. A diagram represents a proposition, and writing the diagram on the sheet of assertion is to assert it (that is, the corresponding proposition). Peirce had a very broad understanding of the term 'diagram' (see for example [Dau04c, Shi02a]), so the question arises how the diagrams should look like. As diagrams have to be contemplated, the underlying goal is that '*a diagram ought to be as iconic as possible; that is, it should represent relations by visible relations analogous to them* ' (4.433). This goal induces some design decisions Peirce has made in the development of existential graphs.

Peirce continues in 4.433 with an example where two propositions can be taken for true, that is, each of them may be scribed on the sheet of assertion. Let us denote them by P1 and P2. Now it is a self-suggesting idea that *both* P1 *and* P2 may be written on different parts of the sheet of assertion. We then see that P1 *and* P2 are written on the sheet of assertion, and it is very natural to interpret this as the assertion of both P1 *and* P2. Writing two graphs on the sheet of assertion is called *juxtaposing* these graphs, and we have just seen that the juxtaposition of graphs is a highly iconical representation of their conjunction (to be very precisely: the conjunction of the propositions which are represented by the graphs).[6] Note that the juxtaposition of graphs is a commutative and associative operation, thus the commutativity and associativity of conjunction is iconically captured by its representation and has –in contrast to linear forms of logic– not to be covered by rules.

There are several places where Peirce discusses the iconicity of the line of identity. Assume that each of the letters A and B stands for a unary predicate, i.e. we have to complete each of them by an object to obtain a proposition which is false or true (mathematically spoken: A and B are the names of

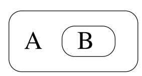
[6] Before Peirce's invention of existential graphs, he worked shortly on a system called *entitative* graphs. In this system, the juxtaposition of two graphs is interpreted as the *disjunction* of the graphs. Peirce realized that this interpretation is counter-intuitive, so he switched to to interpreting the juxtaposition as a conjunction. See [Rob73] for a thorough discussion.

relations with arity 1). Assume we know that both A and B can be completed by the *same* object in order to get a true proposition? In 4.385, Peirce answers as follows:

> A very iconoidal way of representing that there is one quasi-instant [the object] at which both A and B are true will be to connect them with a heavy line drawn in any shape, thus:
>
> 
>
> If this line be broken, thus A— —B, the identity ceases to be asserted.[7]

(A very similar argumentation can be found in 4.442.) In 4.448, he argues that a line of identity is a mixture of a symbol, an index and an icon. Nonetheless, although Peirce does not think that lines of identity are purely iconic, he concludes 4.448 with the following statement: '*The line of identity is, moreover, in the highest degree iconic.*'

In necessary reasoning, Peirce focuses on implications. In the system of existential graphs, a device of two nested cuts, a scroll, can be read as an implication. For example,



is read 'A implies B'. Reading a scroll as implication is usually obtained from the knowledge that cuts represent negation, i.e., the graph is literally read 'it is not true that A holds and B does not hold', which is equivalent to 'A implies B'. In 4.376–4.379, Peirce discusses how implications have to be handled in any logical system, and he draws the conclusion that syntactical devices are needed which allows to separate the premise from the sheet of assertion resp. the conclusion from the the premise, and he argues that this syntactical device negates the part of the implication which is separated by it. Separating a part of a graph which is written on the sheet of assertion is represented on a sheet by a closed line. For this reason, he added the cut as a syntactical devise to graphs, and due to the argument he has provided before, he *concludes* that the cut negates the enclosed subgraph.

We see that the design, the appearance of EGs is, similarly to the design of the rules, driven by Peirce's purpose to provide an instrument for investigating reasoning.

[7] The two lines of identity *may* denote distinct objects, but this is not *necessary*, i.e., they are still allowed to denote the same objects as well.

4.3 Conclusion

To conclude this chapter, it shall be remarked that Peirce himself stresses in 4.424 that his purpose for EGs has not to be confused with other purposes. We have already seen that EGs are not intended as a calculus. Moreover, Peirce stresses that '*this system is not intended to serve as a universal language for mathematicians or other reasoners.*' A universal language is intended to describe only one, i.e., 'the', universe. In a universal language, the signs have a fixed, definite meaning. i.e. there are no different interpretations of a sign.[8] But EGs are about arbitrarily hypothetical universes, and they have to be interpreted in a given universe of discourse (Peirce describes the handling of EGs by means of a communication between a so-called *graphist*, who asserts facts by scribing and manipulating appropriate graphs on the sheet of assertion, and a so-called *grapheus* or *interpreter*[9], who interprets the graphs scribed by the grapheus and checks their validity in the universe of discourse). Thus it is clear that EGs cannot serve as a universal language.

Moreover, although EGs are intended to provide an instrument for the investigation of reasoning, it is important for Peirce that the psychological aspects of reasoning are not taken into account.[10] Finally, Peirce writes that '*although there is a certain fascination about these graphs, and the way they work is pretty enough, yet the system is not intended for a plaything, as logical algebra has sometimes been made.*' After we have elaborated in this chapter why Peirce placed his EGs into the very center of his philosophy, this assessment is by no means surprising.

---

[8] This was Frege's intention when he invented his Begriffsschrift.

[9] Due to Peirce's understanding that reasoning is established by communication, the graphist and grapheus may be different mental states of the same person.

[10] Susan Haack distinguishes in her book 'philosophy of logics' ([Haa78]) three approaches to logic: strong psychologism, where '*logic is descriptive of mental processes*', weak psychologism, where '*logic is prescriptive of mental processes (it prescribes how we* should *think*', and anti-psychologism, where '*logic has nothing to do with mental processes*', and names Kant Peirce and Frege as examples for these three approaches, respectively.

5

Formalizing Diagrams

There are some authors who explored EGs, e.g. Roberts, Shin, Sowa, or Zeman. In all elaborations of EGs, it is claimed that the Alpha part corresponds to propositional calculus, and the Beta part corresponds to first order predicate logic, that is, first order logic with predicates and identity, but without object names and function names. Propositional calculus and first order predicate logic are comprehensively elaborated mathematical theories, thus this claim has to be understood in a mathematical manner. In other words: The system of EGs has to be formalized in a fashion which suits the needs and rigor of contemporary mathematics, and the correspondence between EGs and mathematical logic has to be elaborated and proven mathematically.

Peirce distinguished between graphs and their graphical representations, termed *graph-replicas*. Not all authors employ this distinction in their elaborations, i.e., they treat graphs like they *are* the graphical representations. This approach leads to difficulties in the formalization of EGs. In Sect. 5.1, some of the problems which occur in this handling of EGs are analyzed. In order to do this, the approach of Shin (see [Shi02a]) will be used. Besides the missing distinction between EGs and their graphical representations, it will turn out that Shin's definition of EGs are, from a mathematical point of view, informal, which leads to problems as well. In fact, due to this problems, Shin's (and some authors as well) elaboration of EGs is from a mathematicians point of view insufficient and cannot serve as diagrammatic approach to mathematical logic. I will argue that a separation between EGs as abstract, mathematical structures and their graphical representations is appropriate. A question which then arises immediately is how the graphical representations, i.e. the graph replicas, should be defined and handled.

One might suspect that the graph replicas should be mathematically defined as well, but in Secs. 5.2 it will be shown that a mathematical formalization of the graphical representations may cause a new class of problems. In Sect. 5.3 we will discuss why the symbolic approach to mathematical logic does not have to cope with problems which arise in diagrammatic systems. It will turn out that the preciseness of mathematical logic is possible although the separation between formulas and their representations is usually not discussed. From this result we draw the conclusion that a mathematical formalization of the diagrammatic representations of EGs is *not* needed: It is sufficient to provide conventions how the abstract structures are represented. This result and the outcomes of the discussion in the preceding sections are brought together in Sect. 5.4 in order to describe a second approach for a mathematical foundation of diagrams, which will be used in this treatise.

5.1 Problems with Existential Graphs Replicas

To illustrate some of the problems which may occur in the handling of diagrams, we focus on the EGs as they are described in the book of Shin ([Shi02a]). We start the discussion with Shin's definitions for alpha and beta graphs.

Informal Definition 5.1 (Alpha Graphs) *The set of alpha graphs, $\mathcal{G}_\alpha$, is the smallest set satisfying the following:*

1. *An empty space is in $\mathcal{G}_\alpha$.*
2. *An* sentence symbol *is in $\mathcal{G}_\alpha$.*
3. Juxtaposition closure *If G_1 is in $\mathcal{G}_\alpha$, ..., and G_n is in $\mathcal{G}_\alpha$, then the juxtaposition of these n graphs, i.e. $G_1, \ldots, G_n$, (we write '$G_1 \ldots G_n$' for juxtaposition) is also in $\mathcal{G}_\alpha$.*
4. Cut closure *If G is $\mathcal{G}_\alpha$, then* a single cut of G *(we write '$[G]$' following Peirce's linear notation) is also in $\mathcal{G}_\alpha$.*

Informal Definition 5.2 (Beta Graphs) *The set of beta graphs, $\mathcal{G}_\beta$, is the smallest set satisfying the following:*

1. *An empty space is in $\mathcal{G}_\beta$.*
2. *A* line of identity *is in $\mathcal{G}_\beta$.*
3. Juxtaposition closure *If G_1 is in $\mathcal{G}_\beta$, ..., and G_n is in $\mathcal{G}_\beta$, then the juxtaposition of these n graphs, i.e. $G_1, \ldots, G_n$, (we write '$G_1 \ldots G_n$' for juxtaposition) is also in $\mathcal{G}_\beta$.*
4. Predicate closure *If G is in $\mathcal{G}_\beta$, then a graph with an n-ary* predicate *symbol written at the joint of n loose ends in G is also in $\mathcal{G}_\beta$.*
5. Cut closure *If G is in $\mathcal{G}_\beta$, then a graph in which a* single cut *is drawn in any subpart of G without crossing a predicate symbol is also in $\mathcal{G}_\beta$.*

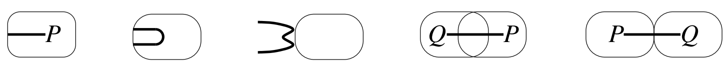
6. Branch closure *If* G *is in* $\mathcal{G}_\beta$*, then a graph in which a line of identity in* G branches *is also in* $\mathcal{G}_\beta$*.*

There are two points remarkable in this definition:

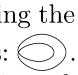
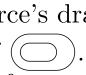
1. Although some mathematical terms are used, the definitions are formulated more or less in common spoken language and cannot be seen as *mathematical* definitions.
2. EGs are considered to be *graphical* entities (this can particularly seen in the cut closure rule for Beta graphs).[1]

This approach, particularly the two points mentioned above, yields different kinds of problems which shall be elaborated in this section. We start with problems caused by the use of common language.

First of all, many important technical terms are defined either in an insufficient way or not at all. For example, the terms *sentence symbol*, *juxtaposition* or *single cut* (this holds already for Shin's definition of Alpha graphs) as well as the terms *lines of identity*, *loose ends* or *branching of LoIs* are not defined. Even if we have some pre-knowledge on terms like 'single cut' (e.g. we know that a single cut is a closed, double-point-free curve on the plane) or LoIs, these definitions leave some issues open. E.g., is it not clear whether two cuts may touch, cross, intersect, or partly overlap, or whether a LoI may terminate on a cut. For example, we might ask which of the following diagrams are well-defined diagrams of Beta graphs:



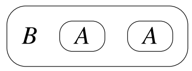
Examining the first three examples, in Shin's book we do not find any example of an Beta graph where a LoI terminates on a cut. But, in contrast, this case is explicitly investigated in Peirce's manuscripts or in the book of Roberts. The fourth example is not explicitly excluded by Shin, but it is implicitly excluded based on the background a reader should have of EGs.

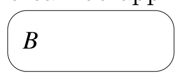
Considering the fifth example, Peirce used to draw to nested cuts, i.e., scrolls, as follows: . So it seems that a touching of cuts is allowed. But we can raise the question whether scrolls should be handled as own syntactical devices, or whether Peirce's drawing is a sloppy version of two nested cuts which do not touch, i.e. of . So we have to make a *decision* whether cuts may touch or not (see the footnote on page 9). Answering these question is not only about deciding whether a diagram is an EG or not. It is even more important to have rigor definitions for all technical terms when transformation rules, e.g.

[1] In [HMST02], she revised her approach to diagrammatic reasoning and stresses that a distinction between abstract structures and graphical representations is needed.

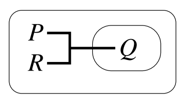
the rules of a calculus, are applied to EGs. An important example for this is the in most publications undefined term *subgraph*. A rule in the calculus allows us to scribe a copy of a subgraph in the same cut (this is a special case of the *iteration-rule*), which occurs in the treatises of Peirce, Shin, Roberts, and later on in this treatise.

To get an impression of the problems, we raise some simple questions on subgraphs. Consider the following Alpha graph:

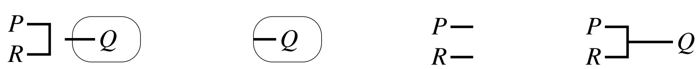
$\mathfrak{G}_\alpha :=$ 

In order to know how the iteration-rule can be applied, it must be possible to answer the following questions: Is  a subgraph of $\mathfrak{G}_\alpha$? Is $A \quad A$ a subgraph of $\mathfrak{G}_\alpha$? Do we have one or two subgraphs  of $\mathfrak{G}_\alpha$ (this is a question which corresponds to the distinction between subformulas and subformula instances in $\mathcal{FO}$)?

For Beta, it is important to know how LoIs are handled in subgraphs. Consider the following Beta graph:

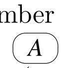
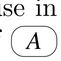
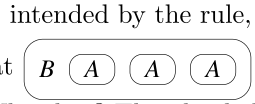
$\mathfrak{G}_\beta :=$ 

We might ask which of the following diagrams are subgraphs of $\mathfrak{G}_\beta$:



Shin (and most other authors) does not offer a definition for subgraphs: The answer of the questions above is left to the intuition of the reader. From the discussion above, we draw a first conclusion:

> **Thesis 1:** The definitions of Shin (and some other authors) are insufficient for a precise understanding and handling of existential graphs.

If EGs are considered as *graphical* entities, i.e., if no distinction between graphs and graph-replicas in employed, a new class of difficulties has to be coped with. Consider again $\mathfrak{G}_\alpha$. Remember that the iteration-rule should allow us to draw a copy of the subgraph  into the outer cut. But if we want to understand EGs and subgraphs as (graphical) diagrams, then this is obviously *not* possible, because in the outer cut, there is simply not enough space left for another copy of . But, of course, this is not intended by the rule, and everybody who is familiar with EGs will agree that  is a result of a valid application of the iteration-rule. Why that? The idea behind is that we may change the shape of LoIs or cuts to a 'certain degree' without changing the meaning of an EG. For this reason it is evident that any attempt which tries to define EGs as purely graphical entities runs into problems.

For a discussion of the term 'certain degree', consider the three Alpha graphs in Fig. 5.1. From the left to the right, we decrease the size of the outer cut.

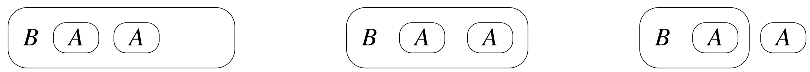
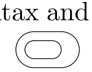


Fig. 5.1. Diagrams of Alpha graphs

There are obviously visual differences between these three diagrams. The question is whether the differences between the first two diagrams are comparable to the differences between the last two diagrams. We have already seen that the shape of a cut is – in some sense – of no relevance. The only facts we have to know is which other items of the graph are enclosed by the cut and which are not. Thus we see that the first two diagrams are (in some sense) the same graph, particularly they have the same meaning. In contrast to that the third diagram has a different meaning and has therefore to be treated differently.

If we – due to the visual differences – treat the first two diagrams to be syntactically different, we would get a syntax which is much too fine-grained. Any kind of equivalence between graphs would be postponed to the semantical level. Furthermore, we would need transformation rules which allow to transform the first graph into the second graph (and vice versa). This syntax would become very complicated and nearly unusable. Thus we see that any appropriate syntax should not distinguish between the first two diagrams.

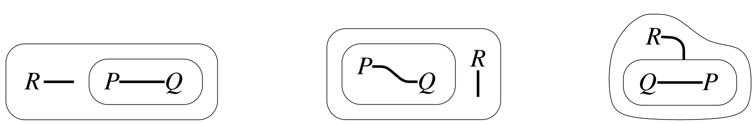
Now the question arises which properties of the first two diagrams cause us to identify them. Should we syntactically identify graphs when they have the same meaning? This would inappropriately mix up syntax and semantics. For example, the empty sheet of assertion and the graph  have the same meaning, but they should obviously be syntactically distinguished.

In defining a reasonable syntax for the graphs, we see that we have to dismiss certain graphical properties of the diagrams (e.g. the *form* of a cut), while other properties are important (e.g. the number of cuts and other items, or whether an item of the diagram is enclosed by a cut or not). Particularly, in any formalization, EGs should not be understood as graphical entities at all. Instead of this, we have, according to Peirce's distinction between graph and graph-replicas, to distinguish between graphs as abstract structures and the diagrams which *represent* the graphs.

When we employ a distinction between EGs and their replicas, we have to investigate which graphical properties in a diagram represent facts of the represented structure, and which not (this has already been discussed at the end of Sect. 3.2). In other words: When are two diagrams replicas of the same EG? Peirce explicitly said that arbitrary features of the diagrams may vary, as long as they represent the same EG. At the beginning of [Pei35] he says:

> Convention No. Zero. Any feature of these diagrams that is not expressly or by previous conventions of languages required by the conventions to have a given character may be varied at will. This "convention" is numbered zero, because it is understood in all agreements.

A similar explication is his '*Rule of Deformation*' in 4.507, where he says that '*all parts of the graph may be deformed in any way, the connexions of parts remaining unaltered; and the extension of a line of identity outside a sep*[2] *to an otherwise vacant point on that sep is not to be considered to be a connexion.*'. For LoIs, he says even more explicit in [Pei35] that '*its shape and length are matters of indifference*', and finally, in 4.500 we find that '*Lines of Identity are replicas of the linear graph of identity*'. Particularly the last quotation is illuminating, as it makes very clear that a deformation of a line of identity is indeed only a change in the representation, but not a change in the represented graph. From this we can conclude that the other mentioned deformations (for example, changing the shape of a cut) are mere changes in the representation of a fixed, represented EG. Due to this understanding, the first two diagrams in Fig. 5.1 are not different graphs with the same meaning, but different representations, i.e., diagrams, of the same graph, and the third diagram is a representation of a different graph. Similarly, the following diagrams are all replicas of the same beta graph.



Please note that the line of identity in the right diagram terminates on a cut. Even in this case, the diagram is still a replica of the same EG, as '*the extension of a line of identity outside a sep to an otherwise vacant point on that sep is not to be considered to be a connexion*'. But those diagrams can easily be transformed such that LoIs terminating on a cut do not occur, while they still represent the same EG. We will come back to LoI terminating on a cut in Chpt. 11.

The important distinction between graphs graph replicas is summarized in the next conclusion:

> **Thesis 2:** EGs should not be defined as graphical entities. Instead of that, we need a definition of EGs which copes exactly the crucial

[2] Recall that 'sep' is another word Peirce used for cut.

features of EGs, and the diagrams should be understood as (mere) representations of an underlying EG.

As the distinction between graphs and graph-replicas corresponds to the distinction between types and tokens, roughly sketched, we now have the following situation:

$$\begin{tikzcd}
\text{Type} \arrow[r, leftrightarrow, "\text{corresponds to}"] & \text{Ex. Graph} \arrow[r, leftrightarrow, "\text{corresponds to}"] & \text{Math. Structure} \\
\underbrace{\text{Token}}_{\text{Semiotics}} \arrow[u, "\text{represents}"'] \arrow[r, leftrightarrow, "\text{corresponds to}"] & \underbrace{\text{Ex. Graph Replica}}_{\text{Peirce}} \arrow[u, "\text{represents}"'] \arrow[r, leftrightarrow, "\text{corresponds to}"] & \underbrace{\text{Diag. Represent.}}_{\text{This Treatise}} \arrow[u, "\text{represents}"']
\end{tikzcd}$$

5.2 The First Approach to Diagrams

One of the most important features of mathematics is its preciseness. The preciseness of mathematics is based on a very strict understanding of mathematical definitions and proofs. We have seen that the informal definitions of EGs lead to problems in their understanding and handling. Moreover, the claim that the Alpha and Beta part of EGs correspond to propositional calculus first order predicate logic, resp., lead to the conclusion that EGs should be defined as mathematical structures, and the diagrams are representations for these structures. This approach raises the question whether the graph replicas, i.e., the diagrams of EGs, should be defined mathematically as well. This approach shall be discussed in this section.

Let us assume in this section that we want to define the graphical representations of Alpha or Beta graphs mathematically. Then we would have two different kinds of objects: Mathematical structures which model EGs, and mathematical structures which model the representations of EGs, i.e., the diagrams. Let us call the first structures *type-structures* and the second structures *token-structures*. In finding a definition for the token-structures, we have two fundamental issues to cope with: First to find a definition for the token-structures which encodes the informally given diagrams as best as possible. Secondly, we have to show how the type-structures are represented by the token-structures. It should be possible to show that each token-structure represents uniquely a type-structure, and that each type-structure is represented by at least one token-structure. Let us call these two principal problems *representation problem*.

An obvious approach is to model the lines of an EG (i.e., the cuts and the LoIs) as families of curves in the Euclidean plane $\mathbb{R}^2$. For example, we can model each LoI by a smooth, double-point-free curve and each cut by a by a smooth, double-point-free and closed curve.[3] Consider the fist two graphs of Fig. 5.1. They are two different tokens of the same type. Diagrams like these shall be called *type-equivalent* (this term is adopted from [HMST02]).

If we define type-equivalence, we have to refer to the relationship between types and tokens. But the diagrams can be compared directly as well. If we consider again the first two graphs of Figure 5.1, we see that we have mappings from the cuts resp. the occurrences of propositional variables from the first graph to the second which fulfill certain conditions. For example, the mappings are bijective and the preserve some entailment-relations (e.g. if an occurrence of a propositional variable or a cut is enclosed by a cut, then this holds for the images as well). In some sense, we can say that the first graph can be topologically transformed into the second graph. Graphs like this shall be called *diagrammatically equivalent* (again, this term is adopted from [HMST02]). If we have found adequate definitions for the type- and token-structures as well as for the relation 'a token-structure represents a type-structure', it should be mathematically *provable* that being type-equivalent and being diagrammatically equivalent means the same.

In any description of the diagrams, particularly if we provide a mathematical definition for them, we have to decide some of the border cases we have discussed in Sect. 5.1. For example, we have to decide whether (and how often) LoIs may touch cuts, or whether cuts may touch or even intersect each other. But in no (reasonable) mathematical definition, we can encode *all* graphical properties of a diagram directly (e.g. by curves). This is easiest to see for letters or words, i.e. the occurrences of propositional variables in Alpha graphs or for the relation names in Beta graphs. Of course the *location* of the occurrence of a propositional variable in an Alpha graph is important, but neither the size or the font of the propositional variable will be of relevance. Similar considerations hold for relation names in Beta graphs. As the shape of propositional variables or relation names should not be captured by the definition of the diagrams, it is reasonable to handle these items in a different way. The question arises how much of the properties of these items has to captured by a definition of the diagrams. For example, the occurrences of propositional variables in an Alpha graph could be modeled by points or spots of the Euclidean plane to which we assign the variables. We see that even in a mathematical definition for the diagrams, we have to prescind certain graphical features.

On the other hand: If we try to capture graphical properties of diagrams, some of the items of a diagram will probably be overspecified. For example,

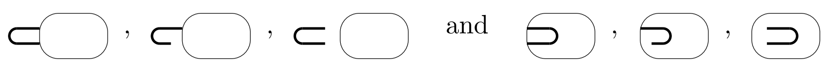
[3] It should be noted that Peirce's understanding of EGs depends on his understanding of the continuum, and this understanding is very different from the set $\mathbb{R}$. Nevertheless a mathematization of the diagrams as a structure of lines and curves in $\mathbb{R}^2$ is convenient as $\mathbb{R}^2$ is the standard mathematization of the Euclidean plane in contemporary mathematics.

how should a simple line of identity – i.e., ——— – be modeled mathematically? We already said that LoI could be modeled by a curve in the Euclidean plane. But when a diagram is given, it is usually drawn without providing a coordinate system. Thus, which length should this curve have? Where in the Euclidean plane is it located? Again we see that we cannot capture a diagram exactly by a mathematical definition.

Finally, it is worth to note that we have a couple of (unconscious) heuristics in drawing diagrams for EGs. A simple example is that pending edges should be drawn 'far away enough' from any relation-signs. To see this, consider the following diagrams:

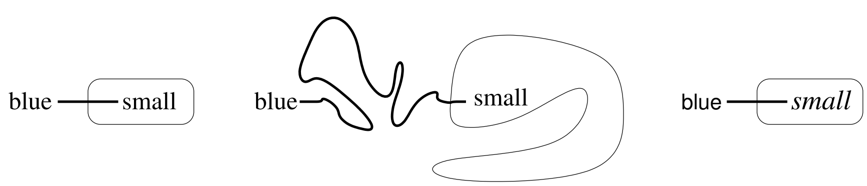
The leftmost diagram should be understood as 'There is thing which is P which is in relation Q to another thing', The rightmost diagram should be understood as 'There is thing which is P and there is thing which is Q'. But the readings, thus the meanings, of the diagrams in the middles are not clear. Thus one will avoid drawing diagrams like these.

Similar considerations have to be taken into account if we think about how the diagrams are handled further. For example, we have to be careful in the definition of the relation 'being diagrammatically equivalent'. To see this, consider the following simple example which consists of two triples of diagrams:



Peirce treated point on a cut to be placed *outside* a cut. Thus the first three diagrams are in Peirce's understanding different representations of the same graph, while the last three diagrams are pairwise not type-equivalent. Hence an appropriate definition for EGs should yield that the first three graphs are diagrammatically equivalent, but the last three graphs are not.

Another heuristic is to draw a diagram as simple as possible. Furthermore, although we have seen that neither the size or the font of the propositional variable or a relation name will be of relevance, it is clear that the choice of a font and its size is not arbitrary if a diagram is drawn in a convenient way. This should became clear with the following three diagrams:



Although all diagrams are representations of the same graph (with the meaning 'there exists something which is blue, but not small), it is clear that the left-most diagram is the best representation of these three diagrams. Conventions like these cannot by captured by any mathematical definition.

Moreover, we have to think how any operations with graphs (i.e., with the type-structures) are reflected by their representations. Examples for operations are the juxtaposition of graphs or all transformation rules of a calculus. Shall these operations be defined on the type- or on the token-level? The discussion in Sect. 5.1 (see the discussion of Fig. 5.1) shows that a definition on the type-level is more convenient. Nonetheless, then the question remains how a operation on the type-level is represented on the token-level. For the herein investigated system of existential graphs, such a scrutiny is carried out in Chpt. 21.

Remember that the reason for finding mathematical definitions for diagrams was to solve the representation problem. We wanted to grasp the distinction between non well-formed and well-formed diagrams, as well as the relationship between graphs and their diagrams, as best as possible. We have already seen that we cannot capture all graphical features of a diagram by a mathematical definition (the more graphical properties are encompassed by a definition, the more technical overhead has to be expected). Now the final question is how a mathematically defined diagram is related to a concrete drawing of a diagram on a sheet of paper. This is a crucial last step from mathematical objects to objects of the real world. Thus, even if we provide mathematical definitions for the diagrams, we still have a representation problem. The initial representation problem between mathematically defined graphs and mathematically defined diagrams has shifted to a representation problem between mathematically defined diagrams and diagrams – i.e., drawings – in the real world. A mathematical definition for diagrams can clarify a lot of ambiguities, but it cannot solve the representation problem finally.

5.3 Linear Representations of Logic

In the last section we have raised some questions concerning the representation problem, and we have seen that mathematics alone is not enough to solve these problems. It is likely the often unclear relationship between diagrams and represented structures which causes mathematicians to believe that diagrams cannot have a proper place in mathematical argumentation, esp. proofs. It is argued that only a symbolic system for logic can provide the preciseness which is needed in mathematical proofs (for a broader discussion of this see [Shi02a]). It seems that the problems we have discussed in the last section simply do not occur in mathematics, esp. mathematical logic. In this section we will have a closer look on this. We start with a definition of the well-formed formulas of $\mathcal{FO}$.[4]

Definition 5.3. *The alphabet for first order logic consists of the following signs:*

- *Variables:* $x_1, x_2, x_3, \ldots$ *(countably many)*
- *Relation symbols:* $R_1, R_2, R_3, \ldots$ *(countably many). To each predicate symbol* R_i *we assign an* ARITY $ar(R_i) \in \mathbb{N}$.[5]
- *Connectives:* $\wedge, \neg, \exists$
- *Auxiliary Symbols:* ., ,, (,).

Definition 5.4. *The formulas of* $\mathcal{FO}$ *are inductively defined as follows:*

1. *Each variable and each constant name is a* TERM.
2. *If* R *is a predicate symbol with arity* n *and if* $t_1, \ldots, t_n$ *are terms, then* $f := R(t_1, \ldots, t_n)$ *is a formula.*
3. *If* f' *is a formula, then* $f := \neg f'$ *is a formula.*
4. *If* f_1 *and* f_2 *are formulas, then* $f := (f_1 \wedge f_2)$ *is a formula.*
5. *If* f' *is a formula and* α *is a variable, then* $:= \exists\alpha.f'$ *is a formula.*

It is easy to capture the idea behind this definitions: First, we fix a set of *signs*, and a formula is a *sequence* of these signs which has been composed according to certain *rules*.

Let us consider the following two strings (the relation name R_1 has arity 2):

$$\exists x_1.\exists x_2.R_1(x_1, x_2)$$
$$\exists x_1.\ \exists x_2.\ R_1(x_1, x_2)$$

Although these two strings are written in different places and although they look slightly different, they clearly represent the same formula. For our considerations, it is worth to raise the question which steps a reader has to pass from the perception of a string to the identification of a formula which is represented by the string. Roughly speaking, if a reader reads the two strings above, she passes the following steps:

1. The region on the paper must be identified where the representation of the formula is written on. In the example above, this is possible because we have written the two different strings into two different lines.

---

[4] The following definitions are only needed to discuss the type-token-issue for symbolic logic. In Chpt. 18, a more thorough introduction into the notations of symbolic first order logic is provided.

[5] We set $\mathbb{N} := \{1, 2, 3, \ldots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

2. In this region, we must be able to identify representations of the *signs* which may occur in a formula (e.g. the sign '$\exists$', which appears twice, or the sign 'R_1', which appears only once). Nothing else may occur.
3. The representations of the signs must be assembled in a way such that we are able to identify their ordering on the region. That is: We must be able to identify the *sequence*. For our examples, we identify the ordering of the instances of signs by reading them from left to right.
4. Finally, after we have reconstructed the sequence of signs (internally), we can check whether this sequence is a well-defined formula, i.e., whether it is composed with the rules of Definition 5.4.

In the following, we will use the label $(*)$ to refer to these four steps. The process $(*)$ yields the same result for the two strings above: In both cases, the lines represent the same sequence of signs, which is in fact a well-formed formula. Thus every mathematician would (hopefully) agree that these two strings represent the same (well-defined) formula.

We want to stress that the process of perceiving a representation of a formula is not 'deterministic': It is not clear without ambiguity for each string whether it represents a formula or not. To see this, consider the following strings (the 'type-setting problems' are intended). The question is: Which of these strings represents a well-defined formula?

$$\rlap{$\exists$}x_1\!\!\rlap{.}\exists x_2\!\rlap{.}R_1\!\!\rlap{(}x_1\!\!,x_2) \tag{5.1}$$

$$\begin{array}{llll} & .^{x_2} & R_1(& \;\;\;\;) \\ \exists & \;\;\exists\;\;\cdot & & \\ x_1 & \;\;\;\;\;\vdots & x_1{}^{,}x_2 & \end{array} \tag{5.2}$$

$$\exists x_1 . \exists \heartsuit . R_1(x_1, x_2) \tag{5.3}$$

$$), \exists .. R_1 x_1 \exists x_1 (x_2 x_2 \tag{5.4}$$

$$\exists \mathrm{x}_1 \, . \exists \mathrm{x}_2 \, . R_1(\mathrm{x}_1, \mathrm{x}_2) \tag{5.5}$$

$$\scriptstyle \exists x_1 . \exists x_2 . R_1(x_1, x_2) \tag{5.6}$$

$$\exists x_1 . \exists_{x_2 . R_1}(x_1, x_2) \tag{5.7}$$

$$\exists x_1 . \qquad \exists x_2 \, . R_1(\qquad x_1, x_2) \tag{5.8}$$

$$\scriptstyle\exists \displaystyle \qquad x_1 . \qquad \exists_{x_2 . R_1}(\qquad x_1, x_2) \tag{5.9}$$

In line 5.1, we are neither able to identify the signs, nor to identify their ordering. That is we cannot pass the steps (2) and (3) of $(*)$, thus this line does not represent a formula. In line 5.2, we are able to identify the signs, but we are not able to identify their ordering. That is we cannot pass the step (3) of $(*)$, thus this line does not represent a formula as well. Moreover, it may be doubted whether step 1 of $(*)$ can be passed without problems. In line 5.3, we are able to identify the signs, but one of these signs does obviously not belong to our alphabet of first order logic, thus this line does not represent a formula. In line 5.4, we are able to identify the signs, all of them belong to to our alphabet of first order logic, and we are able to identify their ordering, that is we reconstruct a sequence of signs of our alphabet. But we see that this sequence is not build according to our rules (we cannot pass the step (4) of $(*)$.). Thus this line does not represent a formula. In the remaining lines, it is not uniquely determined whether the lines represent a formula or not. In line 5.5, the font for the variables has changed. In mathematical texts, different fonts are often used to denote mathematical entities of different kinds, but this is not a general rule. So it depends on the context whether lines 5.5 or 5.6 are accepted to represent formulas. Using different sizes of a font is usually driven by a specific purpose. The same holds for the use of significantly different distances between signs. It is hardly conceivable to find a purpose for using different font sizes or significantly different distances in formulas. Thus it is possible, but not sure, that the lines 5.7–5.9 are not accepted by a mathematician to represent a formula.

In standard books on logic, usually only the last step of $(*)$ is discussed. The main reason for this is the following: The linear notation of formulas corresponds to the way ordinary text is written: Text is assembled of letters which are written side by side and which are read from left to right. As we are used to read texts, we are trained as well to read strings which shall represent formulas. Thus the first three steps of $(*)$ are unconsciously executed when we perceive a representation of a formula.

But as soon we perceive an unfamiliar representation (like in the strings 5.1-5.9), we become aware of the whole process described by $(*)$. We realize that mathematical structures need representations, and in mathematics we have a clear separation between structures and their representations. The representations rely on conventions, either implicit or explicit, based on common cultural background as well as on mathematical socialization, and they are not fully explicated. Nonetheless, this usually poses no problems: Although these conventions can never be fully explicated, as long as we provide representations of mathematical structures in accordance to these conventions, they are strong enough to provide a secure transformation from the external representation of a structure (e.g. on a sheet of paper or on a blackboard) into an internal representation of any mathematician, i.e., they refer to the represented structures in a clear and and non-ambiguous way (as Barwise says in a broader discussion of representations: 'Every representation indicates a genuine possibility' [Bar93]). This yields the next thesis:

> **Thesis 3:** In a mathematical theory, the mathematical structures need representations. A rigor mathematical theory can be developed without providing mathematical definitions for the representations. Instead of that, it is sufficient if we have conventions – either implicit or explicit — which describe the representations, as well as the relationship between structures and their representations, in a clear and non-ambiguous way.

5.4 The Second Approach to Diagrams

In Sect. 5.1, we have argued that in literature, EGs are described in an informal and insufficient way (Thesis 1). Furthermore, we should not mix up graphs and their representations. Particularly, EGs should be defined as abstract, mathematical structures and not as graphical entities (Thesis 2). Nonetheless, we haven't already solved the question whether the representations should be defined mathematically as well.

In Sect. 5.2, we presented some difficulties when we try to define the diagrams mathematically. The arguments of Sect. 5.2 are not strong enough to claim that a mathematical definition of diagrams will always run into problems. In contrast: Finding a appropriate mathematical definition for the diagrams should clarify the points mentioned in Sect. 5.1. That is, a mathematical definition would make clear without ambiguities which diagrams should be considered to be well-formed diagrams of EGs, and which not. Furthermore, the relation between graphs and their representations can be elaborated mathematically as well. But providing mathematical definitions for the diagrams may result in a technical overhead or overspecification of the formalization of EGs and their representations, and none mathematical definition can solve the representation problem finally.

In contrast to the attempt to capture the representations by mathematical definitions, we have seen in Sect. 5.3 that logic is a rigor mathematical theory, although the representations of the types in logic, i.e., formulas, are not fully explicated, but they rely on different conventions. This works because these conventions are mainly based on a common and solid cultural background, namely the form how text is presented.

We now have two approaches to a representation of EGs. We can adopt the approach of thesis 3, that is, the way how EGs are graphically represented is explicated informally in common language. Or, on the other hand, the diagrams are defined mathematically, which is the approach discussed in Sect. 5.2. Of course, the mathematical definition should capture as best as possible the informally provided conventions of the first approach. The advantage of a mathematical definition is its preciseness, but we gain this preciseness for the cost of a technical overspecification of the diagrams.

Moreover, the discussion in Sect. 5.2 showed that even if a mathematical definition of the diagrams is provided, we need (informal) conventions for drawing diagrams. In *both* approaches, we have to provide a set of conventions on how diagrams are written. Roughly speaking: A mathematical definition alone cannot capture exactly the diagrams we intend to use. Mathematical definitions cannot solve the representation problem.

As already said above, arguments like these are not strong enough to *generally* discard mathematical definitions for the token-structures. It has to be estimated whether a mathematical definition is really needed, or whether the conventions for drawing diagrams and for the relationship between representing diagrams and represented structures can be captured sufficiently by descriptions in common language. If the latter is possible, we are able to gain the rigorousness and preciseness of a mathematical theory without a technical overspecification of the diagrams. Thus, for doing logic with diagrams, I claim that the approach of thesis 3 should be preferred. The program of this treatise can now be summarized as follows:

> We have to define EGs as *abstract, mathematical* structures. Particularly, we distinguish between EGs, which are abstract mathematical structures, and their graphical representations (diagrams). We have to provide a set of conventions for drawing a diagram of an EG, and we have to describe the relationship between EGs and their diagrams. Due to these conventions and descriptions, we will show that each diagram uniquely determines an represented EG, and that each EG can be represented by a diagram.

6

Some Remarks to the Books of Zeman, Roberts, and Shin

As already mentioned before, there is some amount of secondary literature about Peirce's graphs, including several publications in journals and some books, written by researchers from different scientific disciplines. In my view, the main treatises are the books of Zeman ([Zem64], 1967), Roberts ([Rob73], 1973), and Shin ([Shi02a]), 2002). Each of this books focuses on different aspects of existential graphs. This holds for this treatise as well: It cannot cover *all* aspects of existential graphs, but focuses on a mathematical elaboration of existential graphs as a diagrammatic approach to mathematical logic.

In this chapter, the scientific orientations of the treatises of Roberts, Shin and Zeman are set out. On the one hand, it will be explicated which aspects of existential graphs are better described in these treatises than in this one. On the hand, I discuss from a mathematical point of view some lacks in these books.

Probably the most prominent book on existential graphs is D. Robert's 'The Existential Graphs of Charles S. Peirce'. This book offers the most comprehensive description of the whole system of existential graphs and its genesis. Particularly, the gamma part of existential graphs is described to a large degree. Peirce describes far more features of gamma graphs than than the broken cuts, which are the main focus of most papers or books which deal with gamma. Moreover, the predecessor of existential graphs, the *entitative graphs*, are described as well. Obviously, Roberts worked through many, many manuscripts of Peirce, and he justifies his elaboration with a lot of quotations. Robert's treatise is definitely an outstanding work.

From a mathematical point of view, compared to the treatises of Shin and Zeman, this book is the most insufficient one. Roberts does not provide any (technical or mathematical) definitions for existential graphs, neither their syntax, semantic, nor inference rules, and he relies solely on the graphical representations of graphs. In the appendix, Roberts provides proofs for the sound- and completeness of Alpha and Beta in which the rules of the calculus of Church for propositional logic are translated to derivations in Alpha, and the rules of the calculus of Quine for first order logic are translated to derivations in Beta. But as neither the existential graphs, nor the translations are defined mathematically, these proofs cannot be understood as strict mathematical proofs. Finally, similar to the treatises of Shin and Zeman, Roberts does not provide a mathematical, extensional semantics for Peirce's graphs.

In contrast to Roberts, J. J. Zeman book 'The Graphical Logic of C. S. Peirce' is, from a mathematical point of view, the best of the books which are here discussed. Zeman provides a mathematical elaboration of Peirce's alpha-graphs, Peirce's beta-graphs, and the part of Peirce's gamma-graphs which extend the beta-part by adding the broken cut.

Zeman does not explicitely discuss the distinction between graphs and graph replicas. Nonetheless, he defines existential graphs inductively as abstract structures. For example, he defines and explains encircling a given graph X by a cut as follows:

1.1ii Where X is an alpha graph, S(X) is an alpha graph.

⋮ ⋮

1.1v The sequence of signs consisting of 'S(' followed by the name of the graph X followed by ')' names a graph; specifically, it names the graph formed by enclosing X by an alpha cut.

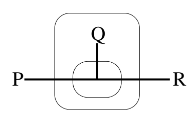
We see that Zeman defines existential graph as sequences of signs which are graphically depicted.

Like in the other treatises, Zeman does not provide a mathematical, extensional semantics for Peirce's graphs. Instead of that, he defines mappings between the systems Alpha, Beta, Gamma and appropriate systems of propositional, first order, and modal logic. These translations from graphs to symbolic are correct, but in my eyes a little bit clumsy (you can find a discussion of this in Shin's book as well). Zeman's book is the most formal book among the books discussed in this chapter. It does not contain a readable introduction into existential graphs and presupposes some knowledge of them. On the other hand, it is the most precise elaboration and the only one which explicates the mathematical relationship between Gamma and modal logic.

Sun Joo Shin treats in her book 'The Iconic Logic of Peirce's Graphs' only the Alpha and Beta part of existential graphs. Her interest in existential graphs is philosophically driven, and she uses existential graphs as a case study for her goal to provide a formal approach to diagrammatic reasoning. As the title of her book suggest, she focuses on the diagrammatic aspects, particularly the iconicity, of existential graphs. In order to do this, she provides an overview into the research field of diagrammatic reasoning and into the semiotics of Peirce. Her main critics to the handling of EGs in secondary literature so far is as follows: '*Despite the fact that EG is known to be a graphical system, the* *iconic features of EG have not been fully implemented either in any existing reading algorithm [translations from graphs to symbolic logic] or in how the transformation rules are stated.*' Her goal is to amend this flaw. She compares symbolic and diagrammatic approaches to mathematical logic and works out that the '*long-standing prejudice against non-symbolic representation in logic, mathematics, and computer science*'are due to the fact that diagrammatic systems are evaluated in terms of symbolic systems. Then, based on her arguments, she reconsiders Alpha and Beta from iconic aspects and rewrite the reading algorithms, that is, the translations from graphs to symbolic logic, and the transformation rules in order to improve their iconicity and naturalness.

Unfortunately, as already discussed in Sect. 5.1, her approach lacks mathematical preciseness. First of all, a distinction of graphs and graph replicas is missing (which is in my eyes very surprising, as Shin elaborates carefully the semiotic aspects of existential graphs). Secondly, as argued in Sect. 5.1, the definitions (and later on, theorems and proofs) she provides cannot be considered to be mathematical definitions. In fact, this leads to a mistake in her reading algorithm, and some of her newly implemented transformation rules are not sound. The flaw in her her reading algorithm and a possible way, based on the results of this treatise, to fix it is presented in [Dau06a], and in the comment below. As done for her reading algorithm in [Dau06a], Shin's flaws can be fixed, but they clearly show the lack of mathematical preciseness in her elaboration. Finally, Shin does not provide an extensional semantics for Peirce's graphs: Her reading algorithms are translations to symbolic logic, thus translations from one formal system to another.

Comment: To make the flaws more precise: On page 128, Shin translates the graph

$\mathfrak{G}_z :=$ 

to $\mathcal{FO}$ as follows: $\exists x.(Px \wedge \forall y.(x = y \vee \neg Qy) \wedge Rx)$. This formula can be converted to a s slightly better readable formula as follows, which is denoted by f_s:

$$f_s := \exists x.(Px \wedge Rx \wedge \forall y.(Qy \to x = y))$$

In contrast to that, the correct approach of Zeman yields a translation which is equivalent to

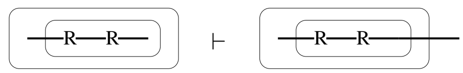
$$f_z := \exists x.\exists z.(Px \wedge Rz \wedge \forall y.(Qy \to (x = y \wedge y = z)))$$

It is easy to see that these two translation are not semantically equivalent; we have $f_s \models f_z$., but $f_z \not\models f_s$. The mistake in Shin's reading algorithm can be found in her rule 2(b). I discussed this with her in personal communication, and Shin reformulates in a letter her rule as follows:

2(b) If a branch of an LI network crosses an odd number of cuts entirely (i.e. an odd number of cuts clip an LI into more than one part), then

(i) assign a different type of a variable to the outermost part of each branch of the LI which joins in the cut, and
(ii) (ii) at the joint of branches (inside the innermost cut of these odd number of cuts), write vi = vj, where vi and vj are assigned to each branch (in the above process).

In her transformation rules, the rule NR3(b)(iii), which allows to extend a LoI outwards through an even number of cuts, and the corresponding dual rule NR4(b)(iii) are not sound. The rule NR3(b)(iii) allows the following, non-valid derivation:



The rule NR5(b), which allows to join two loose ends of identity lines 'if the subgraphs to be connected are tokens of the same time', and the corresponding dual rule NR6(a) are not sound. Although it is not totally clear what exactly is meant by 'the subgraphs to be connected are tokens of the same time', the transformation rule NR5(b) the following, non-valid derivation:

Shin wants to provide a diagrammatic system of logic which is precise and in which the advantages of diagrammatic systems are implemented as best as possible. Due to the lack of mathematical preciseness and the mistakes in her reading algorithm and transformation rules, Shin's book unfortunately cannot be understood to be a formally precise, diagrammatic approach to mathematical logic. Nonetheless, her book is an excellent introduction into the philosophical aspects of diagrammatic reasoning and existential graphs, and her arguments for the development of diagrammatic systems, for unfolding their specific advantages, particularly their iconic features, and for an evaluation to diagrammatic systems which is not based on criteria adopted from symbolic systems remain true. Particularly, her book is turns out to be very good help and guide for the implementation of any diagrammatic logic system.

Alpha

7

Syntax for Alpha Graphs

In this chapter, Alpha graphs and some elementary technical terms for them (like *subgraph*) are formally defined as abstract, mathematical structures. Due to the discussion in the last chapters, it is moreover explained how these structures can be graphically represented.

Alpha graphs are built up from two syntactical devices: sentence symbols and cuts. We first fix the sentence symbols, which we call *propositional variables*.

Definition 7.1 (Propositional Variables). *Let* $\mathcal{P} := \{P_1, P_2, P_3, \ldots\}$ *be a countably infinite set of* PROPOSITIONAL VARIABLES *.*

Alpha graphs are built up from propositional variables and cuts, e.g., by the rules described in Definition 5.1. We can try to transform this inductive definition into a inductive mathematical definition. This is possible, but here we define alpha graphs in one step. Let us briefly discuss what we have to capture in the mathematical definition. Consider first the two diagrams depicted in Fig. 7.1.

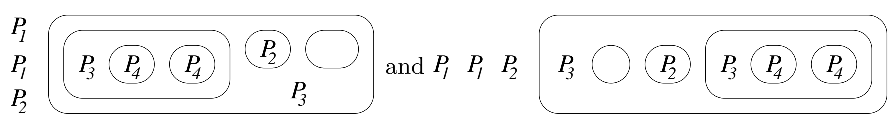


Fig. 7.1. Two Diagrams of the same Alpha graph

According to the discussion in Sec.5.4, these two diagrams represent the same Alpha graph. We have the same occurrences of propositional variables and cuts, although the propositional variables are placed in different locations and although the cuts are drawn differently. Remember that we have said in the discussion of Fig. 5.1 that the shape of a cut has no relevance. The same holds in some respect for the location of the occurrence of a propositional variable. Given a cut, the only fact we have to know which other items of a graph (cuts or occurrences of propositional variables) are enclosed by this cut and which are not. For example, in both diagrams of Fig. 7.1 the outermost cut encloses exactly all other cuts, one occurrence of the propositional variable P_2 and all occurrences of the propositional variables P_3 and P_4. We will say more specifically that the occurrence of the propositional variable P_2 and the two cuts are enclosed *directly*, while all other enclosed items are enclosed *indirectly*, and we will say that the items which are directly enclosed by a cut are placed in the *area* of this cut. It is convenient to say that the outermost items (the outermost cut, the two occurrences of P_1 and one occurrence of P_2 are placed on the area of the sheet of assertion. This enclosing-relation is the main structural relation between cuts and the other items of a graph. For this reason, the two diagrams represent the same alpha graph: We have one-to-one-correspondences between the occurrences of the propositional variables in the left diagram to the propositional variables in the right diagram and between the cuts in the left diagram to the cuts in the right diagram such that the enclosing-relation is respected.

Let us consider two further diagrams.

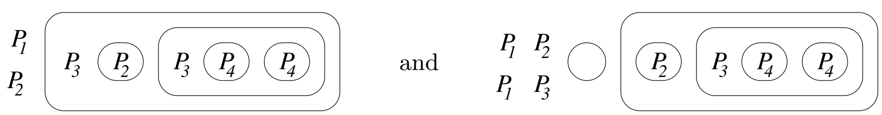


Fig. 7.2. Two Diagrams of different Alpha graphs

The diagrams in Fig. 7.2 represent two alpha graphs which are distinct from the graph of Figure 7.1. In the left diagram, we have different sets of cuts and occurrences of propositional variables (compared to the diagrams of Fig. 7.1). In the right diagram, we have the same sets of cuts and occurrences of propositional variables like in Figure 7.1), but we have a different enclosing-relation.

A mathematical definition of Alpha graphs must therefore capture

1. a set of occurrences of propositional variables,
2. a set of cuts, and
3. the above-mentioned enclosing-relation.

To distinguish between propositional variables and occurrences of propositional variables, we will introduce *vertices* which are labeled with propositional variables. This yields the possibility that a propositional variable may occur several times in a graph, even in the same cut. The cuts are introduced as elements of a set *Cut*. It is reasonable to introduce the sheet of assertion as own syntactical device. The enclosing-relation will be captured by a mapping *area*, which assigns to the to the sheet of assertion and to each cut the set of all other elements of the graph which are directly enclosed by the cut. We know that we have some restrictions for cuts, e.g. cuts must not overlap. These restrictions are captured mathematically by conditions for the mapping *area*. In order to distinguish the mathematically defined alpha graphs from the Peircean Alpha graphs, they are termed *formal* alpha graphs. Formal Alpha graphs are now defined as follows:

Definition 7.2 (Formal Alpha Graph). *A* FORMAL ALPHA GRAPH *is a 5-tuple* $(V, \top, Cut, area, \kappa)$ *with*

- V *and* Cut *are disjoint, finite sets whose elements are called* VERTICES *and* CUTS*, respectively,*
- $\top$ *is a single element with* $\top \notin V \cup Cut$*, called the* SHEET OF ASSERTION*,*
- $area : Cut \cup \{\top\} \to \mathfrak{P}(V \cup Cut)$ *is a mapping such that*
  a) $c_1 \neq c_2 \Rightarrow area(c_1) \cap area(c_2) = \emptyset$,
  b) $V \cup Cut = \bigcup_{d \in Cut \cup \{\top\}} area(d)$,
  c) $c \notin area^n(c)$ *for each* $c \in Cut \cup \{\top\}$ *and* $n \in \mathbb{N}$ *(with* $area^0(c) := \{c\}$ *and* $area^{n+1}(c) := \bigcup \{area(d) \mid d \in area^n(c)\}$*), and*
- $\kappa : V \to \mathcal{P}$ *is a mapping.*

The elements of $Cut \cup \{\top\}$ *are called* CONTEXTS*. As we have for every* $x \in V \cup Cut$ *exactly one context* c *with* $x \in area(c)$*, we can write* $c = area^{-1}(x)$ *for every* $x \in area(c)$*, or even more simple and suggestive:* $c = ctx(x)$*.*

In this definition,[1] we have already captured some important technical terms. Mainly we have defined the *sheet of assertion* and the *cuts* of formal Alpha graphs, and we have introduced the so-far informally described mapping *area*. In the following, we will often speak more simply of 'graphs' instead of 'formal Alpha graphs'.

There is a crucial difference between formal Alpha graphs and most other languages of logic: Formal Alpha graphs are defined in one step. In contrast to that, the well-formed formulas of a language for logic are usually built up inductively. This is mainly done for the following reason: In a linear representations of logic, an inductive definition of formulas is much more easier to provide – often an inductive definition is the canonical definition – than a definition in one step. Moreover, usually the formulas in a linear representations of logic have an unique derivational history, from which a methodology

[1] For Alpha, there are more simple definitions possible. But it will turn out that an extension of this definition is a convenient formalization of Beta graphs. For this reason, this formal definition of Alpha graphs is provided here.

for proofs on formulas is obtained: The proofs are carried out by a principle called *recursion on the construction of formulas* or *induction principle* for short.

This reason do not necessarily hold for the definition of existential graphs. Especially for Beta, a inductive definition for Beta graphs is not straight forward (e.g., there are essential differences in the inductive definitions of Zeman and Shin for Beta graphs), and a inductive definition is not necessarily shorter or even easier than a definition in one step.

More importantly, even if an inductive definition for existential graphs is provided, it is likely that this definition does not lead to a unique derivational history of graphs. Then, for any further definition on the graphs which is carried out along their inductive construction, it has to be shown that it is well-defined. For example, if the semantics is defined inductively, it has to be shown that different derivational histories of a given graph $\mathfrak{G}$ lead nonenetheless to the same semantics of $\mathfrak{G}$ (in the part 'extending the system', a specific class of graphs – so-called *PAL-graphs*, is defined inductively, and on page 309, we will shortly come back to this discussion). Thus the main advantage of inductive definitions gets lost for existential graphs. For this reason, although it is possible to provide inductive definitions for Alpha and Beta, in this treatise the decision is taken to define the graphs in a non-inductive manner.

As just mentioned, in an inductively defined language, the well-formed formulas bear a structure which is obtained from their (normally uniquely determined) inductive construction. Of course, Alpha graphs bear a structure as well: A cut of the graph may contain other cuts, but cuts may not intersect. Thus, for two (different) cuts, we have three possibilities: The first cut encloses the second one, the second cut encloses the first one, or the two cuts are incomparable. If we incorporate the sheet of assertion into this consideration, it has to be expected that this idea induces an order $\leq$ on the contexts (the naive definition of $\leq$ is to define $c < d$ iff c is 'deeper nested' than d) which should be a tree, having the sheet of assertion $\top$ as greatest element. The next definition is the mathematical implementation of this naive idea.

As we mathematize *informal* given entities – here: (diagrams of) Alpha graphs, we cannot *prove* (mathematically) that the mathematization, e.g. the definition of formal Alpha graphs, is 'correct'. But as we have a mathematical definition of Alpha graphs and as the order $\leq$ will be mathematically defined as well, we must be able to *prove* the proposition that $\leq$ is a tree with the sheet of assertion as greatest element. The attempt to prove this proposition can be understood as a test on our mathematical 're-engineering' of Alpha graphs. If we cannot prove this proposition, our mathematization of Alpha graphs does not capture crucial features of Alpha graphs, thus we should rework the definition. If we can prove this proposition, this is a good argument that our definition is 'right'. In fact, this proposition will turn out to be the main instrument to argue that each formal alpha graph can be represented graphically.

In the next definition, we define the ordering on the vertices and contexts which will capture the enclosing-relation.

Definition 7.3 (Ordering on the Contexts, Enclosing Relation). *Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a formal alpha graph. Now we define a mapping $\beta : V \cup Cut \cup \{\top\} \to Cut \cup \{\top\}$ by*

$$\beta(x) := \begin{cases} x \text{ for } x \in Cut \cup \{\top\} \\ ctx(x) \text{ for } x \in V \end{cases},$$

and set $x \leq y :\Longleftrightarrow \exists n \in \mathbb{N}_0 . \beta(x) \in area^n(\beta(y))$ for $x, y \in V \cup Cut \cup \{\top\}$.

To avoid misunderstandings, let

$$x < y :\Longleftrightarrow x \leq y \wedge y \not\leq x \quad \text{and} \quad x \lneq y :\Longleftrightarrow x \leq y \wedge y \neq x$$

. For a context $c \in Cut \cup \{\top\}$, we set furthermore

$$\leq[c] := \{x \in V \cup Cut \cup \{\top\} \mid x \leq c\} \text{ and } \lneq[c] := \{x \in V \cup Cut \cup \{\top\} \mid x \lneq c\}.$$

Every element x of $\bigcup_{n\in\mathbb{N}} area^n(c)$ is said to be ENCLOSED BY *c, and vice versa: c is said to* ENCLOSE *x. For every element of $area(c)$, we say more specifically that it is* DIRECTLY ENCLOSED BY *c.*

The relation $\leq$ is implemented to ease the mathematical treatment of the enclosing-relation. First of all, let us show that the enclosing-relation can be described with the relation $\leq$.

Lemma 7.4 (Order Ideals Generated by Contexts). *Let a formal alpha graph $(V, \top, Cut, area, \kappa)$ be given and let $c \in Cut \cup \{\top\}$. Then:*

$$\leq[c] = \bigcup\{area^n(c) \mid n \in \mathbb{N}_0\} \quad , \text{ and } \quad \lneq[c] = \bigcup\{area^n(c) \mid n \in \mathbb{N}\} \ .$$

For $c_1, c_2 \in Cut \cup \{\top\}$ we have the following implication:

$$c_1 \in \lneq[c_2] \Longrightarrow \leq[c_1] \subseteq \lneq[c_2].$$

Proof: If d is context, we have

$$d \in \leq[c] \Longleftrightarrow \exists n \in \mathbb{N}_0 . d = \beta(d) \in area^n(c)\} \Longleftrightarrow d \in \bigcup_{n\in\mathbb{N}_0} area^n(c)$$

Analogously, if v is a vertex, we have

$$v \in \leq[c] \Longleftrightarrow \exists n \in \mathbb{N}_0 . ctx(v) = \beta(v) \in area^n(c)\} \Longleftrightarrow d \in \bigcup_{n\in\mathbb{N}} area^n(c)$$

As moreover $area^0(c) = \{c\}$ does not contain any vertices, we are done. $\Box$

If we have two cut-lines in the graphical representations of an alpha graph, either one of the cut-lines is enclosed by the other one, or (as cut-lines must not intersect) there is no element in the diagram which is enclosed by both cut-lines. Of course, this statement has to hold for formal alpha graphs as well, which is the proposition of the next lemma.

Lemma 7.5 (Relations between Order Ideals). *For a formal alpha graph* $(V, \top, Cut, area, \kappa)$ *and two contexts* $c_1 \neq c_2$*, exactly one of the following conditions holds:*

$$\text{i)} \quad \leq[c_1] \subseteq \lneq[c_2] \qquad \text{ii)} \quad \leq[c_2] \subseteq \lneq[c_1] \qquad \text{iii)} \quad \leq[c_1] \cap \leq[c_2] = \emptyset$$

Proof: It is quite evident that neither i) and iii) nor ii) and iii) can hold simultaneously. Suppose that i) and ii) hold. We get $\leq[c_1] \subseteq \lneq[c_2] \subseteq \lneq[c_1]$, hence $c_1 \in \lneq[c_1]$, in contradiction to c) for *area* in Def. 7.2 and to Lem. 7.4. Now it is sufficient to prove the following: If iii) is not satisfied, then i) or ii) holds. So we assume that iii) does not hold. Then we have

$$\begin{aligned}
\emptyset \neq & \leq[c_1] \cap \leq[c_2] \\
&= (\{c_1\} \cup \lneq[c_1]) \cap (\{c_2\} \cup \lneq[c_2]) \\
&= (\{c_1\} \cap \{c_2\}) \cup (\{c_1\} \cap \lneq[c_2]) \cup (\{c_2\} \cap \lneq[c_1]) \cup (\lneq[c_1] \cap \lneq[c_2])
\end{aligned}$$

From $c_1 \neq c_2$ we conclude $\{c_1\} \cap \{c_2\} = \emptyset$. If $\{c_1\} \cap \lneq[c_2] \neq \emptyset$ holds, i.e. $c_1 \in \lneq[c_2]$, Lem. 7.4 yields i). Analogously follows ii) from $\{c_2\} \cap \lneq[c_1] \neq \emptyset$. So it remains to consider the case $\lneq[c_1] \cap \lneq[c_2] \neq \emptyset$. For this case, we choose $x \in area^m(c_1) \cap area^n(c_2)$ such that $n + m$ is minimal. We distinguish the following four cases:

- $m = 1 = n$: This yields $x \in area(c_1) \cap area(c_2)$ in contradiction to $c_1 \neq c_2$ and condition a) of Def. 7.2.
- $m = 1, n > 1$: Let ${c_2}' \in area^{n-1}(c_2)$ such that $x \in area({c_2}')$. From a) of Def. 7.2 and $x \in area(c_1) \cap area({c_2}')$ we conclude $c_1 = {c_2}'$. Hence $c_1 \in area^{n-1}(c_2)$ holds, and we get $c_1 \cup \lneq[c_1] \subseteq \lneq[c_2]$, i.e., condition i).
- $m > 1, n = 1$: From this, we conclude ii) analogously to the last case.
- $m > 1, n > 1$: Let ${c_1}' \in area^{m-1}(c_1)$ such that $x \in area({c_1}')$, and let ${c_2}' \in area^{n-1}(c_2)$ such that $x \in area({c_2}')$. We get $area({c_1}') \cap area({c_2}') \neq \emptyset$, hence ${c_1}' = {c_2}'$. This yields $area^{m-1}(c_1) \cap area^{n-1}(c_2) \neq \emptyset$, in contradiction to the minimality of $m + n$. $\Box$

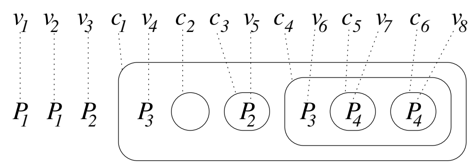
Now we get the following lemma:

Lemma 7.6 ($\leq$ is a Tree on $Cut \cup \{\top\}$). *Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a formal alpha graph. Then $\leq$ is an quasiorder on $V \cup Cut \cup \{\top\}$, and the restriction of $\leq$ to $Cut \cup \{\top\}$ is an order on $Cut \cup \{\top\}$ which is a tree with the sheet of assertion $\top$ as greatest element.*

Proof: We have $x \leq y \iff \beta(x) \leq \beta(y) \iff \leq[\beta(x)] \subseteq \leq[\beta(y)]$. Hence $\leq$ is a quasiorder. Now Lem. 7.5 yields that the restriction of $\leq$ to $Cut \cup \{\top\}$ is an order which is furthermore a tree.

As $Cut \cup \{\top\}$ is finite, is contains a maximal element c. Assume that $c \neq \top$. Then condition b) for *area* in Def. 7.2 yields a d with $c \in area(d)$. From this we conclude $c < d$, a contradiction to the maximality of c. Hence $\top$ is the only maximal element of $Cut \cup \{\top\}$, i.e., $\top$ is the greatest element. □

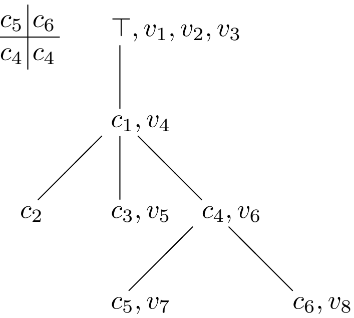
To provide an example of a formal Alpha graph, we consider the right diagram of Fig. 7.1. Additionally, we have labeled the vertices and cuts with names for pairwise distinct elements. Below the diagram, the formal Alpha graph which is represented by the diagram is provided.



$$\begin{array}{ll}
\mathfrak{G} := (\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}, \top, \{c_1, c_2, c_3, c_4, c_5, c_6\}, & \Big\} \; V, \top, Cut \\
\quad \left.\begin{array}{l} \{(\top, \{v_1, v_2, v_3, c_1\}), (c_1, \{v_4, c_2, c_3, c_4\}), (c_2, \emptyset), \\ \; (c_3, \{v_5\}), (c_4, \{v_6, c_5, c_6\}), (c_5, \{v_7\}), (c_6, \{v_8\})\}, \end{array}\right\} & area \\
\quad \left.\begin{array}{l} \{(v_1, P_1), (v_2, P_1), (v_3, P_2), (v_4, P_3), \\ \; (v_5, P_2), (v_6, P_3), (v_7, P_4), (v_8, P_4)\}) \end{array}\right\} & \kappa
\end{array}$$

Next we show the mapping *ctx*, and the quasiorder $\leq$ is presented by its Hasse-diagram.

x	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	c_1	c_2	c_3	c_4	c_5	c_6
$ctx(x)$	$\top$	$\top$	$\top$	c_1	c_3	c_4	c_5	c_6	$\top$	c_1	c_1	c_1	c_4	c_4



Note that we did not further specify the objects $v_1, \ldots, v_8$, $\top$, and $c_1, , \ldots, c_6$. It is quite obvious that a diagram of an EG cannot determine the mathematical objects for vertices or cuts, it only determines the *relationships* between these objects. In fact we can choose arbitrary sets for these mathematical objects (we only have to take into account that $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$, $\{\top\}$, $\{c_1, c_2, c_3, c_4, c_5, c_6\}$ must be pairwise disjoint). In other words: The diagram of a graph determines the graph only up to *isomorphism.* The isomorphism-relation is canonically defined as follows:

Definition 7.7 (Isomorphism). *Let $\mathfrak{G}_i := (V_i, \top_i, Cut_i, area_i, \kappa_i)$ with $i = 1, 2$ be two formal alpha graphs. Then $f = f_V \dot{\cup} f_{Cut}$ is called* ISOMORPHISM, *if $f_V : V_1 \to V_2$ and $f_{Cut} : Cut_1 \cup \{\top_1\} \to Cut_2 \cup \{\top_2\}$ are bijective and satisfy $f_{Cut}(\top_1) = \top_2$ such that $f[area_1(c)] = area_2(f(c))$ for each $c \in Cut_1 \cup \{\top_1\}$ (we write $f[X]$ for $\{f(x) \mid x \in X\}$), and $\kappa_1(v) = \kappa_2(f_V(v))$ for all $v \in V_1$.*

From now on, isomorphic formal alpha graphs are implicitely identified.

We have seen that a DIAGRAM OF AN ALPHA GRAPH is a diagram which is built up from two different kinds of items, namely of closed, double-point-free and smooth curves which represent cuts (we will call them *cut-lines*), and signs which denote the propositional variables P_i, such that two different items of the diagram neither overlap nor intersect.

So far, we have not discussed why the relationship between formal alpha graphs and their graphical representations. This is needed to argue that formal alpha graphs capture exactly the essential features of the diagrams of alpha graphs. We have to answer the following two questions:

1. Can each formal alpha graph be represented by a diagram?
2. Does each diagram represent an alpha graph? How can this graph be reconstructed from the diagram?

In order to answer these questions, a further clarification of the term 'diagram' of an alpha graph is needed.

Informal Definition 7.8 (Diagram of Alpha Graphs) *A* DIAGRAM OF AN ALPHA GRAPH *is a diagram which is built up from two different kinds of items, namely of*

1. *closed, doublepoint-free and smooth curves which are named* CUT-LINES, *and*
2. *signs which denote the propositional variables P_i,*

such that the following conditions hold:

1. *Nothing else may occur (i.e. there are no other graphical items but cut-lines or signs denoting a propositional variable), and*
2. *two different items the diagram must neither overlap nor intersect.*

Obviously, a cut-line separates the plane into two distinct regions: What is outside the cut-line and what is inside the cut-line. Of course we say that another item of the diagram is enclosed by this cut-line if and only if it is placed in the inner region of this cut-line. Furthermore, we agree that each item is placed on the sheet of assertion, thus it is convinient to say that each item is *enclosed* by the sheet of assertion. As no items, of the diagram, particulary no cut-lines, may neither overlap nor intersect, we can *decide* for two items whether one of them enclosed the other one or not. Now we say that an item x is *directly enclosed* by a cut-line cl (resp. by the sheet of assertion) if and only if it is enclosed by cl (resp. by the sheet of assertion) and there is no other cut-line cl' which is enlosed by cl (resp. by the sheet of assertion) and which encloses x.

Now let a diagram of an alpha graph be given. It is easy to find (up to isomorphism) the corresponding formal Alpha graph $(V, \top, Cut, area, \kappa)$: We choose sets V and Cut of which its elements shall stand for the occurences of propositional variables resp. the cut-lines in the diagram, and the mapping κ is defined accordingly. Then, the mapping area $area$ is now defined as follows: Let $c \in Cut$ be a cut. So we have a uniquely given cut-line cl in our diagram which corresponds to c. Now let $area(c)$ be the set of all $x \in V \dot\cup Cut$ such that x corresponds to an item of the diagram (an occurence of a PV or a cut-line) which is directly enclosed by the cut-line cl. Furthermore, let $area(\top)$ be the set of all $x \in V \dot\cup Cut$ such that x corresponds to an item which is not enclosed by any cut-line. So we obtain a formal Alpha graph which is represented by the diagram, i.e., we have an one-to-one correspondence between the occurences of the signs which denote the propositional variables in the diagram and V, and a an one-to-one correspondence between the cut-lines of the diagram and Cut, such that the following holds:

1. If we have an occurence of a sign which denotes a propositional variable P_i in the diagram and which is mapped to a vertex v, then $\kappa(v) = P_i$.
2. Let an arbitrary item of the diagram be given, which is mapped to an $x \in V \dot\cup Cut$, and let an arbitrary cut-line be given (resp. the sheet of assertion), which is mapped to a cut $c \in Cut$ (resp. to $\top$). Then the item is directly enclosed by the cut-line (resp. by the sheet of assertion) if and only if $x \in area(c)$ (resp. $x \in area(\top)$).

Now let on the other hand be a formal alpha graph $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be given. Using Lem. 7.6, inductively over the tree $(Cut \cup \{\top\}, \leq)$, we can assign to each context $c \in Cut \cup \{\top\}$ a diagram $Diag(c)$. So let $c \in Cut \cup \{\top\}$ be a context such that we have already constructed $Diag(d)$ for each $d \in area(c)$. Then $Diag(c)$ is constructed by writing side by side (without overlapping) the following items:

1. For each $v \in area(c) \cap V$, we write an sign which denotes the propositional variables $\kappa(v)$, and

2. For each $d \in area(c) \cap Cut$, we write the diagram $Diag(d)$ and a further cut-line which encloses $Diag(d)$ (that is, it encloses excatly all items of $Diag(d)$).

It is easy to see that $Diag(\top)$ is a diagram of an alpha graph which represents $\mathfrak{G}$.

To summarize: We provided an (informal) definition of diagrams of alpha graphs, as well as a mathematical (formal) definition of formal alpha graphs. We explicated when a diagram is a representation of a formal alpha graph. We have argued that each diagram of an alpha graph is a representation of a formal alpha graph, and, vice versa, that each formal alpha graph can be represented by a diagram.

This is similar to the handling of formulas and their representations, as we have discussed it in Section 5.4. Thus, from now on, we have the opportunity to use the diagrams of formal alpha graphs in mathematical proofs.

For the further treatise, especially for the calculus, the notation of a *subgraph* is needed. Informally spoken, a SUBGRAPH is a part of a graph placed in a context c such that

- if the subgraph contains a cut d, then it contains all what is enclosed by d, i.e. $\lneq[d]$, and
- every element of the subgraph is enclosed by another cut of the subgraph or by c.

Definition 7.9 (Subgraph). *Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a formal alpha graph. The formal graph alpha $\mathfrak{G}' := (V', \top', Cut', area', \kappa')$ is called a* SUBGRAPH OF $\mathfrak{G}$ IN THE CONTEXT $\top'$ *if*

- *$V' \subseteq V$, $Cut' \subseteq Cut$ and $\top' \in Cut \cup \{\top\}$,*
- *$\kappa' = \kappa\big|_{V'}$ (i.e., the mapping κ' is the restriction of κ to V'),*
- *$area'(\top') = area(\top') \cap (V' \cup Cut')$ and $area'(d) = area(d)$ for each cut $d \in Cut'$, and*
- *$ctx(x) \in Cut' \cup \{\top'\}$ for each $x \in V' \cup Cut'$.*

We write: $\mathfrak{G}' \subseteq \mathfrak{G}$ and $area^{-1}(\mathfrak{G}') = \top'$ resp. $ctx(\mathfrak{G}') = \top'$.

The claim that if a subgraph contains a cut d, is has to contain all what is enclosed by d, is somewhat hidden in the third condition of its definition. That this claim is fulfilled is shown with the next lemma.

Lemma 7.10 (Subgraphs Respect Enclosing Relation). *Let a formal alpha graph $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be given, let $\mathfrak{G}' := (V', \top', Cut', area', \kappa')$ be a subgraph. If $d \in Cut'$, then $\lneq[d] \subseteq V' \cup Cut'$.*

Proof: It holds $x \lneq d \Leftrightarrow \exists n \in \mathbb{N}\colon x \in area^n(d) \Leftrightarrow \exists n \in \mathbb{N}\, \exists d_1, \ldots, d_n \in Cut\colon$ $d = d_1$, $d_2 \in area(d_1), \ldots, d_{n-1} \in area(d_n)$ and $x \in area(d_n)$ for each cut or vertex $x \in V \cup Cut$.

We have $d = d_1 \in Cut'$. The third condition for subgraphs yields now that $d_2 \in area(d_1) = area'(d_1)$, thus we have $d_2 \in Cut'$ as well. Analogously, we can now conclude that we have $d_3 \in Cut'$, etc., and finally we conclude $x \in area(d_n) = area'(d_n)$, thus $x \in V' \cup Cut'$. □

In Figs. 7.3 and 7.4, some examples for the definition of subgraphs, based on the graph of Figure 7.1, are provided. We consider substructures of this graphs which are depicted by printing them black (the items which do not belong to the substructures are grey). Note that we have for example two different subgraphs P_4 . Considered as graphs, these two subgraphs are isomorphic and therefore treated to be identical. Considered as subgraphs, there are treated to be different.

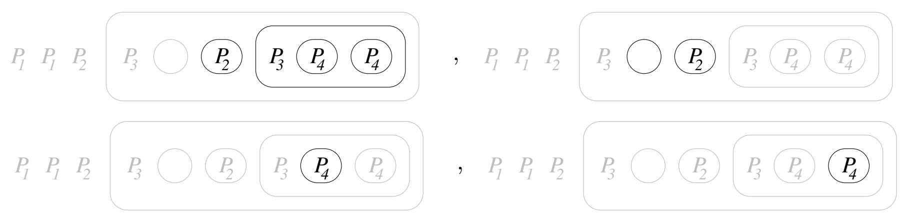


Fig. 7.3. Four different subgraphs

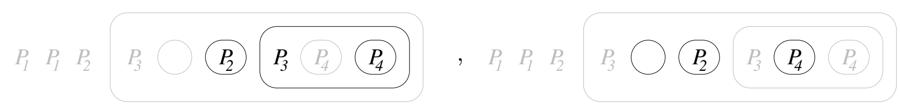


Fig. 7.4. Two substructures which are no subgraphs

The next two definitions are essential in the definitions of calculus we will present in Chpt. 8. Most of the rules in this calculus modify only parts of a graph which are enclosed by a specific context, the rest of the graph remains unchanged. We will say that the starting graph and the resulting graph are isomorphic except for that context, which is captured by Def. 7.11.

Definition 7.11 (Partial Isomorphism). *Let two formal alpha graphs* $\mathfrak{G}_i := (V_i, \top_i, Cut_i, area_i, \kappa_i)$ *with contexts* $c_i \in Cut_i \cup \{\top_i\}$, $i = 1, 2$ *be given. For* $i = 1, 2$, *we set*

- ${V_i}' := \{v \in V_i \mid v \not\leq c_i\}$ *and*
- ${Cut_i}' := \{d \in Cut_i \cup \{\top_i\} \mid d \not< c_i\}$.

Let ${\mathfrak{G}_i}'$ be the restriction of $\mathfrak{G}_i$ to ${V_i}'$ and ${Cut_i}'$, i.e., for ${area_i}' := area_i\big|_{{Cut_i}'}$ and ${\kappa_i}' := \kappa_i\big|_{{V_i}'}$, we set ${\mathfrak{G}_i}' := ({V_i}', \top, {Cut_i}', {area_i}', {\kappa_i}')$. If $f = f_V \,\dot{\cup}\, f_{Cut}$ is an isomorphism between $\mathfrak{G}'_1$ and $\mathfrak{G}'_2$ with $f_{Cut}(c_1) = c_2$, then f is called (PARTIAL) ISOMORPHISM FROM $\mathfrak{G}_1$ TO $\mathfrak{G}_2$ EXCEPT FOR $c_1 \in Cut_1 \cup \{\top_1\}$ AND $c_2 \in Cut_2 \cup \{\top_2\}$.

Let us draw two technical remarks on this definition.

1. Please note that we have defined ${Cut_i}' := \{d \in Cut_i \cup \{\top_i\} \mid d \not< c_i\}$ instead of ${Cut_i}' := \{d \in Cut_i \cup \{\top_i\} \mid d \not\leq c_i\}$. This yields that we have $c_i \in {Cut_i}'$ for $i = 1, 2$, but not $v_i \in {V_i}'$ for $v_i \in area(c_i) \cap V_i$, $i = 1, 2$ and not $e_i \in {E_i}'$ for $e_i \in area(c_i) \cap E_i$, $i = 1, 2$.
2. For the restrictions ${area_i}' := area_i\big|_{{Cut_i}'}$, we agree that the range of ${area_i}'$ is restricted to ${V_i}' \cup {Cut_i}'$ as well. For $d_i \in {Cut_i}'$ with $d \neq c_i$, this makes no difference: We simply have ${area_i}'(d_i) = area_i(d_i)$, but for c_i, we have ${area_i}'(c_i) = \emptyset$.

For some rules we have to distinguish whether this context is enclosed by an odd or even number of cuts. For this reason the next definition is needed.

Definition 7.12 (Evenly/Oddly Enclosed, Pos./Neg. Contexts). *Let $\mathfrak{G} = (V, \top, Cut, area, \kappa)$ be a formal alpha graph, let x be a subgraph or let x be an element of $V \cup Cut \cup \{\top\}$. We set set $n := |\{c \in Cut \mid x \in \leq[c]\}|$. If n is even, x is said to be* EVENLY ENCLOSED *or* POSITIVE ENCLOSED*, otherwise x is said to be* ODDLY ENCLOSED *or* NEGATIVE ENCLOSED*. The sheet of assertion $\top$ and each oddly enclosed cut is called a* POSITIVE CONTEXT*, and each an evenly enclosed cut is called* NEGATIVE CONTEXT.

Writing diagrams side by side is a common operation called *juxtaposition*. On the level of formal Alpha graphs, it is the disjoint[2] union of a set of graphs, which is captured by the next definition.

Definition 7.13 (Juxtaposition of Formal Alpha Graphs). *For an $n \in \mathbb{N}_0$ and for each $i = 1, \ldots, n$, let $\mathfrak{G}_i := (Ve_i, \top e_i, Cute_i, areae_i, \kappa e_i)$ be a formal alpha graph. The* JUXTAPOSITION OF THE $\mathfrak{G}_i$ *is defined to be the following formal alpha graph $\mathfrak{G} := (V, \top, Cut, area, \kappa)$:*

- $V := \bigcup_{i=1,\ldots,n} V_i \times \{i\}$ *, and* $Cut := \bigcup_{i=1,\ldots,n} Cut_i \times \{i\}$ *,*
- *area is defined as follows:* $\begin{array}{c} area((c,i)) = area_i(c) \times \{i\} \text{ for } c \in Cut_i \\ area(\top) = \bigcup_{i=1,\ldots,n} area_i(\top_i) \times \{i\} \end{array}$ *, and*
- $\kappa(v, i) := \kappa_i(v)$ *for all* $v \in V \cup E$ *and* $i = 1, \ldots, n$.

[2] In order to ensure that the graphs are disjoint, a simple technical trick – all sets of the ith graph are indexed by i – is applied in the definition.

In the graphical notation, the juxtaposition of the $\mathfrak{G}_i$ *is simply noted by writing the graphs next to each other, i.e. we write:* $\mathfrak{G}_1\ \mathfrak{G}_2\ \ldots\ \mathfrak{G}_n$.

It should be noted that the juxtaposition of an empty set of graphs is allowed, too. It yields the empty graph, i.e. $(\emptyset, \top, \emptyset, \emptyset, \emptyset)$.

8

Semantics and Calculus for Formal Alpha Graphs

In this chapter, the semantics and calculus for formal Alpha graphs are provided.

8.1 Semantics for Formal Alpha Graphs

In this section we provide a semantics for formal alpha graphs. As in propositional logic, we assign truth values to the propositional variables by *valuations*. Propositional variables stand for propositions which are simply true or false.

Definition 8.1 (Valuation). *Let* $\mathtt{ff}$ *stand for the truth-value 'false' and* $\mathtt{tt}$ *for the truth-value 'true'. A* valuation *is a mapping* $val : \mathcal{P} \to \{\mathtt{ff}, \mathtt{tt}\}$.

Now we have to extend valuations to graphs by reflection of the meaning of cuts and juxtaposition. This is done close to the so-called *endoporeutic method* of Peirce.

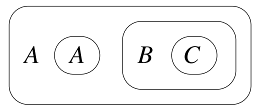
Definition 8.2 (Evaluations). *Let a valuation* val *and let a formal alpha* $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ *be given. We evaluate* $\mathfrak{G}$ *for* val *inductively over* $c \in Cut \cup \{\top\}$*. The evaluation of* $\mathfrak{G}$ *in a context* c *is written* $val \models \mathfrak{G}[c]$*, and it is inductively defined as follows:* $val \models \mathfrak{G}[c] :\Longleftrightarrow$

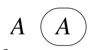
- $val(\kappa(v)) = \mathtt{tt}$ *for each* $v \in V \cap area(c)$ *(vertex condition), and*
- $val \not\models \mathfrak{G}[c']$ *for each* $c' \in Cut \cap area(c)$ *(cut condition: iteration over* $Cut \cup \{\top\}$*)*

For $val \models \mathfrak{G}[\top]$ *we write* $val \models \mathfrak{G}$ *and say that* $\mathfrak{G}$ *is* VALID FOR val *resp.* val IS A MODEL *for* $\mathfrak{G}$*. If we have graphs* $\mathfrak{G}_1$, $\mathfrak{G}_2$ *such that* $val \models \mathfrak{G}_2$ *for each valuation* val*, we write* $\mathfrak{G}_1 \models \mathfrak{G}_2$.

8.2 Some Remarks to the Calculus

Before we come to the formal definition of the rules in the next section, some further scrutiny of the rules, particularly iteration and deiteration, is helpful. In order to do that, consider the following graph (to improve the readability of the examples, we use the letters 'A,B,C' as propositional variables):

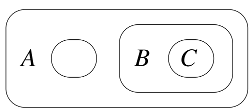
$\mathfrak{G} :=$ 

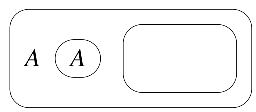
The graph $\mathfrak{G}$ contains the subgraph 

, which is placed in a negative context. As the subgraph alway evaluates to $\mathtt{ff}$, it is easy to see that the whole graph alway evaluates to $\mathtt{tt}$, i.e., it represents a tautology.

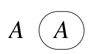
We start out investigations with the rules 'erasure' and 'insertion'. The erasure-rule allows to erase a subgraph from a positive context, the insertion-rule allows to insert a subgraph into a negative context. In this sense, the rules are mutually dual, and it is sufficient to investigate the erasure-rules only.

Possible results of the erasure-rule, applied to $\mathfrak{G}$, are

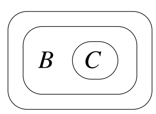


and


.

But erasing the subgraph 

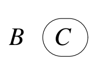
is not allowed. This would yield



,

which may evaluate to $\mathtt{ff}$ (e.g. for $val(B) = \mathtt{ff}$), thus erasing a subgraph from a negative context may lead from a true premise to a false conclusion. Analogously, inserting a subgraph into a positive context is not allowed, neither.

The erasure-rule removes information from a graph, i.e., it is a *generalizing* rule. For this reason, in contrast to the rules iteration, deiteration or double cut, it may only be carried out in one direction. This cannot be seen with $\mathfrak{G}$ (because it represents a tautology), but an example for an application of the erasure-rule which may not be reversed can easily be found. Below, the right graph is obtained from the left with the erasure-rule, and the left graph is not semantically implied by the right graph.

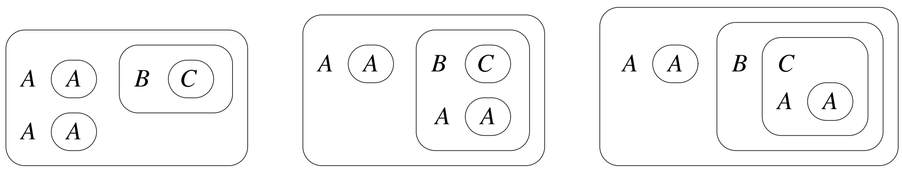


$\overset{\text{era}}{\vdash}$ B .

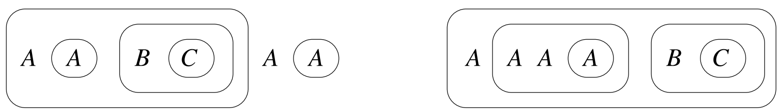
Analogously, the insertion-rule may only be carried out in one direction too.

Next we consider the iteration- and deiteration-rule. The iteration-rule allows to add redundant copies of the subgraph to the graph, the deiteration-rule in turn allows to remove such redundant copies. For this reason, it is sufficient to discuss only the iteration-rule.

For the iteration-rule, it is crucial that if a redundant copy of a subgraph is added to a graph, this redundant copy may not be placed in arbitrary context. It may only be placed in the same or a deeper nested context of the subgraph, and this context is not allowed to be a cut of the subgraph. That is, the following three graphs are the possible applications of the iteration-rule to the graph $\mathfrak{G}$, when the subgraph  is iterated:



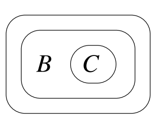
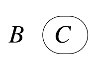
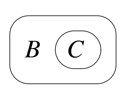
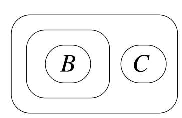
Similar to $\mathfrak{G}$, it is easy to check that all three graphs always evaluate to $\texttt{tt}$. On the other hand, if the subgraph is in another contexts, we may obtain graphs which are no more tautologous. The next two graphs are obtained from a wrong application of the iteration-rule:

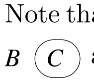
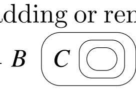


The first graph is obtained by an application of the subgraph into a *higher* context, namely the sheet of assertion. In contrast to $\mathfrak{G}$, this graph always evaluates to $\texttt{ff}$, thus this application of the rule is a fault. The second graph is obtained by an application of the subgraph into a cut of the subgraph. If we assign the truth-value $\texttt{tt}$ to the propositional variables A, B, C, the whole graph evaluates to $\texttt{ff}$. Thus this application of the rule may lead from a true premise to wrong conclusion, i.e., it is a fault, too.

The iteration-rule does not add information to a graph, neither it removes information. For this reason, it may be carried out in both directions (the opposite direction is exactly the deiteration-rule).

The same holds for the double-cut rule, which adds or removes double negations. Here are three pairs of alpha graphs which can mutually be derived from each other with the double cut rule.

 and  ,  and  ,

Note that adding or removing a empty double cut is possible, too. For example,  and  can mutually be derived from each other as well.

8.3 Calculus for Alpha Graphs

In this section we will provide a calculus for formal alpha graphs. For the sake of intelligibility, the whole calculus is first described using common spoken language. After this, the rules are described in a mathematically precise manner. The calculus we present here should be understood as a *diagrammatic calculus*, i.e., all rules can be carried out by manipulating the *diagrams* of formal alpha graphs.

Definition 8.3 (Calculus for Formal Alpha Graphs). *The calculus for formal alpha graphs over $\mathcal{P}$ consists of the following rules:*

- **erasure (era)**

  In positive contexts, any subgraph may be erased.

- **insertion (ins)**

  In negative contexts, any subgraph may be inserted.

- **iteration (it)**

  Let $\mathfrak{G}_0 := (V_0, \top_0, Cut_0, area_0, \kappa_0)$ be a subgraph of $\mathfrak{G}$ and let $c \leq ctx(\mathfrak{G}_0)$ be a context such that $c \notin Cut_0$. Then a copy of $\mathfrak{G}_0$ may be inserted into c.

- **deiteration (deit)**

  If $\mathfrak{G}_0$ is a subgraph of $\mathfrak{G}$ which could have been inserted by rule of iteration, then it may be erased.

- **double cut (dc)**

  Double cuts (two cuts c_1, c_2 with $area(c_1) = \{c_2\}$) may be inserted or erased.

These rules have to be written down mathematically. Here are the appropriate mathematical definitions:

- **erasure and insertion**

  We first provide a general definition for inserting and erasing a subgraph.

  Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a graph which contains the subgraph $\mathfrak{G}_0 := (V_0, \top_0, Cut_0, area_0, \kappa_0)$. Let $\mathfrak{G}' := (V', \top', Cut', area', \kappa')$ be defined as follows:

  – $V' := V \backslash V_0$, $\top' := \top$ and $Cut' := Cut \backslash Cut_0$,

  – $area'(d) := \begin{cases} area(d) & d \neq \top_0 \\ area(d) \backslash (V_0 \cup Cut_0) & d = \top_0 \end{cases}$.

  – $\kappa' := \kappa\big|_{V'}$

Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING THE SUBGRAPH $\mathfrak{G}_0$ FROM THE CONTEXT $\top_0$, and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by INSERTING THE GRAPH $\mathfrak{G}_0$ INTO THE CONTEXT $\top_0$. The rules 'erasure' and 'insertion' are restrictions of the general definition as follows: Let $\mathfrak{G}$ be a graph and let $\mathfrak{G}_0$ be a subgraph of $\mathfrak{G}$ with $c := ctx(\mathfrak{G}_0)$, and let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by erasing $\mathfrak{G}_0$ from the context c. If c is positive, then $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING $\mathfrak{G}_0$ FROM A POSITIVE CONTEXT, and if c is negative, than $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by INSERTING $\mathfrak{G}_0$ INTO A NEGATIVE CONTEXT.

- **iteration and deiteration**

  Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a graph which contains the subgraph $\mathfrak{G}_0 := (V_0, \top_0, Cut_0, area_0, \kappa_0)$, and let $c \leq \top_0$ be a context with $c \notin Cut_0$.

  Let $\mathfrak{G}' := (V', \top', Cut', area', \kappa')$ be the following graph:

  – $V' := V \times \{1\} \cup V_0 \times \{2\}$, $\top' := \top$ and $Cut' := Cut \times \{1\} \cup Cut_0 \times \{2\}$.

  – $area'$ is defined as follows:

    For $(d, i) \in Cut' \cup \{\top'\}$ and $d \neq c$ let $area'((d, i)) := area(d) \times \{i\}$, and $area'((c, 1)) := area(c) \times \{1\} \cup area_0(\top_0) \times \{2\}$.

  – $\kappa'((k, i)) := \kappa(k)$ for all $(k, i) \in V'$

  Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ITERATING THE SUBGRAPH $\mathfrak{G}_0$ INTO THE CONTEXT c and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by DEITERATING THE SUBGRAPH $\mathfrak{G}_0$ FROM THE CONTEXT c.

- **double cuts**

  Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a graph and $c_1, c_2 \in Cut$ with $area(c_1) = \{c_2\}$. Let $c_0 := ctx(c_1)$ and set $\mathfrak{G}' := (V, \top, Cut', area', \kappa)$ with

  – $Cut' := Cut \backslash \{c_1, c_2\}$

  – $area'(d) := \begin{cases} area(d) \text{ for } d \neq c_0 \\ area(c_0) \cup area(c_2) \text{ for } d = c_0 \end{cases}.$

  Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING THE DOUBLE CUTS c_1, c_2 and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by INSERTING THE DOUBLE CUTS c_1, c_2.

Based on the calculus, we can now define the syntactical entailment relation.

Definition 8.4 (Syntactical Entailment Relation). *Let $\mathfrak{G}_a$, $\mathfrak{G}_b$ be two graphs. Then $\mathfrak{G}_b$* CAN BE DERIVED FROM *$\mathfrak{G}_a$ (which is written $\mathfrak{G}_a \vdash \mathfrak{G}_b$), if there is a finite sequence $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ with $\mathfrak{G}_a = \mathfrak{G}_1$ and $\mathfrak{G}_b = \mathfrak{G}_n$ such that each $\mathfrak{G}_{i+1}$ is derived from $\mathfrak{G}_i$ by applying one of the rules of the calculus. The sequence is called* A PROOF FOR *$\mathfrak{G}_a \vdash \mathfrak{G}_b$. Two graphs $\mathfrak{G}_1, \mathfrak{G}_2$ with $\mathfrak{G}_1 \vdash \mathfrak{G}_2$ and $\mathfrak{G}_2 \vdash \mathfrak{G}_1$ are said to be* PROVABLY EQUIVALENT.

If $\mathfrak{H} := \{\mathfrak{G}_i \mid i \in I\}$ is a (possibly empty) set of graphs, then A GRAPH $\mathfrak{G}$ CAN BE DERIVED FROM $\mathfrak{H}$ *if there is a finite subset $\{\mathfrak{G}_1, \ldots, \mathfrak{G}_n\} \subseteq \mathfrak{H}$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \vdash \mathfrak{G}$ (remember that $\mathfrak{G}_1 \ldots \mathfrak{G}_n$ is the juxtaposition of $\mathfrak{G}_1, \ldots, \mathfrak{G}_n$).*

8.4 Some Simple Theorems

In [Pei35] Peirce provided 16 useful transformation rules for EGs which he derived from his calculus. These rules are logical metalemmata in the sense that they show some *schemata* for proofs with EGs, i.e., they are derived 'macro'-rules. In this section we provide the formal alpha graph versions for two of these transformation rules. We start with a (weakened) version of the first transformation rule of Peirce.

Lemma 8.5 (Reversion Theorem). *Let $\mathfrak{G}_a$ and $\mathfrak{G}_b$ be two formal alpha graphs. Then we have $\mathfrak{G}_a \vdash \mathfrak{G}_b$ if and only if $\boxed{\mathfrak{G}_b} \vdash \boxed{\mathfrak{G}_a}$.*

Proof: Let $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ with $\mathfrak{G}_1 = \mathfrak{G}_a$ and $\mathfrak{G}_b = \mathfrak{G}_n$ be a proof for $\mathfrak{G}_a \vdash \mathfrak{G}_b$. Then, due to the symmetry of the calculus, ($\boxed{\mathfrak{G}_n}$, $\boxed{\mathfrak{G}_{n-1}}$, ..., $\boxed{\mathfrak{G}_1}$) is a proof for $\boxed{\mathfrak{G}_b} \vdash \boxed{\mathfrak{G}_a}$. The inverse direction holds as well. □

All rules in the calculus which are applied in a context only depend on whether the context is positive or negative. In particular if a proof for $\mathfrak{G}_a \vdash \mathfrak{G}_b$ is given, this proof can be carried out in arbitrary positive contexts. Together with the previous lemma, this yields the following lemma, which is a combination of the (full) first and the sixth transformation rule of Peirce. It can also be found in [Sow97a] (from where we adopted the name of the theorem).

Lemma 8.6 (Cut-And-Paste-Theorem). *Let $\mathfrak{G}_a \vdash \mathfrak{G}_b$ for two formal alpha graphs $\mathfrak{G}_a$, $\mathfrak{G}_b$, let $\mathfrak{G}$ be a further graph. Then we have:*

- *If $\mathfrak{G}_a$ is a subgraph of $\mathfrak{G}$ in a pos. context, then $\mathfrak{G}_a$ may be replaced by $\mathfrak{G}_b$.*
- *If $\mathfrak{G}_b$ is a subgraph of $\mathfrak{G}$ in a neg. context, then $\mathfrak{G}_b$ may be replaced by $\mathfrak{G}_a$.*

In particular we have that derivable graphs $\mathfrak{G}_0$ (i.e., graphs with $\vdash \mathfrak{G}_0$) can be inserted into arbitrary contexts of arbitrary graphs.

From this lemma we obtain the formal alpha graph version of the well known deduction theorem.

Lemma 8.7 (Deduction Theorem). *Let $\mathfrak{G}_a$, $\mathfrak{G}_b$ be formal alpha graphs. Then we have*

$$\mathfrak{G}_a \vdash \mathfrak{G}_b \quad \Longleftrightarrow \quad \vdash \boxed{\mathfrak{G}_a \ \boxed{\mathfrak{G}_b}}$$

Proof: We show both directions separately.

'$\Longrightarrow$': $\overset{\text{dc}}{\vdash} \boxed{\quad \boxed{\phantom{\mathfrak{G}_a}}} \overset{\text{ins}}{\vdash} \boxed{\mathfrak{G}_a \ \boxed{\phantom{\mathfrak{G}_a}}} \overset{\text{it}}{\vdash} \boxed{\mathfrak{G}_a \ \boxed{\mathfrak{G}_a}} \overset{\text{L. 8.6}}{\vdash} \boxed{\mathfrak{G}_a \ \boxed{\mathfrak{G}_b}}$

'$\Longleftarrow$': $\quad \mathfrak{G}_a \overset{\text{L. 8.6}}{\vdash} \mathfrak{G}_a \; \boxed{\mathfrak{G}_a \; \boxed{\mathfrak{G}_b}} \overset{\text{deit}}{\vdash} \mathfrak{G}_a \; \boxed{\quad \boxed{\mathfrak{G}_b}} \overset{\text{dc}}{\vdash} \mathfrak{G}_a \quad \mathfrak{G}_b \overset{\text{era}}{\vdash} \mathfrak{G}_b$ □

The following lemma is quite obvious:

Lemma 8.8. *Let $\mathfrak{G}$, $\mathfrak{G}_a$ and $\mathfrak{G}_b$ be formal alpha graphs with $\mathfrak{G} \vdash \mathfrak{G}_a$ and $\mathfrak{G} \vdash \mathfrak{G}_b$. Then $\mathfrak{G} \quad \vdash \quad \mathfrak{G}_a \; \mathfrak{G}_b$.*

Proof: $\qquad \mathfrak{G} \overset{\text{it}}{\vdash} \mathfrak{G} \; \mathfrak{G} \overset{\text{L. 8.6}}{\vdash} \mathfrak{G}_a \; \mathfrak{G} \overset{\text{L. 8.6}}{\vdash} \mathfrak{G}_a \; \mathfrak{G}_b$ □

9

Soundness and Completeness

In this chapter we will show that the rules we presented in Chapter 8 are sound and complete with respect to the given semantics. In the first section, we will prove the soundness of the calculus, in the next section we will prove its completeness.

9.1 Soundness

Most of the rules modify only the area of one specific context c (for example, the rule 'erasure' removes a subgraph from the area of a positive context). If a graph $\mathfrak{G}'$ is derived from a graph $\mathfrak{G}$ by applying one of these rules (i.e., by modifying a context c in the graph $\mathfrak{G}$), $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for c. As it has to be shown that no rule can transform a valid graph into a nonvalid one, the following theorem is the basis for proving the soundness of most rules.

Theorem 9.1 (Main Soundness Lemma). *Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ and $\mathfrak{G}' := (V', \top', Cut', area', \kappa')$ be formal alpha graphs, let $f = f_V \dot{\cup} f_{Cut}$ be an isomorphism from $\mathfrak{G}$ to $\mathfrak{G}'$ except for the contexts $c \in Cut \cup \{\top\}$ and $c' \in Cut' \cup \{\top'\}$. Let $val : \mathcal{P} \to \{\mathtt{ff}, \mathtt{tt}\}$ be a valuation. Let $P(d)$ be the following property for Cuts $d \in Cut \cup \{\top\}$:*

- *If d is positive and $val \models \mathfrak{G}[d]$, then $val \models \mathfrak{G}'[f(d)]$, and*
- *If d is negative and $val \not\models \mathfrak{G}[d]$, then $val \not\models \mathfrak{G}'[f(d)]$.*

If P holds for c, then P holds for each $d \in Cut \cup \{\top\}$ with $d \not< c$. In particular follows $v \models \mathfrak{G}'$ from $v \models \mathfrak{G}$.

Proof: We set $D := \{d \in Cut \cup \{\top\} \mid d \not< c\}$. D is a tree such that for each $d \in D$ with $d \neq c$ and each $e \in Cut \cup \{\top\}$ with $e < d$ we have $e \in D$. For this reason we can carry out the proof by induction over D. As c satisfies P, it is sufficient to carry out the induction step for $d \neq c$. So let $d \in D$, $d \neq c$ be a context such that $P(e)$ holds for all cuts $e \in area(d) \cap Cut$.

First Case: d is positive and $val \models \mathfrak{G}[d]$.

We have to check the vertex- and cut-conditions for $f(d)$. We start with the vertex conditions for $f(d)$, i.e., for vertices $v' \in V'$ with $ctx'(v') = f(d)$.

For each $v \in V$ with $ctx(v) = d$, it holds $\kappa(v) = \kappa'(f(v))$, hence

$$val(\kappa(v)) = \mathtt{tt} \Longleftrightarrow val(\kappa'(f(v))) = \mathtt{tt}.$$

As f_V is a bijection from $area(d) \cap V$ to $area'(f(d)) \cap V'$, we gain the following: All vertex conditions in d hold iff all vertex conditions in $f(d)$ hold.

As we have $val \models \mathfrak{G}[d]$, we get that $val \not\models \mathfrak{G}[e]$ for all cuts $e \in area(d)$. These cuts are negative and are mapped bijectively to the cuts $e' \in area(f(d))$. As they are negative, we conclude from the induction hypothesis or the presupposition (for $e = c$) that $val \not\models \mathfrak{G}'[f(e)]$ for all cuts $e \in area(d)$, i.e., $val \not\models \mathfrak{G}'[e']$ for all cuts $e' \in area'(f(d))$.

As we have checked all vertex- and cut-conditions for $f(d)$, we get $val \models \mathfrak{G}'[f(d)]$.

Second Case: d is negative and $val \not\models \mathfrak{G}[d]$.

This is shown analogously to the first case. □

With this lemma, we can prove the correctness of the rules. We start with the soundness of the rules 'iteration' and 'deiteration'.

Lemma 9.2 (Iteration and Deiteration are Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two formal alpha graphs, val is a valuation with $val \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by applying one of the rules 'iteration' or 'deiteration', then $val \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}_0 := (V_0, \top_0, Cut_0, area_0, \kappa_0)$ be the subgraph of $\mathfrak{G}$ which is iterated into the context $c \leq ctx(\mathfrak{G}_0)$, $c \notin Cut_0$. We use the mathematical notation which was given in Section 8.3. In particular, $(c, 1)$ is the context in $\mathfrak{G}'$ which corresponds to the context c in $\mathfrak{G}$. There are two cases to consider:

First Case: $val \models \mathfrak{G}_0$. From this we conclude $val \models \mathfrak{G}[c] \Longleftrightarrow val \models \mathfrak{G}'[(c,1)]$. As $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for $c \in Cut \cup \{\top\}$ and $(c,1) \in Cut' \cup \{\top'\}$, Lemma 9.1 can be applied now. This yields

$$val \models \mathfrak{G} \Longleftrightarrow val \models \mathfrak{G}' \ . \qquad (*)$$

Second Case: $val \not\models \mathfrak{G}_0$. This yields $val \not\models \mathfrak{G}[\top_0]$ and $val \not\models \mathfrak{G}'[(\top_0, 1)]$. As $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for $\top_0 \in Cut \cup \{\top\}$ and $(\top_0, 1) \in Cut' \cup \{\top'\}$, Lem. 9.1 can be applied now. This yields again $(*)$.

The direction '$\Longrightarrow$' of $(*)$ yields the correctness of the iteration-rule. The opposite direction '$\Longleftarrow$' of $(*)$ yields the correctness of the deiteration-rule. □

Lemma 9.3 (Erasure and Insertion are Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two formal alpha graphs, v is a valuation with $val \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by applying one of the rules 'erasure' or 'insertion', then $val \models \mathfrak{G}'$.*

Proof: We start with 'erasure'. Let $\mathfrak{G}_0 := (V_0, \top_0, Cut_0, area_0, \kappa_0)$ be the subgraph which is erased. $\mathfrak{G}_0$ is erased from the area of the positive context $c := \top_0$. Obviously, if $val \models \mathfrak{G}[c]$, then $val \models \mathfrak{G}'[c]$. Furthermore $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for $c \in Cut \cup \{\top\}$ and $c \in Cut' \cup \{\top'\}$, hence Lemma 9.1 can be applied now. This yields $val \models \mathfrak{G}'$.

The soundness of the insertion-rule is proven analogously. □

Lemma 9.4 (Double Cut is Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two formal alpha graphs, val is a valuation with $val \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by applying the rule 'double cut', then $val \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}$ be derived from $\mathfrak{G}'$ by erasing two cuts c_1, c_2 with $area(c_1) = \{c_2\}$. We set $c := ctx(c_1)$. We want to apply Lemma 9.1 and therefore have to show that property P of Lemma 9.1 is valid for $c \in \mathfrak{G}$ and $c \in \mathfrak{G}'$. We have

$$area'(c) = (area(c) \cup area(c_2)) \backslash \{c_1\} \qquad (*)$$

With $(*)$ we get

$$\begin{aligned}
val \models \mathfrak{G}[c] &\overset{\text{Def} \models}{\iff} val \text{ fulfills all vertex- and cut-conditions of } area(c) \\
&\iff val \text{ fulfills all v.-, c.-cond. of } area(c)\backslash\{c_1\}, \text{ and } v \not\models \mathfrak{G}[c_1] \\
&\iff val \text{ fulfills all v.-, c.-cond. of } area(c)\backslash\{c_1\}, \text{ and } val \models \mathfrak{G}[c_2] \\
&\overset{(*)}{\iff} val \text{ fulfills all v.-, c.-cond. of } area'(c) \\
&\overset{\text{Def} \models}{\iff} val \models \mathfrak{G}'[c]
\end{aligned}$$

Now Lemma 9.1 yields that we have $val \models \mathfrak{G} \iff val \models \mathfrak{G}'$. □

From the preceding lemmata we obtain the soundness of the calculus.

Theorem 9.5 (Soundness of the Alpha-Calculus). *A set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of formal alpha graphs over $\mathcal{A}$ satisfies*

$$\mathfrak{H} \vdash \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \models \mathfrak{G} \quad .$$

Proof: Let $\mathfrak{H} \vdash \mathfrak{G}$. Then there are $\mathfrak{G}_1, \ldots, \mathfrak{G}_n$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \vdash \mathfrak{G}$. From the preceding Lemmata 9.3, 9.4 and 9.2 we conclude

$$\mathfrak{G}_1 \ldots \mathfrak{G}_n \models \mathfrak{G} \qquad (9.1)$$

Now let val be a valuation $val \models \mathfrak{H}$, i.e., $val \models \mathfrak{G}'$ for each $\mathfrak{G}' \in \mathfrak{H}$. Then we have particularly $val \models \mathfrak{G}_i$ for each $1 \leq i \leq n$, thus $val \models \mathfrak{G}_1 \ldots \mathfrak{G}_n$. Now Eqn. 9.1 yields $val \models \mathfrak{G}$. □

9.2 Completeness

As the empty sheet of assertion is always true, the graph $\boxed{}$ is always false.[1] This leads to the following definition and lemmata.

Definition 9.6 (Consistent Sets of Alpha Graphs). *A set $\mathfrak{H}$ of formal alpha graphs is called* CONSISTENT *if $\mathfrak{H} \vdash \boxed{}$ does not hold. A formal alpha graph $\mathfrak{G}$ is called* CONSISTENT *if $\{\mathfrak{G}\}$ is consistent.*

Lemma 9.7 (Consistency Lemma 1). *A set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of formal alpha graphs is not consistent if and only if $\mathfrak{H} \vdash \mathfrak{G}'$ for every formal alpha graph $\mathfrak{G}'$.*

Proof: Only '$\Longrightarrow$' has to be shown. Let $\mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H}$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \vdash \boxed{}$. We conclude:

$$\mathfrak{G}_1 \ldots \mathfrak{G}_n \quad \vdash \quad \boxed{} \quad \overset{\text{ins.}}{\vdash} \quad \boxed{\boxed{\mathfrak{G}'}} \quad \overset{\text{dc}}{\vdash} \quad \mathfrak{G}' \quad \overset{\text{Def.}\vdash}{\Longrightarrow} \quad \mathfrak{H} \vdash \mathfrak{G}' \qquad \square$$

Lemma 9.8 (Consistency Lemma 2). *Let $\mathfrak{H} \cup \{\mathfrak{G}\}$ be a set of formal alpha graphs. Then we have*

$$\mathfrak{H} \vdash \mathfrak{G} \iff \mathfrak{H} \cup \{\boxed{\mathfrak{G}}\} \vdash \boxed{} \quad \textit{and} \quad \mathfrak{H} \vdash \boxed{\mathfrak{G}} \iff \mathfrak{H} \cup \{\mathfrak{G}\} \vdash \boxed{} \quad .$$

In particular, for two formal alpha graphs $\mathfrak{G}_1$ and $\mathfrak{G}_2$, we have

$$\mathfrak{G}_1 \vdash \mathfrak{G}_2 \iff \mathfrak{G}_1 \boxed{\mathfrak{G}_2} \vdash \boxed{} \quad \textit{and} \quad \mathfrak{G}_1 \vdash \boxed{\mathfrak{G}_2} \iff \mathfrak{G}_1 \ \mathfrak{G}_2 \vdash \boxed{}$$

Proof of the first equivalence (the second is shown analogously): We have

$$\begin{aligned}
\mathfrak{H} \vdash \mathfrak{G} &\overset{\text{Def.}\vdash}{\iff} \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \mathfrak{G}_1 \ldots \mathfrak{G}_n \vdash \mathfrak{G} \\
&\overset{\text{T. 8.7}}{\iff} \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \vdash \boxed{\mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}}} \\
&\overset{\text{dc}}{\iff} \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \vdash \boxed{\mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}} \boxed{\boxed{}}} \\
&\overset{\text{T. 8.7}}{\iff} \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}} \vdash \boxed{} \\
&\overset{\text{Def.}\vdash}{\iff} \mathfrak{H} \cup \{\boxed{\mathfrak{G}}\} \vdash \boxed{} \qquad \square
\end{aligned}$$

Consistent sets of graphs can be extended to maximal (with respect to $\subseteq$) consistent sets graphs, which have canonically given valuations satisfying them. This will be elaborated with the next lemmata.

[1] Peirce treated EGs as *judgments*, i.e., as propositions which are asserted to be true in some context. A 'graph' which is never true in any context cannot be asserted and is therefore in Peirce's view not a graph. For this reason Peirce called the sketched graph (and all equivalent graphs) 'pseudograph' (see [Pei35]).

Lemma 9.9 (Properties of Maximal Consistent Sets). *Let $\mathfrak{H}$ be a maximal (with respect to $\subseteq$) consistent set of graphs. Then:*

1. *Either $\mathfrak{H} \vdash \mathfrak{G}$ or $\mathfrak{H} \vdash \boxed{\mathfrak{G}}$ for each formal alpha graph $\mathfrak{G}$.*
2. *$\mathfrak{H} \vdash \mathfrak{G} \iff \mathfrak{G} \in \mathfrak{H}$ for each formal alpha graph $\mathfrak{G}$.*
3. *$\mathfrak{G}_1 \quad \mathfrak{G}_2 \in \mathfrak{H} \iff \mathfrak{G}_1 \in \mathfrak{H}$ and $\mathfrak{G}_2 \in \mathfrak{H}$ for all formal alpha graphs $\mathfrak{G}_1, \mathfrak{G}_2$.*

Proof: It is easy to see that $\mathfrak{G} \boxed{\mathfrak{G}} \vdash \boxed{\phantom{\mathfrak{G}}}$. Hence if $\mathfrak{H}$ is consistent, $\mathfrak{H} \vdash \mathfrak{G}$ and $\mathfrak{H} \vdash \boxed{\mathfrak{G}}$ cannot hold both. Now we can prove all propositions of this lemma:

1. Assume $\mathfrak{H} \nvdash \mathfrak{G}$ for a graph $\mathfrak{G}$. Lem. 9.8 yields that $\mathfrak{H} \cup \{\boxed{\mathfrak{G}}\}$ is consistent. As $\mathfrak{H}$ is maximal, we conclude that $\boxed{\mathfrak{G}} \in \mathfrak{H}$, hence we have $\mathfrak{H} \vdash \boxed{\mathfrak{G}}$.
2. Let $\mathfrak{H} \vdash \mathfrak{G}$. As $\mathfrak{H}$ is consistent, we get that $\mathfrak{H} \nvdash \boxed{\mathfrak{G}}$. So Lem. 9.8 yields that $\mathfrak{H} \cup \{\mathfrak{G}\}$ is consistent. As $\mathfrak{H}$ is maximal, we conclude $\mathfrak{G} \in \mathfrak{H}$.
3. Follows immediately from 1. and 2. □

Consistent sets of formal alpha graphs can be extended to maximal consistent sets of formal alpha graphs. This is done as usual in logic.

Lemma 9.10 (Extending Consistent Sets to Maximal Sets). *Let $\mathfrak{H}$ be a consistent set of formal alpha graphs. Then there is a maximal set $\mathfrak{H}'$ of formal alpha graphs with $\mathfrak{H}' \supseteq \mathfrak{H}$.*

Proof: Let $\mathfrak{G}_1, \mathfrak{G}_2, \mathfrak{G}_3, \ldots$ be an enumeration of all formal alpha graphs.[2] We define inductively a set of graphs $\mathfrak{H}_i$ for each $i \in \mathbb{N}$. We start with setting $\mathfrak{H}_1 := \mathfrak{H}$. Assume now that $\mathfrak{H}_i \supseteq \mathfrak{H}$ is defined and consistent.

If $\mathfrak{H}_i \vdash \boxed{\mathfrak{G}_i}$ does not hold, then $\mathfrak{H}_{i+1} := \mathfrak{H}_i \cup \{\mathfrak{G}_i\}$ is consistent due to Lemma 9.8.

Otherwise, if $\mathfrak{H}_i \vdash \boxed{\mathfrak{G}_i}$ holds, then $\mathfrak{H}_{i+1} := \mathfrak{H}_i \cup \{\boxed{\mathfrak{G}_i}\}$ is consistent.

Now $\mathfrak{H}' := \bigcup_{n\in\mathbb{N}} \mathfrak{H}'_n$ is a consistent maximal graph set with $\mathfrak{H}' \supseteq \mathfrak{H}$. □

Maximal consistent set of graphs have canonically given valuations, i.e., models.

Theorem 9.11 (Valuations for Maximal Consistent Sets). *Let $\mathfrak{H}$ be a maximal consistent set of formal alpha graphs. Then there exists a canonically given valuation val such that val $\models \mathfrak{G}$ for each graph $\mathfrak{G} \in \mathfrak{H}$.*

[2] Remember that we assumed that $\mathcal{A}$ is finite, hence we have only countably many formal alpha graphs (more precisely: countably many isomorphism-classes of formal alpha graphs). If we allowed infinite alphabets, the proof could be carried out with an application of the prim ideal theorem or of (the stronger) Zorn's Lemma.

Proof: Let us first define a graph which states the propositional variable P_i. We set

$$\mathfrak{G}(P_i) := (\{v\}, \top, \emptyset, \emptyset, \{(v, P_i)\}) \text{ with an arbitrary vertex } v\,.$$

Now let $val : \mathcal{P} \to \{\mathtt{ff}, \mathtt{tt}\}$ be defined as follows:

$$val(P_i) := \mathtt{tt} : \iff \mathfrak{H} \vdash \mathfrak{G}(P_i)$$

Now let $\mathfrak{G}' := (V, \top, Cut, area, \kappa)$ be a formal alpha graph. We show

$$val \models \mathfrak{G}'[c] \iff \mathfrak{H} \vdash \mathfrak{G}'[c] \tag{9.2}$$

for each $c \in Cut \cup \{\top\}$. The proof is done by induction over $Cut \cup \{\top\}$. So let $c \in Cut \cup \{\top\}$ be a cut such that (9.2) holds for each $d < c$. We have:

$$\begin{aligned}
val \models \mathfrak{G}'[c] &\overset{\text{Def. evaluation}}{\iff} && val(\kappa(v)) = \mathtt{tt} \text{ for each } v \in V \cap area(c) \\
& && \text{and } val \not\models \mathfrak{G}[d] \text{ for each } d \in Cut \cap area(d) \\
&\overset{\text{Def. val and Ind.Hyp.}}{\iff} && \mathfrak{H} \vdash \mathfrak{G}(\kappa(v)) \text{ for each } v \in V \cap area(c) \\
& && \text{and } \mathfrak{H} \not\vdash \mathfrak{G}'[d] \text{ for each } d \in Cut \cap area(d) \\
&\overset{\text{L. 9.9}}{\iff} && \mathfrak{G}(\kappa(v)) \in \mathfrak{H} \text{ for each } v \in V \cap area(c) \\
& && \text{and } \boxed{\mathfrak{G}'[d]} \in \mathfrak{H} \text{ for each } d \in Cut \cap area(d) \\
&\overset{\text{L. 9.9}}{\iff} && \mathfrak{G}[c] \in \mathfrak{H}
\end{aligned}$$

As we have $\mathfrak{G} = \mathfrak{G}[\top]$, (9.2) yields for $c := \top$

$$val \models \mathfrak{G}' \iff \mathfrak{H} \vdash \mathfrak{G}'$$

thus we are done. □

Now we are prepared to prove the completeness of the calculus.

Theorem 9.12 (Completeness of the Calculus). *A set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of formal alpha graphs over an alphabet $\mathcal{A}$ satisfies*

$$\mathfrak{H} \models \mathfrak{G} \implies \mathfrak{H} \vdash \mathfrak{G}$$

Proof: Each valuation which satisfies $\mathfrak{H}$ satisfies $\mathfrak{G}$ as well, i.e., $\mathfrak{H} \cup \left\{\boxed{\mathfrak{G}}\right\}$ has no model. Thus, according to Thm. 9.11, $\mathfrak{H} \cup \left\{\boxed{\mathfrak{G}}\right\}$ is not consistent. That is, there are $\mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H}$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}} \vdash \boxed{}$. The Deduction Theorem Thm. 8.7, applied to the two graphs $\mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}}$ and $\boxed{}$, yields

$$\vdash \quad \boxed{\mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}} \boxed{\boxed{}}} \quad .$$

With the double-cut rule we conclude:

$$\vdash \quad \boxed{\mathfrak{G}_1 \ldots \mathfrak{G}_n \boxed{\mathfrak{G}}} \quad .$$

Now we apply the Deduction Theorem in the opposite direction to $\mathfrak{G}_1 \ldots \mathfrak{G}_n$ and $\boxed{\mathfrak{G}}$, which yields:

$$\mathfrak{G}_1 \ldots \mathfrak{G}_n \quad \vdash \quad \mathfrak{G}$$

Thus we have $\mathfrak{H} \vdash \mathfrak{G}$. □

10

Translation to Propositional Logic

In the last chapters, the Alpha part of EGs is developed as a mathematical logic. This includes definitions for a syntax, semantics, and a calculus, as well as a proof that the calculus is adequate, i.e., sound and complete. In formal Alpha graphs, we have propositional variables and the possibility to construct the negation of a given graph, as well as the conjunction of some given graphs. Thus, it is somewhat evident that the system of Alpha graphs correspond to propositional logic.

This correspondence will be worked out in this section by providing translations from Alpha graphs to formulas of propositional logic, and vice versa, and by showing that these translations are meaning-preserving. In the Beta-part of this treatise, corresponding the translations between Beta graphs and formulas of first order logic are provided. In Beta, these translations turn out to be much more important (as the completeness-proof in Beta refers to them), and the definition and understanding of these translations will be discussed in more detail. For Alpha, the definitions and proofs in this chapter are straight-forward. For a deeper discussion of translations, I refer to the Beta-part.

We start with the definition of formulas of propositional logic. For our purpose, it is convenient to consider that fragment which uses only the junctors '$\neg$' (not) and '$\wedge$' (and). It is well-known that all other junctors can be expressed with these two, i.e., this fragment of propositional logic is functionally complete.

Definition 10.1 (Propositional Logic). *The formulas of propositional formulas (over the set of propositional variables $\mathcal{P}$) is inductively be defined as follows:*

1. *Each propositional variable $P_i \in \mathcal{P}$ is a formula,*
2. *If f' is a formula, then $f := \neg f'$ is a formula, and*
3. *if f_1 and f_2 are formulas, then $f := (f_1 \wedge f_2)$ is a formula.*

The evaluation of formulas for a given valuation is canonically be defined as follows:

Definition 10.2 (Evaluations Of Formulas). *We define the semantical entailment relation $val \models f$ between valuations and formulas inductively over the composition of formulas as follows:*

- *For $P_i \in \mathcal{P}$ we set $val \models P_i \;:\Longleftrightarrow\; val(P_i) = \mathtt{tt}$.*
- *For $f = f_1 \wedge f_2$, we set $val \models f_1 \wedge f_2 \;:\Longleftrightarrow\; val \models f_1$ and $val \models f_2$.*
- *For $f = \neg f'$, we set $val \models \neg f' \;:\Longleftrightarrow\; val \not\models f_1$.*

Now we are prepared to define the translations between the system of Alpha graphs and propositional logic. Both translations are inductively defined.

Translations from graphs to symbolic logic are denoted by the letter Φ, translations in the opposite direction by the letter Ψ.[1] In order to distinguish the mappings from the Alpha part from the mappings which will be defined in the Beta part, they are indexed with the letter α.

We start with the definition of the translation Ψ_α from the system of formal alpha graphs to propositional logic. This definition is straight-forward.

Definition 10.3 ($\boldsymbol{\Psi_\alpha}$). *We define Ψ_α inductively over the composition of formulas.*

- *For $P_i \in \mathcal{P}$, let $\Psi_\alpha(P_i) := (\{v\}, \top, \emptyset, \emptyset, \{v, P_i)\})$. That is, we set:*

$$\Psi_\alpha(P_i) := \quad P_i \qquad .$$

- *For $f \;=\; f_1 \wedge f_2$, let*

$$\Psi_\alpha(f) \;:=\; \Psi_\alpha(f_1) \quad \Psi_\alpha(f_2)$$

  be the juxtaposition of $\Psi_\alpha(f_1)$ and $\Psi_\alpha(f_2)$.

- *For $f \;=\; \neg f'$ with $\Psi_\alpha(f') = (V, \top, Cut, area, \kappa)$, let $c_0 \notin V \cup Cut \cup \{\top\}$ be a new cut, and let $\Psi_\alpha(f) := (V, \top, Cut', area', \kappa)$ with $Cut' := Cut \cup \{c_0\}$ and $area' = area \backslash \{(\top, area(\top))\} \cup \{(c_0, area(\top)), (\top, c_0)\}$. I.e., we set:*

$$\Psi_\alpha(f) \;:=\; \boxed{\Psi_\alpha(f)} \qquad .$$

For the definition of the translation Φ_α, we have to take care that in existential graphs, empty cuts may occur, which have no counterparts in propositional

[1] These letters are used for Sowa's conceptual graphs, which are based on Peirce's existential graphs, too.

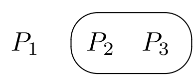
logic ('' is a well-formed graph, but '$\neg$' is not a well-formed formula). We use the following workaround: An empty area (the area of an empty cut or the empty sheet of assertion) is translated to the formula $\neg(P_1 \wedge \neg P_1)$, which is always evaluated to true. The remaining definition of Φ_α is straight-forward.

Definition 10.4 ($\boldsymbol{\Phi_\alpha}$). *Let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a formal alpha graph. Using Lem. 7.6, we assign inductively to each context $c \in Cut \cup \{\top\}$ a formula $\Phi_\alpha(\mathfrak{G}, c)$ as follows:*

If c is an empty context, i.e., $area(\mathfrak{G}, c) = \emptyset$, we set $\Phi_\alpha(\mathfrak{G}, c) := \neg(P_1 \wedge \neg P_1)$. If c is not empty, let $\Phi_\alpha(c)$ be the conjunction of the formulas $\neg\Phi_\alpha(d)$ for cuts $d \in area(c)$ and the formulas $\kappa(v)$ for vertices $v \in area(c)$.

Finally we set $\Phi_\alpha(\mathfrak{G}) := \Phi_\alpha(\mathfrak{G}, \top)$, and the definition of Φ_α is finished.

It should be noted that Φ_α is, strictly speaking, not a function. The phrase 'let $\Phi_\alpha(c)$ be the conjunction of the formulas' does does not specify in which order the formulas $\neg\Phi_\alpha(d)$ for $d \in area(c)$ and $\kappa(v)$ for $v \in area(c)$ have to be composed, neither how brackets are used. Thus, $\Phi_\alpha(\mathfrak{G})$ is only given up the order of subformulas of conjunctions. For example, the graph

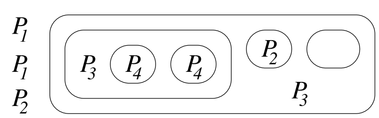


can be translated to the following formulas:

$$P_1 \wedge \neg(P_2 \wedge P_3)\ ,\ P_1 \wedge \neg(P_3 \wedge P_2)\ ,\ \neg(P_2 \wedge P_3) \wedge P_1\ , \text{ and } \neg(P_3 \wedge P_2) \wedge P_1\ .$$

But, as conjunction is (in a semantical sense) an associative and commutative operation, all possible translations of a graph are semantically equivalent, thus we consider Φ_α as a mapping which assigns one formula to each graph.

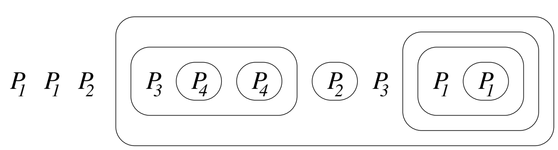
To provide a further example, we consider the graph of Fig. 7.1, i.e.



This graph is translated to:

$$P_1 \wedge P_1 \wedge P_2 \wedge \neg(\neg(P_3 \wedge \neg P_4 \wedge \neg P_4) \wedge \neg P_2 \wedge P_3 \wedge \neg\neg(P_1 \wedge \neg P_1))$$

and, vice versa, this formula is translated back into the following graph:



The next theorem shows that Ψ_α and Φ_α are indeed meaning-preserving.

Theorem 10.5 (Ψ_α and Φ_α are Meaning-Preserving). *Let a valuation $val : \mathcal{P} \to \{\mathtt{ff}, \mathtt{tt}\}$ be given, let $\mathfrak{G}$ be a formal alpha graph and f be a formula of propositional logic. Then we have:*

$$val \models f \iff val \models \Psi_\alpha(f) \qquad \textit{and} \qquad val \models \mathfrak{G} \iff val \models \Phi_\alpha(\mathfrak{G}) \quad .$$

Proof: Both the set of all propositional formulas and the definition of Ψ_α are defined inductively. Thus, the proof for the first equivalence is carried out induction over the composition of formulas. If P_i is a propositional variable, it is easy to see that we have $val \models P_i \iff val \models \Psi_\alpha(P_i)$. If $g = g_1 \wedge g_2$ is a formula, we obviously have

$$\begin{aligned} val \models g &\overset{\text{Def.10.2}}{\iff} val \models g_1 \text{ and } val \models g_2 \\ &\overset{\text{Ind. Hyp.}}{\iff} val \models \Psi_\alpha(g_1) \text{ and } val \models \Psi_\alpha(g_2) \\ &\overset{\text{Def.8.2}}{\iff} val \models \Psi_\alpha(g_1) \quad \Psi_\alpha(g_2) \end{aligned}$$

The proof for the case $g = \neg g'$ can be done analogously, which proves the first equivalence.

Now let $\mathfrak{G} := (V, \top, Cut, area, \kappa)$ be a fixed, formal alpha graph. Similar to the proof for Ψ_α, we can prove by induction over the tree $Cut \cup \{\top\}$ that we have

$$val \models \mathfrak{G}[c] \quad \iff val \models \Phi_\alpha(\mathfrak{G}, c) \tag{10.1}$$

for each context $c \in Cut \cup \{\top\}$ (see Def. 8.2 for the definition of $val \models \mathfrak{G}[c]$). Thus we have

$$val \models \mathfrak{G} \overset{\text{Def.8.2}}{\iff} val \models \mathfrak{G}[\top] \overset{\text{Eqn. 10.1}}{\iff} val \models \Phi_\alpha(\mathfrak{G}, \top) \overset{\text{Def.10.4}}{\iff} val \models \Phi_\alpha(\mathfrak{G}) \, ,$$

which proves the second equivalence. □

The next corollary fixes some immediate consequences of this theorem.

Corollary 10.6. *Let f be a formula of and let F be a set of formulas. Then we have:*

$$F \models f \quad \iff \quad \Psi_\alpha(F) \models \Psi_\alpha(f) \quad . \tag{10.2}$$

Analogously, let $\mathfrak{G}$ be a formal alpha graph and let $\mathfrak{H}$ be a set of formal alpha graphs. Then we have:

$$\mathfrak{H} \models \mathfrak{G} \quad \iff \quad \Phi_\alpha(\mathfrak{H}) \models \Phi_\alpha(\mathfrak{G}) \quad . \tag{10.3}$$

Moreover we have that $\mathfrak{G}$ and $\Psi_\alpha(\Phi_\alpha(\mathfrak{G}))$, as well as f and $\Phi_\alpha(\Psi_\alpha(f))$, are semantically equivalent.

Proof: The semantical equivalence of $\mathfrak{G}$ and $\Psi_\alpha(\Phi_\alpha(\mathfrak{G}))$, resp. f and $\Phi_\alpha(\Psi_\alpha(f))$, follows immediately from Thm. 10.5. It remains to prove Eqns. 10.2 and 10.3.

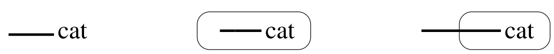
We show only the direction '$\Longrightarrow$" of the Eqn. 10.2, the other direction and the Eqn. 10.3 are shown analogously. So let f be a formula of and let F be a set of formulas with $F \models f$. Let val be a valuation with $val \models \Psi_\alpha[F]$, i.e., $val \models \Psi_\alpha(f')$ for all $f' \in F$. Theorem. 10.5 yields $val \models f'$ for all $f' \in F$. From $F \models f$ we obtain $val \models f$. Now Thm. 10.5 yields $val \models \Psi_\alpha(F)$. $\square$

Beta

11

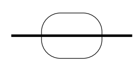
Getting Closer to Syntax and Semantics of Beta

We have already seen in the introduction some examples for Beta-graphs. Let us repeat the first examples of Chpt. 2:

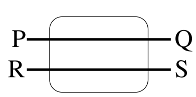


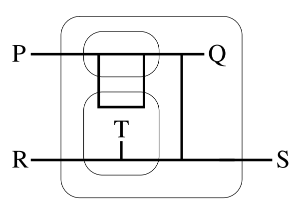
In all cases we have a heavy line (It will become clear soon why I do not write 'line of identity') which can be understood to denote one object. The meaning of the graphs are 'there is a cat', 'it is not true that there is a cat', and 'there is something which is not a cat', respectively.

But a heavy line does not necessarily stands for *one* object. We have already seen that the graph

$\mathfrak{E}_{twothings} :=$ 

has the meaning 'there are at least to things'. This is due to the fact that a heavy line traversing a cut denotes *non-identity*. But so far, this seemed to be a mere convention. Moreover, a comprehensive method for interpreting more complex diagrams is still missing. For example, what about the following diagrams with more complex structures of heavy lines crossing cuts? Can we be sure to grasp the precise meaning of them?





In this chapter, several examples of EGs will be discussed in detail. These examples should hopefully cover all features and aspects of EGs. The purpose of the discussion it twofold:

1. It will be elaborated how EGs are read. This is a reconstruction of Peirce's understanding of EGs.
2. From the reconstruction of Peirce's understanding we will obtain the basis for the forthcoming formalization of EGs in the next chapters.

These two purposes are more connected with each other than one would expect. Of course, we need a precise understanding of the readings of EGs to elaborate an appropriate formalization. It will turn out that the main clue to a deeper understanding of EGs is the idea that LoIs are composed of so-called identity spots. This insight will not only yield a method which allows to understand arbitrarily complex diagrams. Moreover, from this idea we will obtain the main idea for an appropriate formalization of diagrams.

11.1 Lines of Identities and Ligatures

We start with an investigation of the element which is added to existential graphs in the step from Alpha to Beta: the line of identity. Peirce describes a LoI as follows: '*The line of identity is [...] a heavy line with two ends and without other topical singularity (such as a point of branching or a node), not in contact with any other sign except at its extremities.*' (4.116), and in 4.444 he writes: '*Convention No. 6. A heavy line, called a line of identity, shall be a graph asserting the numerical identity of the individuals denoted by its two extremities.*').

It should be noted that Peirce does not claim that a LoI denotes *one* object. Instead of this, each of the two ends of the LoI denotes an object, which are identical. Of course, this is semantically equivalent to the existence of one object. The reason why Peirce does not use this interpretation of a LoI is, although Peirce called LoI 'indivisible graphs', that LoIs bear a kind of inner structure, which shall be unfolded now. Roughly speaking: Lines of identity are assembled of overlapping, heavily marked points. These points are described by Peirce in 4.405 by the following convention:

> Convention No. V. Every heavily marked point, whether isolated, the extremity of a heavy line, or at a furcation of a heavy line, shall denote a single individual, without in itself indicating what individual it is.

We find a similar quotation later in 4.474, where he writes

> Now every heavily marked point, whether isolated or forming a part of a heavy line, denotes an indesignate individual. [...] A heavy line is to be understood as asserting, when unenclosed, that all its points denote the same individual.

Thus, the most basic graph is not a single LoI. It is a simple, heavy marked point, a heavy dot like this:

•

From now on, these dots will be called 'identity spots'. Thus an identity spot '*shall denote a single individual, without in itself indicating what individual it is*', that is, it stands for the proposition 'there exists something'.

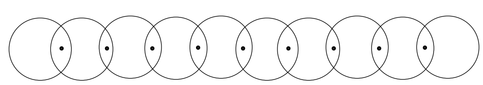
These spots may overlap, and this means that they denote the same object. As Peirce writes in 4.443: '*Convention No. 5. Two coincident points, not more, shall denote the same individual.*' Moreover, a LoI is composed of identity spots which overlap. Peirce writes in 4.561:

> A heavy line shall be considered as a continuum of contiguous dots; and since contiguous dots denote a single individual, such a line without any point of branching will signify the identity of the individuals denoted by its extremities.

Let us consider the following existential graph which is a simple LoI:

$$\mathfrak{E}_1 := \;\text{—}$$

The best way to depict Peirces understanding of LoIs is, roughly speaking, to magnify them, such that identity spots the LoI is composed of become visible. In a letter to Lady Welby, p.4, Peirce remarks that '*every line of identity ought to be considered as bristling with microscopic points of teridentity, so that* —— *when magnified shall be seen to be* ••••••••••••' (this quotation is adopted from Roberts ([Rob73], p. 117, footnote 5). We conclude that Peirce understands a heavy line, i.e., a LoI, to be composed of identity spots.[1] In his book 'A Peircean Reduction Thesis' ([Bur91a]), Burch provides magnifications of existential graphs.[2] As these magnifications are invented by Burch, they cannot be found in the works of Peirce, but they depict very clearly Peirce's understanding of LoI and are therefore very helpful to understand Peirce intuition. One possible magnification of $\mathfrak{E}_1$ is

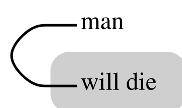


[1] The *teridentity* mentioned by Peirce is the triadic relation expressing the identity between *tree* objects. Its vital role in existential graphs is thoroughly discussed by Burch in 'A Peircean Reduction Thesis' ([Bur91a]). We will come back later in some places to this relation to unfold some of its specific properties, and particularly Chpt. 26 is dedicated to an extension of Burch's proof of Peirce's reduction thesis.

[2] Furthermore, he writes 'lines of identity are simply lines that are themselves composed of spots of identity (of various adicities) that are directly joined together', thus he shares this understanding of LoIs as well.

The identity spots are drawn as circles. The overlapping of these circles represents that the identity spots are coincident. Each point 'denotes an indesignate individual', and 'two coincident points [...] shall denote the same individual', that is, the individuals are identified. This identity relation is represented by the small dot in the intersection of two circles.

In the magnification, we have chosen a number of *ten* identity spots, but of course, this number is arbitrary. For a number of 10, the most explicit meaning of $\mathfrak{E}_1$ is: There are individuals $o_1, o_2, o_3, \ldots, o_{10}$, o_1 is (identical with) o_2, o_2 is (identical with) o_3, ..., and o_9 is (identical with) o_{10}. Of course, the meaning of a LoI does not depend on the number of identity spots it is composed of, as all identity spots finally denote the same object. As Peirce writes in a different place ([PS00]): '*The line of identity can be regarded as a graph composed of any number of dyads '-is-' or as a single dyad*' and he describes the graph



as follows: '*There is a man that is something that is something that is not anything that is anything unless it be something that will not die.*' The precise understanding of this passage will become clear in Sect. 11.3, where we discuss heavy lines crossing cuts. Moreover, describing a graph this way appears to be '*unspeakably trifling, – not to say idiotic*', as Peirce admits. Nonetheless, for our discussion, it is worth to note that these passages makes clear that the magnifications we use are indeed very close to Peirce's understanding of LoIs.

As a LoI and a single, heavy spot are different items, we can know understand why Peirce does not regard a LoI simply to denote one object. We conclude that Convention No. 6 of 4.444, where Peirce writes that a LoI is '*asserting the numerical identity of the individuals denoted by its two extremities*' is not a *convention or definition*, but a *conclusion* from Peirce's deeper understanding of LoIs.

The next graph we have to investigate is a device of branching heavy lines, i.e., we consider

$\mathfrak{E}_2 :=$ 

In order to understand a branching of heavy lines, Peirce provides in 4.445 and in [PS00] similar examples, which are depicted in Fig. 11.1. In 4.445, he explains the left graph of Fig. 11.1 as follows: '*The next convention to be laid down is so perfectly natural that the reader may well have a difficulty in perceiving that a separate convention is required for it. Namely, we may make a line of identity branch to express the identity of three individuals. Thus, Fig. 79 will express that some black bird is thievish.*' Similar, in his tutorial [PS00] of 1909 he writes that the right graph of Fig. 11.1 '*is a graph instance composed of instances of three indivisible graphs which assert 'there is a male',*

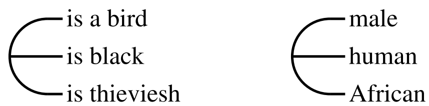
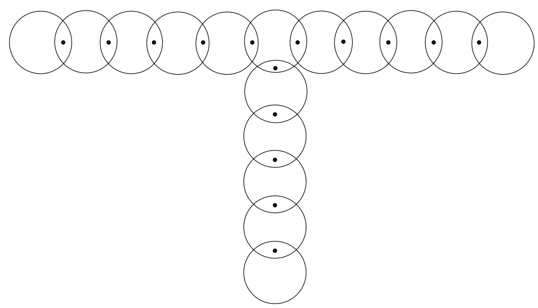


Fig. 11.1. Fig. 79 of 4.445 and an example the of the tutorial [PS00]

'there is something human' and 'there is an African'. The syntactic junction or point of teridentity asserts the identity of something denoted by all three.'

In contrast to the common notation of identity as dyadic which expresses the identity of two objects,[3] the teridentity expresses the identity of three objects. Consider the following magnification of $\mathfrak{E}_2$:



The point where all lines meet is the point of teridentity (4.406 '*Also, a point upon which three lines of identity abut is a graph expressing the relation of teridentity*'). In the magnification, the three identities are depicted by the three small dots in the circle in the middle.

The so-called teridentity, that is, the identity relation for *three* objects, plays a crucial role in Peirce's diagrammatic logic. In a linear representation of first order predicate logic like $\mathcal{FO}$, where devices like variables or names are used to denote objects, it is sufficient to have a dyadic identity relation. For example, to express that three variables x, y, z denote the same object, we simply use the formula $x = y \wedge x = z$ or $x = y \wedge y = z \wedge x = z$. But, it seems quite obvious that in the system of existential graphs, we need a device like branching heavy lines to express the identity of more than two objects. In 4.445, Peirce writes '*Now it is plain that no number of mere bi-terminal bonds [...], can ever assert the identity of three things, although when we once have a three-way branch, any higher number of terminals can be produced from it, as in Fig. 80.*'[4]

This passage contains two kinds of informations. First of all, Peirce states that branching points are needed to express the identity of more than two

---

[3] More precisely: The identity of the objects which are denoted by two signs. To quote Peirce (4.464): '*But identity, though expressed by the line as a dyadic rela-*

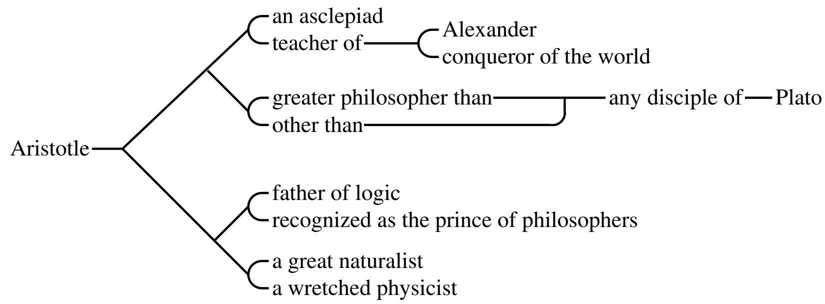


Fig. 11.2. Fig. 80 of 4.445

objects. But Peirce did not take branching points with an arbitrary number of branches into account: It is likely that he considered only EGs where no branching points with more than three branches occur. For example, in Convention No. 7, which will immediately be provided, Peirce says that a branching line of identity expresses the identity of *three* individuals), or in the quotation in the letter to Lady Welby given on page 99 he explicitely states that a line of identity is composed of *ter*identity spots. In fact, none of the examples Peirce provides in Book II, 'Existential Graphs' of [HB35] have branching points with more than three branches. existential graphs only

A branching point with three branches expresses the identity of three objects. The other information in the above-quoted passage is an argument that identity between more than three objects can be expressed by means of teridentity. To put it differently: Allowing only branching points three branches (which is a syntactical restriction) does not lead to a loss of expressiveness. In the formal elaboration of EGs, this will be mathematically proven (see Lem. 16.3 and the following examples).

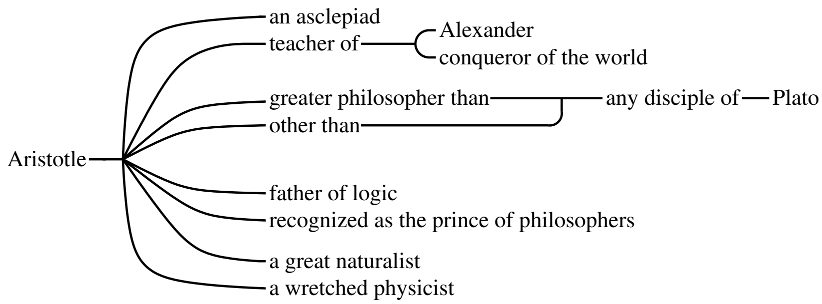
For this reason, Peirce hardly considers EGs where branching points with more than three branches occur. The quotations above indicate, as they explicitely refer to the identity of *three* objects, that Peirce did even not allow branching points with more than three branches.

Considering EGs where only branching points with three branches are allowed is in my view a restriction which leads to an unnecessary syntactical overhead. Moreover, it will turn out that the semantics and transformation rules can be canonically extended to existential graphs where branching points with more

tion, is not a relation between two things, but between two representamens of the same thing.'

[4] In this diagram, two slight thoughtlessnesses are remarkable: First of all, it should recognized that Peirce uses singular terms, i.e. names for objects, in this diagram (namely 'Aristotle', 'Alexander' and 'Plato'). Secondly, he implicitly brings in an *universal* quantification by the use of a relation '*any* disciple of'.

than three branches are permitted. Thus we will consider existential graphs having branching points with more than thee branches, too. For example, the graph of Fig. 11.1 could be transformed into the following graph, having such a branching point.



The need of incorporating the teridentity into existential graphs, and the fact that identity relations of higher arities than 3 can be expressed by means of teridentity, is a part of Peirce's famous reduction thesis.. Roughly speaking, this thesis claims that each each relation with an arity greater than three can be reduced in some sense to ternary relations, but it is impossible to reduce relation to binary relations. In fact, the need of the branching points is *not* based on the graphical representations of EGs, i.e., on the *syntax* of EGs. It can be be proven *semantically* that we need the teridentity relation (which is diagrammatically represented by a branching point with three branches). This will be elaborated in Chpt. 26.

To conclude the short discussion of the number of branches, I provide one one the rare examples of an graph where such a branching point is allowed and discussed. In the paper 'the logic of relatives' (The monist, vol. 7, 1897, see 3.456–3.552), he writes in 3.471

> Several propositions are in this last paragraph stated about logical medads which now must be shown to be true. In the first place, although it be granted that every relative has a definite number of blanks, or loose ends, yet it would seem, at first sight, that there is no need of each of these joining no more than one other. For instance, taking the triad "– kills – to gratify –," why may not the three loose

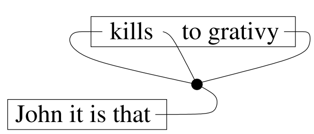


Fig. 11.3. Fig. 3 of 4.471

ends all join in one node and then be connected with the loose end of the monad "John is –" as in Figure 3 making the proposition "John it is that kills what is John to gratify what is John"? The answer is, that a little exercise of generalizing power will show that such a four-way node is really a tetradic relative, which may be expressed in words thus, "– is identical with – and with – and with –"; so that the medad is really equivalent to that of Figure 4 [...]

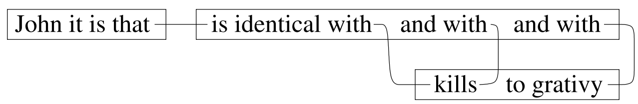
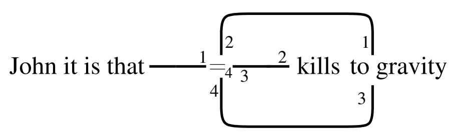


Fig. 11.4. Fig. 4 of 4.471

The graphical notation Peirce uses in this place differs from the common notation for EGs. In this passage, Peirce explicitely discusses the identity with four places. If we had a symbol $\doteq_4$ for this identity, the last graph could be depicted as follows:



We will come back to this in Chpts. 21 and 16. Moreover, it is remarkable that Peirce uses here a dedicated bold spot to refer to an individual, and this spot is linked with lines to the relatives (relations). This notation is very close to the formal elaboration of EGs, which will be soon be provided.

Peirce sometimes uses the term 'line of identity' for a linked structure of heavy lines. In his Cambridge lectures of 1898 ([Pei92]) we find the phrase: '*Now as long as there is but one such line of identity, whether it branches or not [...].*', and even in the collected papers, 4.446, we find '*Convention No. 7. A branching line of identity shall express a triad rhema signifying the identity of the three individuals, whose designations are represented as filling the blanks of the rhema by coincidence with the three terminals of the line.*' But Peirce's quotation should be understood to be a simplification for the sake of convenience. In both quotations, he speaks about linked structures of heavy lines which are wholly placed on the sheet of assertion. In this case, such a linked structure can -similar to a LoI- be still understood to denote a single object, and in this respect, using the term 'line of identity' is not misleading. But linked structures of heavy lines may cross cuts, and it will turn out that this situation deserves a special treatment, and there are cases where such a linked structure cannot any more be understood to denote a single object.

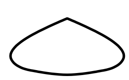
For this reason, Peirce introduces a new term for linked structures of LoIs. In the collected papers, he writes in 4.407: '*A collection composed of any line of identity together with all others that are connected with it directly or through still others is termed a ligature. Thus, ligatures often cross cuts, and, in that case, are not graphs*', and later on in 4.416, he writes '*The totality of all the lines of identity that join one another is termed a ligature. A ligature is not generally a graph, since it may be part in one area and part in another. It is said to lie within any cut which it is wholly within.*' So he explicit discriminates between *one* line of identity and a linked structure of lines of identity which he calls ligature. In this treatise, the distinction between lines of identity and ligatures is adopted.

The quoted passages indicate even more: Peirce speaks of collections of LoIs '*together with* all *others*', and he considers '*the* totality *of all the lines of identity that join one another*', thus Peirce's understanding of a ligature is a *maximal* connected network of LoIs. In this treatise, this condition will *not* be used, when ligatures are formally defined.

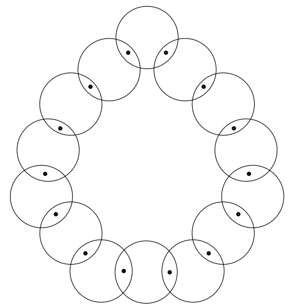
A single LoI is understood to be a ligature as well, but a ligature, as soon as it has branches or when it crosses a cut, is not a LoI. As Peirce writes in 4.499: '*Let us, then, call a series of lines of identity abutting upon one another at seps, a ligature; and we may extend the meaning of the word so that even a single line of identity shall be called a ligature. A ligature composed of more than one line of identity may be distinguished as a compound ligature.*' Thus, $\mathfrak{E}_2$ is made up of three LoIs which form a (compound) ligature.

Comment: In secondary literature, linked structures of heavy lines are sometimes called 'line of identity' as well. For example, Roberts writes in [Rob73]: '*We could consider the [...] lines as a single line of identity with three extremities which have a point in common [...]. And the totality of all the lines of identity that join one another he* (Peirce) *called a 'ligature'. we prefer the former terminology [...]*', and he provides the following convention: '*C8: A branching line of identity with n number of branches will be used to express the identity of the n individuals denoted by its n extremities.*' Sowa shares the understanding that the linked structure can be regarded as a single LoI. For example, in [Sow97a] he says: '*In Peirce's graphs, a bar or a linked structure of bars is called a line of identity*', and in his commentary in [PS00] he describes a graph similar to the right graph of Fig.11.3 as follows: '*[...] part of the line of identity is outside the negation. When a line of identity crosses one ore more negations [...]*'.

Finally, it should be noted that we have closed heavy lines as well. Consider the following graph and its magnification:

$\mathfrak{E}_3 :=$ 

This graph can be magnified as follows:



One might have the impression that the discussion so far is much too tedious. But it will help us to understand how existential graphs are read, no matter how complicated they are. Particularly, they will help us to answer the questions we raised at the beginning of this chapter. Moreover, it leads us to an approach how existential graphs can be mathematically be formalized. The first step will be presented now.

It is a natural approach to use the notations of (mathematical) graph theory for a formalization of Peirce's graphs. The main idea is to encode the identity spots of an existential graph by vertices in a mathematical graph. When two identity spots are coincident (i.e., they denote the same object), we draw an edge between the corresponding edges. For example, below we have depicted two mathematical graphs which could be seen to be formalizations of $\mathfrak{E}_1$ resp. $\mathfrak{E}_2$:

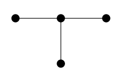
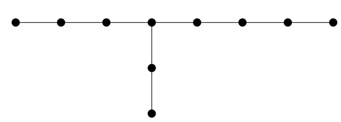
 for $\mathfrak{E}_1$, and  for $\mathfrak{E}_2$.

Fig. 11.5. Two possible formalizations for $\mathfrak{E}_1$ and $\mathfrak{E}_2$

The number of identity spots which form a LoI is of course not fixed. In contrast, in the magnifications, we have *chosen* an arbitrary, finite number of spots to represent a LoI (Peirce said that a LoI '*can be regarded as a graph composed of any number of dyads '-is-' or as a single dyad.*'). Thus, the following mathematical graphs can be understood to be formalizations for $\mathfrak{E}_1$ and $\mathfrak{E}_2$ as well:

 for $\mathfrak{E}_1$, and  for $\mathfrak{E}_2$

.

Fig. 11.6. Two different possible formalizations for $\mathfrak{E}_1$ and $\mathfrak{E}_2$

$\mathfrak{E}_3$ could be seen as two LoI which are joined at both extremities. Then we would formalize $\mathfrak{E}_3$ as graph with two vertices and two edges. On the other hand, $\mathfrak{E}_3$ could be seen as one LoI of which both extremities are joined: This would yield a mathematical graph with one vertex and one 'self-returning' edge. We will allow this graph as well, i.e., the following two graphs will be possible formalizations of $\mathfrak{E}_3$.[5]

In the ongoing formalization, an isolated heavy point • is distinguished from a line of identity. That is, $\mathfrak{E}_0$ is formalized only by the mathematical graph which is made up of a single vertex. Vice versa, a single vertex is not an appropriate formalization of $\mathfrak{E}_3$.

11.2 Predicates

In Fig. 11.1, we have already used an EG with predicates in this chapter. In order to start our investigation on predicates, we consider the following two subgraphs of the graph in Fig. 11.1:

Fig. 11.7. Two subgraphs of Fig. 11.1

At a first glance, the meanings of both graphs are clear: The left graph is read 'there is a father of logic', and the right graph is read 'Aristotle is the teacher of Alexander'. This understanding is not wrong, but for the right graph, some further discussion is needed.

First of all, in the left graph, we have a LoI attached to the string 'father of logic'. This string does not denote an object: It is the name of a unary predicate. Being 'father of logic' is an attribute: Some objects (of our universe of discourse) may have this attribute, while others have not. For our reading of the right graph, we used our (background) knowledge that the names 'Aristotle' and 'Alexander' denote unique objects instead of unary predicates. This makes of course a crucial difference.

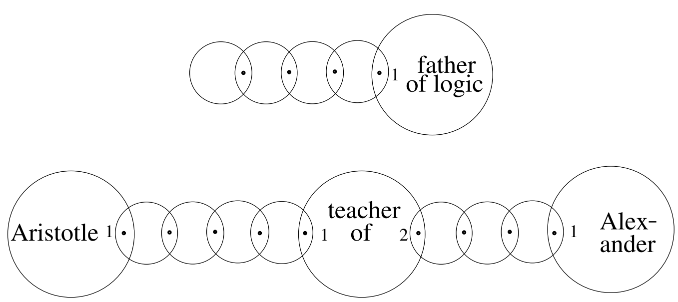
Peirce wanted to develop a 'logic of relatives' (i.e., relations). In fact, in his calculus for EGs, Peirce did not provide any rules for the treatment of object

[5] One might think that it is better to consider only those mathematical graphs with a 'minimal number' of vertices (i.e., the graphs of Fig. 11.6 should be dismissed). But the forthcoming formalization of the transformation rules is much easier if we allow graphs with a higher number of vertices as well.

names, i.e. he treated all names in EGs as names for predicates.[6] Thus, more formally, the meaning of the right graph is 'there are two objects such that the first object is Aristotle (is of type 'Aristotle'), the second object is Alexander (is of type 'Alexander'), and the first object is a teacher of the second object.'

Peirce understood a predicate as a '*blank form of [a] proposition*' (4.438 and 4.560). The relation 'teacher of' can be written as such a blank form as follows: __teacher of__. The two blanks, i.e., the places of the predicate, have to be filled with two (not necessarily different) arguments to obtain a proposition. Similar to the identity spots LoIs are composed of, in EGs, predicates occur as so-called *predicate spots*. Peirce imagined that to each blank of an n-ary predicate corresponds a 'certain place' on the periphery of the predicate spot, called a *hook* of the spot. We can attach extremities of LoIs to these hooks (which corresponds to the filling of the blanks of the proposition with arguments). EGs are formalization of propositions: Particularly, in EGs, all blanks of predicates are filled, that is, to each hook of each predicate an LoI is attached. For this reason, there is no graphical representation for hooks, or, as Zeman writes in [Zem64]: '*Strangely enough, however, we shall not in practice see these hooks; in any graph properly so called, all hooks are already filled, connected to the appropriate signs for individuals.*' At a first glance, an empty hook of a spot can be compared with a free variable in a formula of first order predicate logic, but an empty hook should better be understood to correspond to a *missing* variable in an n-ary predicate, thus leading to a non-well-formed-formula.

For the magnification of EGs, I adopt the approach of Burch and draw the predicate spots larger than the identity spots. Moreover, for the magnification it makes sense to represent the hooks graphically. The n hooks of an n-ary predicate are are indicated by n small dots (similar to the dots in the intersection of identity spots) which are (in contrast to Burch) labeled with the numbers $1, \ldots, n$. Thus, the graphs of Fig. 11.2 can be magnified as follows:



Of course, the order of the hooks is crucial. For example, considering the right graph of Fig. 11.2, it makes a difference whether Aristotle was the teacher of

[6] As the right graph of Fig. 11.2 show, it is desirable to have object names. In the part 'Extending the System', object and function names will be added to EGs.

Alexander, or Alexander was the teacher of Aristotle. We read the graph from left to right: Therefore we grasp its intended meaning. Of course, Peirce was aware that the order of the arguments of a relation is important, but there are only very few passages where Peirce explicitely discusses how this order is depicted graphically in EGs. In 4.388, we find:

> In taking account of relations, it is necessary to distinguish between the different sides of the letters. Thus let l be taken in such a sense that X—l—Y means "X loves Y." Then X l Y will mean "Y loves X."

Moreover, for the gamma system, in 4.470, he writes that some LoI are attached to the hooks of a spot are '*taken in their order clockwise.*' Our goal is still to provide a formalization of EGs which prescinds from the graphical properties of the diagrams. So far, for the formalization of LoIs, we used the notation of mathematical graph theory: Identity spots are formalized by vertices of a mathematical graph, and LoIs by edges between these vertices. An edge encodes the identity between the objects denoted by the vertices. Identity spots have a two-fold purpose: First of all, they denote objects. Moreover, by letting identity spots overlap, the identity relation is represented by identity spots as well. In our ongoing formalization, these two different functions are formalized by two different syntactical entities: Vertices will denote objects, and the identity relation is formalized by edges. Identity is a special dyadic relation, so it is natural to extend the formalization to relations as follows: We consider graphs with so-called so-called *directed hyper-edges*. An occurrence of an n-ary relation name will be formalized as an n-ary directed hyper-edge, labeled with the relation name. The identity relation will be captured by 2-ary edges, labeled with the special relation name =. The precise definition will be given in the next chapter; in this chapter, this idea shall be illustrated with some examples.[7]

The left graph of Fig. 11.2 is encoded with one vertex and one edge which is attached to this vertex. Furthermore, this edge is labeled with the predicate name 'father of logic'. This yields the following mathematical graph:

$\bullet$——^{1} father of logic

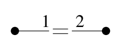
The edge is represented by writing the name of its label and drawing lines from this name to the dot which represent the incident vertex.

[7] The magnifications offer another possibility to formalize the predicates of EGs. It is possible to encode the predicates as *vertices* as well. Formalizations like this have been carried out by several authors, for example by Chein and Mugnier for CGs (see [CM92, CM95]), or by Pollandt in [Pol02] and by Hereth Correia and Pöschel in [HCP04, HCP06] for relation graphs. I consider an encoding with edges instead of vertices more practical, but somehow, this decision depends on matter of (mathematical) taste.

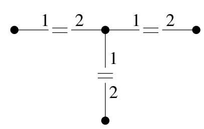
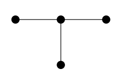
The right graph of Fig. 11.2 is encoded as follows:

Here we have two vertices and three edges. The first edge is an edge which is coincident with one vertex (the left one) and labeled with the name ‘Aristotle’. The second edge is coincident with both vertices. This edge is represented by writing the name of its label, i.e., ‘teacher of’ and drawing lines from this name to the dots which represent the incident vertices. The order of the incident vertices is represented by writing the numbers 1 resp. 2 above the lines which go from the name to the first resp. second vertex (i.e., the representing dots). Analogously, we have a third edge which is labeled with ‘Alexander’ and which is incident with the second vertex. The next figure is another representation of exactly the same mathematical graph, where the three edges are informally indicated.

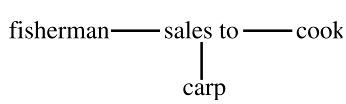
As said above: Identity is a special relation, and the former ‘identity edges’ are formalized as as directed edges which are labeled with a name for identity, like ‘=’. That is, we have the following formalizations of $\mathfrak{E}_1$ and $\mathfrak{E}_2$:

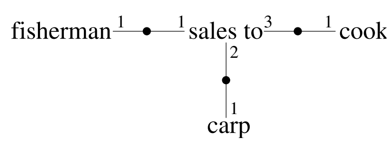
 instead of

and

 instead of  .

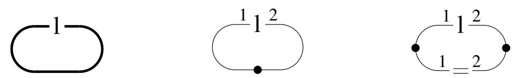
As we used directed hyper-edges, we have no difficulties to encode relations of arity three or higher as well. For example, if we have a ternary relation ‘sales to’, then we can depict an EG with the meaning ‘a fisherman sales a carp to a cook’ and its formalization as follows:





A possible magnification of this graph is given in Fig. 11.2.

Finally, a vertex may be linked multiple to a predicate. In the next figure, you find an EG with the meaning ‘somebody loves himself’ (with l standing for ‘loves’, like in the above quotation of Peirce) and two possible formalizations.



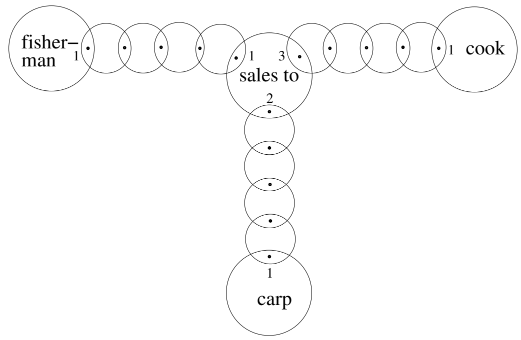


Fig. 11.8. A magnification of a graph with a ternary relation '_sales_to_'.

11.3 Cuts

In this section, we extend our investigation to the cuts of existential graphs. In existential graphs without cuts, every LoI, even every ligature can be understood to represent a *single* object. The investigation of LoIs and ligatures has to be extended when we take existential graphs with cuts into account. Even Peirce writes in 4.449: '*There is no difficulty in interpreting the line of identity until it crosses a sep. To interpret it in that case, two new conventions will be required.*'

We start with the graph $\mathfrak{E}_{twothings}$ and two further examples of Peirce in which a LoI seems to cross resp. pass a cut.[8]

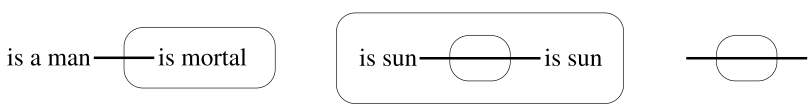


Fig. 11.9. Fig. 67 and 68 of 4.407, and $\mathfrak{E}_{twothings}$

In all graphs, one might think we have only one LoI, which, then, should denote one object. In fact, the meaning of the left graph is 'there is a man who is not mortal', i.e., the heavy line stands for one object.

But we have already seen that the meaning of $\mathfrak{E}_{twothings}$ is 'there are at least two things'; that is, the heavy line of this graph does *not* stand for one object. Analogously, the meaning of the graph n the middle is 'it is not true that there are two suns which are different', or 'there is at most one sun' for short.

Peirce explains in 4.407 the first and second graph (for the first graph, an similar graph an explanation can be found in [PS00] as well) as follows:

[8] I changed the shape of the LoIs and cuts.

> A heavily marked point may be on a cut; and such a point shall be interpreted as lying in the place of the cut and at the same time as denoting an individual identical with the individual denoted by the extremity of a line of identity on the area of the cut and abutting upon the marked point on the cut. Thus, in Fig. 67, if we refer to the individual denoted by the point where the two lines meet on the cut, as X, the assertion is, "Some individual, X, of the universe is a man, and nothing is at once mortal and identical with X"; i.e., some man is not mortal. So in Fig. 68, if X and Y are the individuals denoted by the points on the [inner] cut, the interpretation is, "If X is the sun and Y is the sun, X and Y are identical."

There are two things remarkable in this quotation: First of all, Peirce speaks about 'points on cuts'. These points deserve a deeper investigation. Secondly, he says we have in the left graph *two* lines of identity which meet on the cut. It has to be clarified how such an overlapping of two lines of identity on a cut has to be interpreted. These questions are addressed by the two convention Peirce spoke about in 4.449. These conventions are as follows:

> 4.450: Convention No. 8. Points on a sep shall be considered to lie outside the close of the sep so that the junction of such a point with any other point outside the sep by a line of identity shall be interpreted as it would be if the point on the sep were outside and away from the sep.
>
> 4.451: Convention No. 9. The junction by a line of identity of a point on a sep to a point within the close of the sep shall assert of such individual as is denoted by the point on the sep, according to the position of that point by Convention No. 8, a hypothetical conditional identity, according to the conventions applicable to graphs situated as is the portion of that line that is in the close of the sep.

These conventions shall be discussed in detail. We start our investigation with points on a cut, particularly, why Peirce says that ' *Points on a sep shall be considered to lie outside the close of the sep*'. In 4.502, Peirce provides the background of this convention. Consider the graphs of Fig. 11.3.

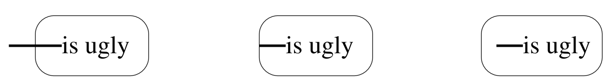
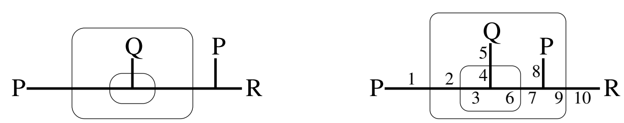


Fig. 11.10. Fig. 154, 155 and 156 of 4.502

In 4.502, he writes: '*[...] consider Fig. 154. Now the rule of erasure of an unenclosed graph certainly allows the transformation of this into Fig. 155,* *which must therefore be interpreted to mean "Something is not ugly," and must not be confounded with Fig. 156, "Nothing is ugly."* ' In fact, if we interpreted a point on a cut to lie inside the area of the cut, Figs. 155 and 156 had the same meaning, and the rule of erasure would allow to conclude 'nothing is ugly' from 'something is not ugly', which obviously is not a correct implication. This explains why points on a cut must be interpreted to lie outside the area of the cut.

Now, before we discuss conventions No. 8 and 9 further, we first have to investigate how heavily drawn lines which cross cuts are syntactically understood in terms of lines of identity and ligatures. The heavy line in Peirce's Fig. 154 can still be interpreted to denote one object, thus, is seems to be natural that the heavy line can be understood to be a line of identity. But it has already been mentioned that lines of identity do not cross cuts. Recall that a LoI is a heavy line '*without other topical singularity (such as a point of branching or a node), not in contact with any other sign,*', or in 4.406, Peirce writes that a LoI does not have '*any sort of interruption*'. For this reason, he can draw the following corollary in 4.406: ' *Corollaries. It follows that no line of identity can cross a cut.*' Recall that a line of identity is '*a heavy line with two ends and without other topical singularity (such as a point of branching or a node), not in contact with any other sign except at its extremities.*' Networks of heavy lines of heavy lines crossing cuts are called ligatures. Consider the following diagrams:



The left diagram depicts an EG. Obviously, this EG contains one network of heavy lines, i.e., it contains only one maximal ligature. This ligature is composed of at least 10 lines of identity. We have to write "at least", because each of the lines of identity can be understood to be a ligature which is composed of (smaller) lines of identity as well. In the right diagram, these 10 lines of identity are enumerated.

After we have have clarified the term *ligature*, we need to know how ligatures in EGs are interpreted. We already know how LoIs are interpreted (they assert the identity of the two objects denoted by its extremities), hence we know how to interpret branching points as well. So we are able to interpret ligatures *in a given context*. But this does not help if we have a heavy line which crosses a cut. Thus, we have to investigate heavy lines crossing cuts further.

How shall the heavy line of Peirce's Fig. 154 be understood? In 4.416, Peirce says: '*Two lines of identity, one outside a cut and the other on the area of the same cut, may have each an extremity at the same point on the cut.*' This explains how the heavy line of Peirce's Fig. 154 can syntactically be understood: It is composed of two LoIs which meet on a cut. Let us denote the left LoI, which is placed outside the cut, with l_1, and the right LoI, which is placed inside the cut, with l_2.

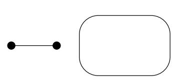
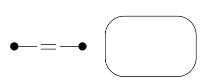
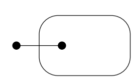
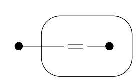
Now we have to investigate how this device of l_1 and l_2 is semantically be interpreted. Peirce's Conventions No. 8 and 9 make the interpretation of points on a cut which are endpoints of one more more LoIs explicit. This shall be discussed now.

A LoI expresses that the objects denoted by its two extremities are identical. The LoIs l_1 and l_2 have a point in common, namely the point on the cut. Let us denote the object denoted by the left endpoint of l_1 by o_1, the object denoted by the common endpoint of l_1 and l_2 on the cut by o_2, and the object denoted by the right endpoint of l_2 by by o_3.

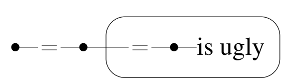
Now Conventions No. 8 and 9, applied to our example, yield the following: The existence of o_1 and o_2 is asserted, but the existence of o_3, as the right endpoint of l_2 is placed inside the cut, is negated. The LoI l_1 expresses that o_1 and o_2 are identical, the LoI l_2 expresses that o_2 and o_3 are identical. The first identity is expressed by a LoI outside the cut, the second by a LoI inside the cut, thus, the identity of o_1 and o_2 is asserted, but the identity of o_2 and o_3 has to be negated. A very explicit (but hardly understandable) translation of the graph of Peirce's Fig. 154 in English is therefore: 'We have two objects o_1 and o_2 which are identical, and it is not true that we have a third object o_3 which is identical with o_2 and which is ugly.' This proposition is equivalent to 'We have an object which is not ugly.'

Peirce's conventions are very helpful for our formalization of existential graphs. We have to add cuts to our formalization, but there is no need to formalize graphs where it is possible to express that a vertex is *on* a cut: It is sufficient to consider structures where the vertex is *inside* or *outside* the cut. That is, we will *not* consider graphs like the following two graphs:

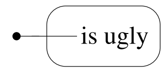
Due to Conventions 8 and 9, we will consider the following two graphs instead:

 i.e.,  , and  i.e.,  .

(But we have to keep in mind that for the second graph, the identity expressed by the edge between the two vertices takes place *inside* the cut.) For Peirce's Fig. 154, if we 'translate' each of the two LoIs by two vertices (the extremities) and an identity edge between them (expressing the identity of the extremities), the following graph is a possible formalization:

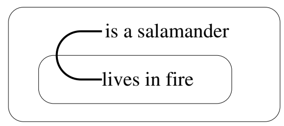


It will turn out in the next chapter that this graph can be transformed to



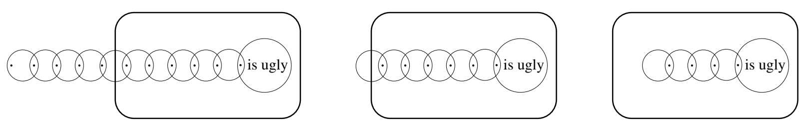
For this graph, it is easy to see that its meaning is ‘there is something which is not ugly’.

Comment: Peirce investigates heavy lines crossing a cut with another example. In 4.449, he asks: ‘*How shall we express the proposition ’Every salamander lives in fire,’ or ’If it be true that something is a salamander then it will always be true that* that something *lives in fire’?*’ He comes to the conclusion that the only reasonable is the graph depicted below.



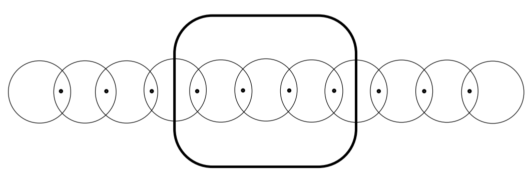
Particularly, he obtains: ‘*In order, therefore, to legitimate our interpretation of Fig. 83, we must agree that a line of identity crossing a sep simply asserts the identity of the individual denoted by its outer part and the individual denoted by its inner part.*’ Then, he comes to Conventions 8. and 9. quoted above.

In our magnifications, we sketched the join of two identity spots, which expresses the identity denoted by identity spot, by small black dots. From the discussion above, we conclude that the following three diagrams are reasonable magnifications of the graphs of Fig. 11.3:

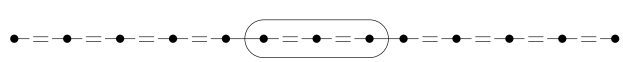


Again, we see that the magnifications correspond to the ongoing formalization of Peirce’s graphs.

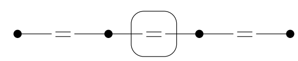
Consider now the graph $\mathfrak{E}_{twothings}$. The meaning of $\mathfrak{E}_{twothings}$ is ‘there are at least two things’. We now have the ability to analyze why this is the correct interpretation of $\mathfrak{E}_{twothings}$. A possible magnification of the graph is



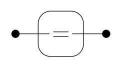
Possible formalizations of $\mathfrak{E}_{twothings}$ can be obtained from the possible magnifications of $\mathfrak{E}_{twothings}$ or from Peirce’s conventions. The formalization obtained from the given magnification is depicted below. Thus, the following is a possible formalization of the graph:



A formalization of $\mathfrak{E}_{twothings}$ can be better obtained from Peirce's Convention No. 9. $\mathfrak{E}_{twothings}$ contains a ligature which is composed of three LoIs. The LoI in the middle has with each of the other two LoIs an identity spot in common, and these two spots are placed on the cut, that is, they are considered to be outside the cut. Moreover, the argumentation above from which Peirce concluded Convention No. 9 explains that the LoI inside the cut corresponds to a '*hypothetical conditional identity, according to the conventions applicable to graphs situated as is the portion of that line that is in the close of the sep.*' Together with the two LoIs outside the cut end their endpoints, we get the following formalization:

 .

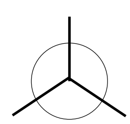
Again, we will see in the next chapter that these formalizations can be simplified. The following graph contains only two vertices, which are placed on the sheet of assertion, and one identity-edge, which is placed in the cut and which connects the vertices.



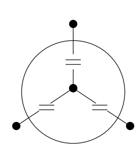
As the identity-edge is placed inside the cut, the identity of the objects denoted by the vertices is denied. I.e., this graph has in fact the meaning 'there are two objects o_1 and o_2 such that it is not true that o_1 and o_2 are identical', that is, there are at least two things. This is probably the best readable formalization of $\mathfrak{E}_{twothings}$.

We have seen that understanding of $\mathfrak{E}_{twothings}$ as 'there are at least two things' is not a convention or definition, but it can be obtained from a deeper discussion of EGs. Consequently, Peirce states in 4.468 the meaning of $\mathfrak{E}_{twothings}$ as a corollary: '*Interpretational Corollary 7. A line of identity traversing a sep will signify non-identity.*'

Analogously, we can now understand the next EG, which is Fig.118 in 4.469:

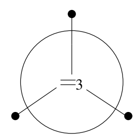


Due to our discussion, a possible formalization of this graph is



We see that the meaning of this graph is 'there are three things which are not all identical'. Note that this is a strictly weaker proposition than 'there are at least three things'. Again, Peirce states the meaning of this graph as a corollary. Directly after the last corollary, he writes in 4.469. '*Interpretational Corollary 8. A branching of a line of identity enclosed in a sep, as in Fig. 118, will express that three individuals are not all identical.*'

If we had a symbol $\doteq_3$ for teridentity (identity of three objects), the graph could even simpler be formalized as follows:

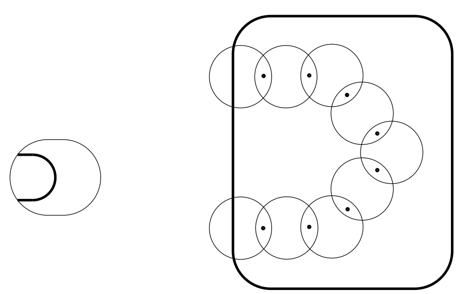


This is even better readable. In the formal definition of EGs, we will use only the usual, dyadic identity, but in Chapter 16, we will come back to this idea in order to obtain an improved reading 'algorithm' for EGs.

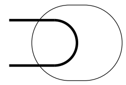
11.4 Border cases: LoI touching or crossing on a Cut

In the last section, in the discussion after Convention 8 and 9, I have already argued that we do not need to incorporate points on a cut in our forthcoming formalization of EGs. In this section we will discuss a few more examples of Peirce's graphs where LoIs only touch a cut, or when more than two LoIs meet on directly on a cut. Let us informally call graphs like these 'degenerated'. In some places, Peirce indeed uses degenerated graphs (in Chpt. 14, where the rules of Peirce will be discussed, on page 153 an example of Peirce with two degenerated graphs is provided). We will see that to each degenerated graph corresponds a canonically given non-degenerated graph, thus, for Peirce's graphs as well, it is sufficient to consider only non-generated graphs.

The main rule to transform a degenerated graph into an non-degenerated graph is: Points on a cut are considered outside the cut. This rule shall be elaborated in this section. We start with a simple example. Consider the following graph and its magnification:

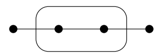


If we consider the points which terminate on the cut to lie outside the cut, we obtain the following diagram



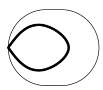
,

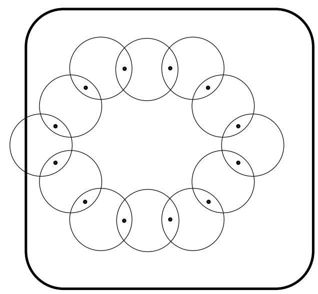
which is simply another way of drawing $\mathfrak{E}_{twothings}$. This is captured by the ongoing formalization as well: Due to the discussion in the last section,



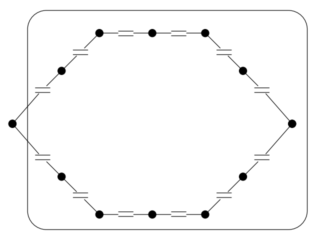
is a possible translation of this graph, which again yields that this graph is equivalent to $\mathfrak{E}_{twothings}$.

A similar example is the following graph and its magnification:

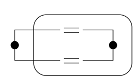




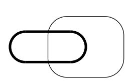
Contrary to the last example, we have *one* identity spot instead of two (which is an endpoint of a LoI). The magnification makes clear that



is an appropriate formalization of this graph, which again can be simplified, for example to

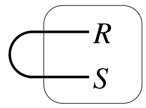


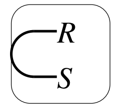
From the possible magnifications, thus the possible formalizations, we conclude that the following Peircean graph is an appropriate substitute for the starting graph:

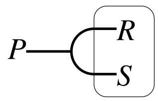


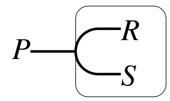
(This is the graph of Fig. 18, page 54 in the book of Roberts. Its meaning is 'there is a thing which is not identical with itself', i.e., this EG is contradictory.)

Analogous considerations show that it is sufficient

- to consider 

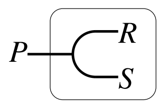
instead of 

or
- to consider 

instead of 

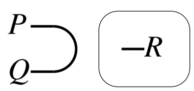
.

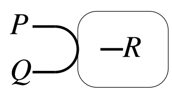
The meaning of the last two graphs is 'there is an object which has property P, but is has not both properties R and S'. It should be noted that these two graphs are semantically equivalent to

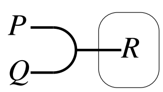


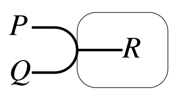
This will be elaborated further in Chpt. 14 and Chpt. 16.

When a LoI touches a cut from the outside, we already know from the discussion after Convention 8 and 9 in the last section that the touching can be omitted. For example,

- we consider 

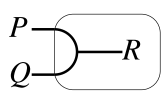
instead of 

or
- we consider 

instead of 

.

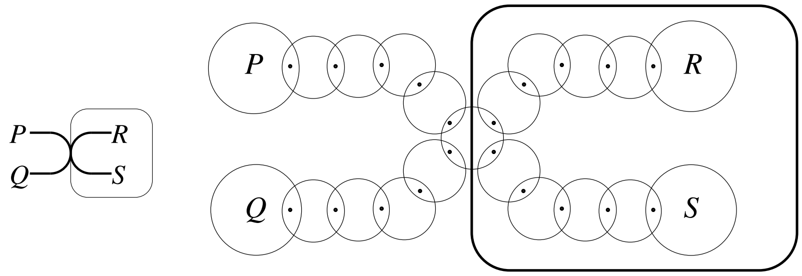
The meaning of the last two graphs is 'there is an object which has properties P and Q, but not R'. It should be noted that the last two graphs entail, but are *not* semantically equivalent to the next graph.



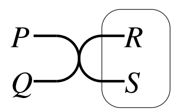
.

(The meaning of this graph is: 'There is an object o_1 with property P and an object o_2 with property Q, such that either o_1 and o_2 are not identical, or o_1 and o_2 are identical, but the property R does not hold for the (identical objects) o_1 and o_2').

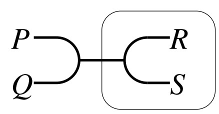
Finally we discuss a graph where two heavy lines cross directly on a cut. Consider the following graph and its magnification:



Again the magnification helps to see that the following graph is the appropriate, non-degenerated substitute:

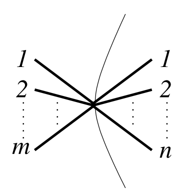


The following graph is semantically equivalent, too:

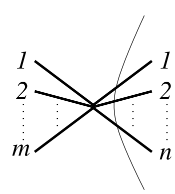


The meaning of these graphs is 'there is an object which has property P and Q, but is has not both properties R and S'.

The examples show how the statement 'points on a cut are considered outside the cut' can be understood to dismiss degenerated EGs. Assume we have a degenerated EG, where some LoIs meet on a cut. This shall be depicted as follows (from the cut, only a segment of the cut-line is depicted):



Then this device can be replaced by



to obtain a equivalent Peircean graph. Note that this 'transformation-rule' can even be applied for $m = 0$ or $n = 0$. With this rule, we can transform each degenerated EG into a non-degenerated EG. Roughly speaking, if we have a degenerated EG with a 'critical' heavy point on a cut, we can move this point (and the attached LoIs) a little bit outwards.

In the next chapter, a formalization of Peirce's EGs is given. The idea of this formalization is obtained from the discussion of Peirce's EGs in this chapter and has already been introduced in an informal manner: Peirce's EGs will be formalized as mathematical graphs with vertices, (labeled) edges and cuts. It has already been said that in this formalization, although we have in Peirce's graphs identity spots which are placed on cuts, it is reasonable to provide a formalization where vertices cannot be placed vertices *on* cuts. Moreover, it will turn out that the formalization is in fact 'only' a formalization of non-degenerated EGs. Peirce discussed and sometimes used degenerated EGs, but our discussion shows that these graphs can be replaced by equivalent non-degenerated EGs, that is, degenerated EGs can be dismissed. Thus the ongoing formalization grasps the whole realm of Peirce's EGs.

12

Syntax for Existential Graphs

In this chapter, the syntax for our formalization of existential graphs is provided. We have already discussed that an existential graph may have different representations, depending on our choice of identity spots a LoI is composed off (see the discussion in Sect. 11.1). For this reason, the definition of formal existential graphs is done in two steps: First, formal existential graph *instances* are defined. An existential graph instance (EGI) is *one* possible formalization of an EG where we have for each LoI chosen a number of identity spots. Depending of this choice, an EG has different EGI which can be understood as formalization of this EG. The *class* of all these EGIs will be the formalization of the EG.

The underlying structure of EGIs are so-called called RELATIONAL GRAPHS. An EGI is a relational graph whose edges are additionally labeled with predicate names. In the part 'Extending the System', the expressivity of EGIs is extended by adding object names or query markers (which can compared to *free* variables in $\mathcal{FO}$), and these extensions are obtained from EGIs by extending the labeling function. For this reason, we first define in Sect. 12.1 relational graphs and investigate their structure. In Sect. 12.2, the labeling of the edges with relation names is added to these graphs: The resulting graphs are EGIs. Then, in Sect. 12.3, some further syntactical notations for EGIs like *subgraph* etc. are introduced. Finally, in Sect. 12.4, we define formal existential graphs as sets of EGIs which can be mutually transformed into each other by a set of very simple rules.

12.1 Relational Graphs with Cuts

As we have already discussed in the last chapter, the underlying structures for our formalization of EGs will be mathematical graphs with directed hyperedges and cuts. Based on the conventions of graph theory, these structures should be called DIRECTED MULTI-HYPERGRAPHS WITH CUTS, but as this is a rather complicated technical term, we will call them RELATIONAL GRAPHS WITH CUTS or, even more simply, RELATIONAL GRAPHS instead.

In this section, the basic definitions and properties for relational graphs with cuts are presented.

Definition 12.1 (Relational Graphs with Cuts). *A* RELATIONAL GRAPH WITH CUTS *is a structure* $(V, E, \nu, \top, Cut, area)$, *where*

- V, E *and* Cut *are pairwise disjoint, finite sets whose elements are called* VERTICES EDGES *and* CUTS, *respectively,*
- $\nu : E \to \bigcup_{k \in \mathbb{N}_0} V^k$ *is a mapping,*[1]
- $\top$ *is a single element with* $\top \notin V \cup E \cup Cut$, *called the* SHEET OF ASSERTION, *and*
- $area : Cut \cup \{\top\} \to \mathfrak{P}(V \cup E \cup Cut)$ *is a mapping such that*
  a) $c_1 \neq c_2 \Rightarrow area(c_1) \cap area(c_2) = \emptyset$,
  b) $V \cup E \cup Cut = \bigcup_{d \in Cut \cup \{\top\}} area(d)$,
  c) $c \notin area^n(c)$ *for each* $c \in Cut \cup \{\top\}$ *and* $n \in \mathbb{N}$ *(with* $area^0(c) := \{c\}$ *and* $area^{n+1}(c) := \bigcup\{area(d) \mid d \in area^n(c)\}$*).*

For an edge $e \in E$ *with* $\nu(e) = (v_1, \dots, v_k)$ *we set* $|e| := k$ *and* $\nu(e)\big|_i := v_i$. *Sometimes, we will write* $e\big|_i$ *instead of* $\nu(e)\big|_i$, *and* $e = (v_1, \dots, v_k)$ *instead of* $\nu(e) = (v_1, \dots, v_k)$. *We set* $E^{(k)} := \{e \in E \mid |e| = k\}$.

For $v \in V$ *let* $E_v := \{e \in E \mid \exists i.\nu(e)\big|_i = v\}$. *Analogously, for* $e \in E$ *let* $V_e := \{v \in V \mid \exists i.\nu(e)\big|_i = v\}$. *For an edge* e *with* $\nu(e) = \{(v)\}$, *we write moreover* $v_e := v$.

The elements of $Cut \cup \{\top\}$ *are called* CONTEXTS.

As for every $x \in V \cup E \cup Cut$ *we have exactly one context* $c \in Cut \cup \{\top\}$ *with* $x \in area(c)$, *we can write* $c = area^{-1}(x)$ *for every* $x \in area(c)$, *or even more simple and suggestive:* $c = ctx(x)$.

In particular the empty graph, i.e. the empty sheet of assertion, exists. Its form is $\mathfrak{G}_\emptyset := (\emptyset, \emptyset, \emptyset, \top, \emptyset, \emptyset)$.

We have to extend some notations we have already introduced for formal alpha graphs, like the order $\leq$ on the elements of the graphs, subgraphs, isomorphisms etc., to relational graphs as well. We start with the order $\leq$, which is a canonical extension of the corresponding definition for formal alpha graphs (see Def. 7.3).

[1] The union $\bigcup_{k \in \mathbb{N}_0} V^k$ is often denoted by V^*. We do not adopt this notation, as V^* will be used in this treatise to denote the set of all generic vertices.

Definition 12.2 (Ordering on the Contexts, Enclosing Relation). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$ be a relational graph with cuts. We define a mapping $\beta : V \cup E \cup Cut \cup \{\top\} \to Cut \cup \{\top\}$ by*

$$\beta(x) := \begin{cases} x \text{ for } x \in Cut \cup \{\top\} \\ ctx(x) \text{ for } x \in V \cup E \end{cases} ,$$

and set $x \leq y :\Longleftrightarrow \exists n \in \mathbb{N}_0 . \beta(x) \in area^n(\beta(y))$ for $x, y \in V \cup E \cup Cut \cup \{\top\}$.

We define $x < y :\Longleftrightarrow x \leq y \wedge y \nleq x$ and $x \lneq y :\Longleftrightarrow x \leq y \wedge y \neq x$. . For a context $c \in Cut \cup \{\top\}$, we set furthermore

$$\leq[c] := \{x \in V \cup E \cup Cut \cup \{\top\} \mid x \leq c\} \text{ and}$$
$$\lneq[c] := \{x \in V \cup E \cup Cut \cup \{\top\} \mid x \lneq c\} .$$

Every element x of $\bigcup_{n \in \mathbb{N}} area^n(c)$ is said to be ENCLOSED BY *c, and vice versa: c is said to* ENCLOSE *x. For every element of $area(c)$, we say more specifically that it is* DIRECTLY ENCLOSED BY *c.*

Analogously to Alpha, we have that x is enclosed by a cut c if and only if $x \lneq c$ (see Lem. 7.4), and we obtain the following corollary:

Corollary 12.3 ($\leq$ Induces a Tree on the Contexts). *For a relational graph with cuts $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$, $\leq$ is a quasiorder. Furthermore, $\leq|_{Cut \cup \{\top\}}$ is an order on $Cut \cup \{\top\}$ which is a tree with the sheet of assertion $\top$ as greatest element.*

The ordered set of contexts $(Cut \cup \{\top\}, \leq)$ can be considered to be the 'skeleton' of a relational graph. According to Def. 12.1, each element of the set $V \cup E \cup Cut \cup \{\top\}$ is placed in exactly one context c (i.e. $x \in area(c)$). This motivates the next definition.

The next definition corresponds to Def. 7.12 for formal alpha graphs.

Definition 12.4 (Evenly/Oddly Enclosed, Pos./Neg. Contexts). *Let $\mathfrak{G} = (V, E, \nu, \top, Cut, area)$ be a relational graph with cuts. Let x be an element of $V \cup E \cup Cut \cup \{\top\}$ and set $n := |\{c \in Cut \mid x \in \leq[c]\}|$. If n is even, x is said to be* EVENLY ENCLOSED*, otherwise x is said to be* ODDLY ENCLOSED.

The sheet of assertion $\top$ and each oddly enclosed cut is called a POSITIVE CONTEXT*, and each an evenly enclosed cut is called* NEGATIVE CONTEXT.

It will turn out in the definition of the semantics that graphs in which vertices exist which deeper nested than some edge they are incident with cannot be evaluated (see the short discussion after Def. 13.4). This is captured by the following definition.

Definition 12.5 (Dominating Nodes). *If $ctx(e) \leq ctx(v)$ ($\Leftrightarrow e \leq v$) for every $e \in E$ and $v \in V_e$, then $\mathfrak{G}$ is said to have* DOMINATING NODES.

12.2 Existential Graph Instances

Existential graph instances (EGIs) are obtained from relational graphs by additionally labeling the edges with names for relations. To start, we have to define the set of these names, i.e. we define the underlying alphabet for EGIs. Of course, as lines of identity are essential in EGs, this alphabet must contain a symbol for identity. Then, using this alphabet, we can define EGIs on the basis of relational graphs. This is done with the following two definitions.

Definition 12.6 (Alphabet). *A pair* $\mathcal{A} := (\mathcal{R}, ar : \mathcal{R} \to \mathbb{N}_0)$ *is called an* ALPHABET. *The elements of* $\mathcal{R}$ *are called* RELATION NAMES, *the function ar assigns to each* $R \in \mathcal{R}$ *its* ARITY $ar(R)$. *Let* $\doteq \in \mathcal{R}$ *with* $ar(\doteq) = 2$ *be a special name which is called* IDENTITY.[2]

We will often more easily say that $\mathcal{R}$ *is the alphabet, without mentioning the arity-function.*

Definition 12.7 (Existential Graph Instance). *Let* $\mathcal{A}$ *be an alphabet. An* EXISTENTIAL GRAPH INSTANCE (EGI) OVER $\mathcal{A}$ *is a structure* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$*, where*

- $(V, E, \nu, \top, Cut, area)$ *is a relational graph with cuts and dominating nodes, and*
- $\kappa : E \to \mathcal{R}$ *is a mapping such that* $|e| = ar(\kappa(e))$ *for each* $e \in E$.

The set E *of edges is partitioned into* $E^{id} := \{e \in E \mid \kappa(e) = \doteq\}$ *and* $E^{nonid} := \{e \in E \mid \kappa(e) \neq \doteq\}$. *The elements of* E^{id} *are called* IDENTITY-EDGES. *Moreover, If* e *is an identity-edge with* $ctx(e) = ctx(e\big|_1)$ *or* $ctx(e) = ctx(e\big|_2)$*, then* e *is called* STRICT IDENTITY-EDGE. *The vertices, edges and cuts of an EGI will be called the* ELEMENTS *of the EGI. The system of all EGIs over* $\mathcal{A}$ *will be denoted by* $\mathcal{EGI}^{\mathcal{A}}$.

In the part 'Extending the System', the last definitions will be extended.

In the following, we will introduce mathematical definitions for the terms *ligature* and *hook*, which have no counterparts in Alpha. A ligature will be, roughly speaking, a set of vertices and identity edges between these vertices, i.e., a mathematical graph. For this reason, we first have to recall some basic notations of mathematical graph theory, as they will be used in this treatise.

An DIRECTED MULTIGRAPH is a structure (V, E, ν) of VERTICES $v \in V$ and EDGES $e \in E$. The mapping ν assigns to each edge e the pair (v_1, v_2) of its incident vertices. Given an EGI $(V, E, \nu, \top, Cut, area, \kappa)$, our aim is to describe

[2] We will usually use the common symbol '=' instead of '$\doteq$', but as we use the symbol '=' in the meta-language, too, sometimes it will be better to use the symbol '$\doteq$' in order to distinguish it from the meta-level '='.

ligatures as subgraphs of $(V, E^{id}, \nu\big|_{E^{id}})$, that is why we start with *directed* multigraphs. Nonetheless, the orientation of an identity edge has no significance (recall that we have a transformation rule which allows to change the orientation of identity edges), thus the remaining definitions are technically defined for directed multigraphs, but they treat edges as if they had no direction. In order to ease the notational handling of identity-edges in ligatures, we introduce the following conventions: If e is an identity-edge which connects the vertices v_1 and v_2, i.e. we have $e = (v_1, v_2)$ or $e = (v_2, v_1)$, we will write $e = \{v_1, v_2\}$ to indicate that the orientation of the edge does not matter, or we will even use use an infix notation for identity-edges, i.e. we will write $v_1\, e\, v_2$ instead of $e = \{v_1, v_2\}$.

A SUBGRAPH of (V, E, ν) is a directed multigraph (V', E', ν') which satisfies $V' \subseteq V$, $E' \subseteq E$ and $\nu' = \nu\big|_{E'}$. A PATH in (V, E, ν) is a subgraph (V', E', ν') with $V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, e_{n-1}\}$ such that we have $v_1\, e_1\, v_2\, e_2\, v_3\, \ldots\, v_{n-1}\, e_{n-1}\, v_n$, and we will say that the path CONNECTS v_1 and v_n. If we have moreover $n > 1$, $v_1 = v_n$ and all vertices $v_2, \ldots, v_{n-1}$ are distinct from each other and v_1, v_n , then the path is called a CYCLE (some authors assume $n > 2$ instead of $n > 1$, but for our purpose, $n > 1$ is the better choice). We say that (V, E, ν) is CONNECTED if for each two vertices $v_1, v_2 \in V$ there exists a path in (V, E, ν) which connects v_1 and v_2. A LOOP is a subgraph $(\{v\}, \{e\}, \{(e, (v, v))\}$, i.e., basically an edge joining a vertex to itself. A FOREST is a graph which neither contains cycles, nor loops. A TREE is a connected forest. It is well known that a connected graph (V, E, ν) is a tree iff we have $|V| = |E| + 1$. A LEAF of a tree (V, E, ν) is a vertex which is incident with exactly one edge.

As already said, for a given EGI $(V, E, \nu, \top, Cut, area, \kappa)$, our aim it to introduce ligatures as subgraphs of $(V, E^{id}, \nu\big|_{E^{id}})$. Strictly speaking, an edge is given by an element $e \in E$ together with $\nu(e)$, which assigns to e its incident vertices, thus we should incorporate the mapping ν into this definition. To ease the notation, we will omit the mapping ν, but we agree that ν is implicitly given. That is, if we speak about a subgraph (W, F) of (V, E^{id}), it is meant that we mean the subgraph $(W, F, \nu\big|_F)$ of $(V, E^{id}, \nu\big|_{E^{id}})$. Now it is easy to define ligatures as follows:

Definition 12.8 (Ligature). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. Then we set $Lig(\mathfrak{G}) := (V, E^{id})$, and $Lig(\mathfrak{G})$ is called the* LIGATURE-GRAPH INDUCED BY $\mathfrak{G}$. *Each connected subgraph of (W, F) of $Lig(\mathfrak{G})$ is called a* LIGATURE OF $\mathfrak{G}$.

Note that the ligatures in an EGI which are loops or cycles correspond to the *closed*, heavily drawn lines in the corresponding Peirce graph. Moreover, please note that for each vertex $v \in V$, $(\{v\}, \emptyset)$ is a ligature. That is, single vertices can be considered ligatures as well.

Next, we provide a formal definitions for Peirce's *hooks*, and the basic operation of replacing a vertex on a hook.

Definition 12.9 (Hook). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. Each pair (e, i) with $e \in E$ and $1 \leq i \leq |e|$ is called a* HOOK OF e, *or* HOOK *for short. If v is a vertex with $e\big|_i = v$, then we say that* THE VERTEX v IS ATTACHED TO THE HOOK (e, i).

A vertex v with is attached to more than two hooks is called a BRANCHING POINT. *The number of hooks v is attached to is called its number of* BRANCHES.

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ is an EGI, v be a vertex, $e = (v_1, \ldots, v_n)$ an edge and $1 \leq i \leq |e|$ with $v_i = v$, and let v' be a vertex with $ctx(v') \geq ctx(e)$. Let $\mathfrak{G}' := (V, E, \nu', \top, Cut, area, \kappa)$ be obtained from $\mathfrak{G}$, where ν' is defined as follows:

$$\nu'(f) = \nu(f) \text{ for all } f \neq e \quad , \text{ and } \quad \nu'(e) = (v_1, \ldots, v_{i-1}, v', v_{i+1}, v_n)$$

Then we say that $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by REPLACING v BY v' ON THE HOOK (e, i).

In order to work with EGIs, their mathematical representations are too clumsy and too difficult to handle. Hence one may prefer graphical representations OF EGIs. It has to be explained how EGIs are drawn. This shall be done now.

The different elements of an EGI are represented by different kinds of graphical items, namely *vertex-spots, edge-lines, cut-lines, signs which represent the relation-names, and numbers* (these terms will be introduced below). First of all, we agree that no graphical items may intersect, overlap, or touch, as long as it is explicitely allowed. Moreover, we agree that no further graphical items will be used.

We start with the representation of the cuts. Similar to alpha, each cut is represented by a closed, doublepoint-free and smooth curve which is called the CUT-LINE OF THE CUT. A cut-line separates the plane into two distinct regions: The inner and the outer region. Recall that we said that another item of the diagram is enclosed by this cut-line if and only if it is placed in the inner region of this cut-line. If c_1, c_2 are two cuts with $c_1 < c_2$, then the cut-line of c_1 has to be enclosed by the cut-line of c_2. Due to our first convention, cut-lines may not intersect, overlap, or touch. Note that it is possible to draw all cut-lines in the required manner because we have proven that the set of contexts of an EGI form a tree (see Lemma 12.3). If c is a cut, then the part of the plane which is enclosed by the cut-line of c, but which is not enclosed by any cut-line of a cut $d < c$ is called the AREA-SPACE OF c (the cut-line of c does not belong to the area-space of c). The part of the plane which is not enclosed by any cut-line is called the area-space of $\top$.

Each vertex v is drawn as a bold spot, i.e. $\bullet$, which is called VERTEX-SPOT OF v. This spot has to be placed on the area-space of $ctx(v)$. Of course different vertices must have different vertex-spots.

Now let $e = (v_1, \ldots, v_n)$ be an edge. We write the sign which represents $\kappa(e)$ on the area-space of $ctx(e)$. Then, for each $i = 1, \ldots, n$, we draw a non-closed and doublepoint-free line, called THE iTH EDGE-LINE OF e or THE EDGE-LINE BETWEEN v_i AND THE HOOK (e, i), which starts at the vertex-spot of v_i and ends close the to sign which represents $\kappa(e)$. Moreover, this line is labeled, nearby the sign which which represents $\kappa(e)$, with the number i. Edge-lines are allowed to intersect cut-lines, but an edge-line and a cut-line must not intersect more than once. This requirement implies that the edge-line of e intersects the cut-line of each cut c with $v_i > c \geq ctx(e\big|_2)$, and no further cut-lines are intersected.

If it cannot be misunderstood, the labels of the edge-line(s) of an edge e are often omitted.

For identity-edges, we have a further, separate convention: If $e = (v_1, v_2)$ is an identity-edge, it is furthermore allowed to replace the symbol '$\doteq$' by an simple line which connects the ends of the first and second edge-line of e which were former attached to the symbol '$\doteq$' (this connection has to be placed in the area where otherwise the symbol '$\doteq$' were). In this case, the labels of the edge-lines are omitted. This convention will become clear in the examples below.

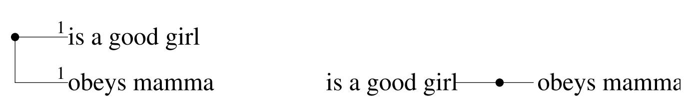
There may be graphs such that its edge-lines cannot be drawn without their crossing one another (i.e., they are not planar). For this reason, the intersection of edge-lines is allowed. But the intersection of edge-lines should be avoided, if it possible.

To illustrate these agreements, I provide some examples. The first three examples are adopted from Peirce's Cambridge Lectures, Lecture three.

We start with a simple EGI without cuts, where only one vertex is incident with two (unary) edges.

$$\begin{aligned}\mathfrak{G}_1 := (&\{v\}, \{e_1, e_2\}, \{(e_1, (v)), (e_2, (v))\}, \top, \emptyset, \{(\top, \{v, e_1, e_2\})\},\\ &\{(e_1, \text{is a good girl}), (e_2, \text{obeys mamma})\})\end{aligned}$$

Below you find two representations for this EGI. In the second one, the labels for the edges are omitted. As both edges are incident with one vertex, this causes no problems.



The next example is again a EGI without cuts and only one vertex. Here this vertex is incident twice with an edge.

$$\mathfrak{G}_2 := (\{v\},\ \{e_1, e_2\},\ \{(e_1, (v)), (e_2, (v, v))\},\ \top,\ \emptyset,\ \{(\top, \{v, e_1, e_2\})\},\\ \{(e_1, \text{is a good girl}), (e_2, \text{obeys the mamma of})\})$$

This EGI can be represented as follows:

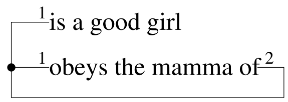
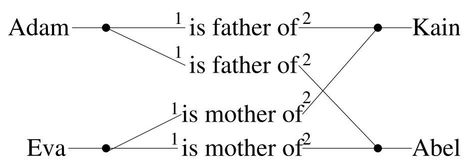


Fig. 12.1. One representation for $\mathfrak{G}_2$

We allow edge-lines to cross each other. For example, consider the following representation of an EGI:

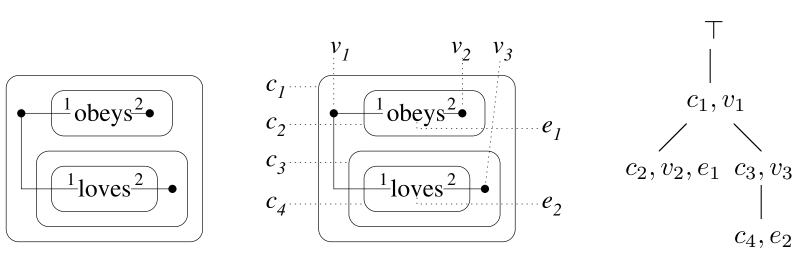


It is well known from graph-theory that not every graph has a planar representation. For this graph, a planar representation would be possible, but the given representation with crossing edge-lines shows nicely the symmetry of the EGI. Crossing edge-lines cause no troubles in the reading of the diagram, and as they are sometimes inevitable, they are allowed in the diagrammatic representations of EGIs.

The next example is a more complex EGI with cuts.

$$\mathfrak{G}_3 := (\{v_1, v_2, v_3\},\ \{e_1, e_2\},\ \{(e_1, (v_1, v_2)), (e_2, (v_1, v_3))\},\ \top,\ \{c_1, c_2, c_3, c_4\},\\ \{(\top, \{c_1\}), (c_1, \{v_1, c_2, c_3\}), (c_2, \{v_2, e_1\}), (c_3, \{v_3, c_4\}), (c_4, \{e_2\})\},\\ \{(e_1, \text{obeys}), (e_2, \text{loves})\})$$

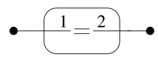
Below, the left diagram is a possible representation of $\mathfrak{G}_3$. In the right diagram on the right, I have sketched furthermore assignments of the elements (the vertices, edges, and cuts) of the EGI to the graphical elements of the diagram. Finally, on the right, the order $\leq$ for $\mathfrak{G}_3$ is depicted.



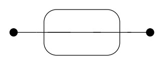
The next example shall explain the two different conventions for identity edges. We had already the first examples in the last chapter on page 110. Consider the following EGI:

$$\mathfrak{G}_4 := (\{v_1, v_2\},\ \{e\},\ \{(e, (v_1, v_2))\},\ \top,\ \{c\}, \{(\top, \{v_1, v_2\}), (c, \{e\})\}, \{(e, \doteq)\})$$

It is a formalization of the well-known graph claiming there exists at least two things. Due to our normal convention for drawing edges, it is graphically represented as follows:



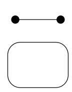
The second convention says that the sign '$\doteq$' may be replaced by a simple line. For our example, we obtain:



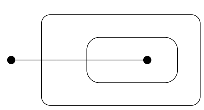
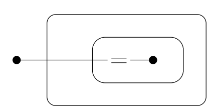
Now the whole identity-edge is represented by a simple line.

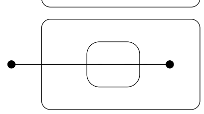
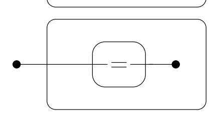
First of all, in this this representation, we loose the information about the orientation of the edge, so this convention can only be used when the orientation of the edge is of no significance. But this will be often the case.

More importantly, as the sign '$\doteq$' we had replaced was placed inside the cut-line, it is important that the new, simple line goes through the inner area of the cut-line as well. The following diagram, in which we have a simple line connecting the vertex-spots as well, is therefore *not* a representation of $\mathfrak{G}_4$:



If we follow the convention in this strict sense, even if an identity-edge is drawn as a simple line, it is still possible to read from the diagram in which cut the identity-edge is placed. For example,

 can only be obtained from , and

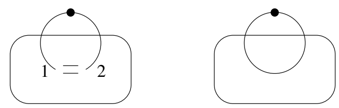
 can only be obtained from .

None of the identity-edges can be placed in a different cut, because we consider only graphs with dominating nodes.

This holds even for identity-edges e with $e\big|_1 = e\big|_2$. Their edge-lines appear to be closed. For example, consider

$$\mathfrak{G}_5 := (\{v\},\, \{e\},\, \{(e,(v,v))\},\, \top,\, \{c\},\, \{(\top,\{v\}),(c,\{e\})\},\, \{(e,\doteq)\})$$

Due to the conventions, this graph may be represented as follows:

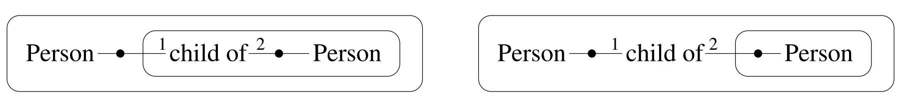


Although the edge-line cannot be mistaken with the cut-line, the right diagram lacks readability, thus one should prefer the left diagram instead.

From the examples it should be clear that each EGI can be represented as a diagram. Similar to Alpha, it shall shortly be investigated which kind of diagrams occur as diagrams of an EGI. We have seen that a diagram is made up of vertex-spots, edge-lines, cut-lines, signs which represent the relation-names, and numbers. Again, these signs may not intersect, touch, or overlap, with the exception that we allow two edge-lines to cross, and we allow edge-lines to cross cut-lines, but not more than once. A diagram of an EGI in which identity edges are drawn to the usual convention of edges satisfies:

1. If the sign of a relation-name of an n-ary relation occurs in the diagram, then there are n edge-lines, numbered with the numbers $1, \ldots, n$, which are attached to the relation-sign, and each of these edge-lines end in a vertex spot. It is allowed that different edge-lines end the same vertex. Only such edge-lines which go from a vertex to a relation-sign may occur.
2. If a vertex-spot is given which is connected to a relation-sign with an edge-line, and if the vertex is enclosed by a cut-line, then the relation-sign is enclosed by this cut-line, too.

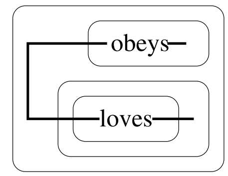
The second condition is a restriction for the diagrams which reflects that EGIs are relation-graphs with domination nodes. For example, the left diagram is a diagram of an EGI, while the right graph is not:



We already used the condition of dominating nodes to provide the second convention for drawing identity edges. So, even if identity edges are drawn due the second condition, from each properly drawn diagram (that is, it has to satisfy the above given conditions) we can reconstruct up to isomorphism (which will we be defined canonically in Def. 12.11) and the orientation of those identity-edges which are drawn as simple lines) the underlying EGI. Thus, from now on, we can use the diagrammatic representation of EGIs in mathematical proofs.

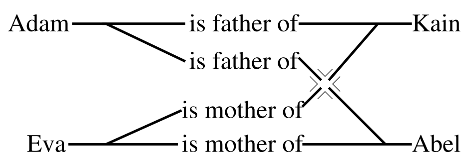
Next, the correspondence between the diagrams of Peirce's Beta graphs and the diagrams of EGIs shall be investigated.

If a diagram of an EGI is given, we can transform it as follows: We draw all edge-lines bold, such that all vertex-spots at the end of an edge-line cannot be distinguished from this edge-line. For example, for $\mathfrak{G}_3$ we get the right diagram of an Peircean EG (the labels of the edges are omitted, and this diagram is exactly one of the diagrams of Peirce's Cambridge Lectures):



It is easy to see that the diagrams we obtain this way are precisely the diagrams of non-degenerated existential graphs (including Peirce graphs with isolated identity-spots).

The only problem in the transformation of diagrams of EGIs into Peirce diagrams which can occur are crossing edge-lines. If we draw these edge-lines bold, then it would seem that we had intersecting lines of identity. Peirce has realized that there may be EGs which cannot be drawn on a plane without the intersection of LoIs.[3] For this reason, he introduced a graphical device called 'bridge', which has to be drawn in this case (see 4.561). For our example with intersecting edge-lines above, we draw



12.3 Further Notations for Existential Graph Instances

Similar to Alpha, we have to define *subgraphs* of EGIs. The basic idea is the same like in Alpha, but he notation of subgraphs in Alpha has to be extended to Beta in order to capture the extended syntax of Beta, particularly edges which do not occur in formal alpha graphs. First of all, if a subgraph contains an edge, then it has to contain all vertices which are incident with the edge, too. Moreover, we distinguish between *subgraphs* and *closed subgraphs*: If a subgraph is given such that for each vertex of this subgraph, all incident edges belong to the subgraph, too, then the subgraph is called a CLOSED SUBGRAPH. The notation of a subgraph will become precise through the following definition.[4]

[3] This is a general problem of mathematical graphs which is extensively investigated in graph theory.

[4] In this definition, subgraphs are first defined for relational graph with cuts, then for EGIs. The reason is that we will slightly extend the syntax for EGIs in Def. 24.1 by adding a second labeling function. Strictly speaking, the following definition

Definition 12.10 (Subgraphs). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$ be a relational graph with cuts. A graph $\mathfrak{G}' := (V', E', \nu', \top', Cut', area')$ is a* SUBGRAPH OF $\mathfrak{G}$ IN THE CONTEXT $\top'$ *iff*

- $V' \subseteq V$, $E' \subseteq E$, $Cut' \subseteq Cut$ *and* $\top' \in Cut \cup \{\top\}$,
- $\nu' = \nu\big|_{E'}$ *(particularly, the restriction ν' of ν to E' is well defined),*
- $area'(\top') = area(\top') \cap (V' \cup E' \cup Cut')$ *and* $area'(c') = area(c')$ *for each* $c' \in Cut'$,
- $ctx(x) \in Cut' \cup \{\top'\}$ *for each* $x \in V' \cup E' \cup Cut'$, *and*
- $V_{e'} \subseteq V'$ *for each edge* $e' \in E'$.

If we additionally have $E_{v'} \subseteq E'$ for each vertex $v' \in V'$, then $\mathfrak{G}'$ is called CLOSED SUBGRAPH OF $\mathfrak{G}$ IN THE CONTEXT $\top'$.

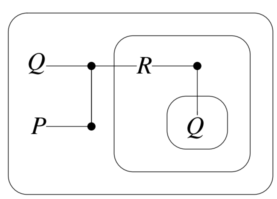
Now let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. We call a graph $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ a SUBGRAPH OF $\mathfrak{G}$ IN THE CONTEXT $\top'$, *iff $(V', E', \nu', \top', Cut', area')$ is a subgraph of $(V, E, \nu, \top, Cut, area)$ in the context $\top'$ which respects the labeling, i.e. if $\kappa' = \kappa\big|_{E'}$ holds.*

In both cases (relational graph with cuts and EGIs), we write $\mathfrak{G}' \subseteq \mathfrak{G}$ and $area^{-1}(\mathfrak{G}') = \top'$ resp. $ctx(\mathfrak{G}') = \top'$.

Similar to Alpha (see Lem. 7.10), we get $\lneq[c'] \subseteq E' \cup V' \cup Cut'$ for each $c' \in Cut'$. In particular, Cut' is an ideal in $Cut \cup \{\top\}$, but in general, $Cut' \cup \{\top'\}$ is *not* an ideal in $Cut \cup \{\top\}$ (there may be contexts $d \in area(\top')$ which are not a element of $Cut' \cup \{\top'\}$).

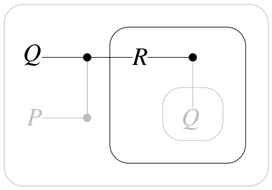
A subgraph is, similar to elements of $E \cup V \cup Cut \cup \{\top\}$, placed in a context (namely $\top'$). Thus, we will say that the subgraph is DIRECTLY ENCLOSED by $\top'$, and it is ENCLOSED by a context c if and only iff $\top' \leq c$. Moreover, we can apply Def. 12.4 to subgraphs as well. Hence we we distinguish whether a subgraph is evenly or oddly enclosed.

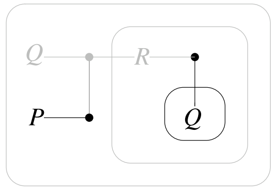
To get an impression of subgraphs of EGI, an example will be helpful. Consider the following graph:

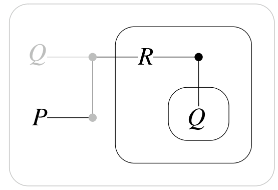


of subgraphs cannot be applied to these graphs, but if we define subgraphs first for relational graph with cuts, it should be clear how this definition should be modified for the extended EGIs of Def. 24.1.

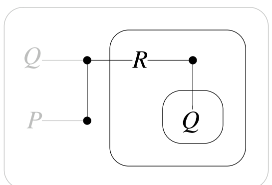
In the following diagrams, we consider some substructures of this graph. All items which do not belong to the substructure are lightened.

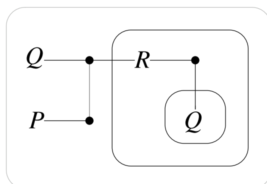


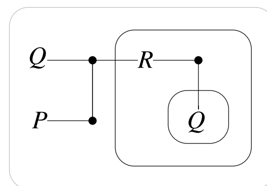




In the first example, the marked substructure contains a cut d, but it does not contain all what is written inside d. In the second example, the vertex which is linked to the relation P is not enclosed by any context of the substructure. In the last example, the substructure contains two edges, where a incident vertex does not belong to the substructure. Hence, in none of the examples above the marked substructure is a subgraph.







These marked substructures are subgraphs in the outermost cut. The first two subgraphs are not closed: In the first example, we have two vertices with incident edges Q and P which do not belong to the subgraph, and in the second example, the identity edge which is incident with these vertices does not belong to the subgraph. The third subgraph is closed.

In Chpt. 21, it is investigated how subgraphs can be diagrammatically depicted.

Analogously to Alpha, isomorphisms between EGIs are canonically defined. Similar (and for the same reasons) as in the definition of subgraphs, we first define isomorphisms for relational graph with cuts, then this definition is extended for EGIs.

Definition 12.11 (Isomorphism). *Let* $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$, $i = 1, 2$, *be relational graphs with cuts.*

Then $f = f_V \,\dot\cup\, f_E \,\dot\cup\, f_{Cut}$ *is called* ISOMORPHISM, *if*

- $f_V : V_1 \to V_2$ *is bijective,*
- $f_E : E_1 \to E_2$ *is bijective,*
- $f_{Cut} : Cut_1 \cup \{\top_1\} \to Cut_2 \cup \{\top_2\}$ *is bijective with* $f_{Cut}(\top_1) = \top_2$,

such that the following conditions hold:

- *Each* $e = (v_1, \ldots, v_n) \in E_1$ *satisfies* $f_E(v_1, \ldots, v_n) = (f_V(v_1), \ldots, f_V(v_n))$ *(edge condition)), and*
- $f[area_1(c)] = area_2(f(c))$ *for each* $c \in Cut_1 \cup \{\top_1\}$ *(cut condition))*
  (with $f[area_1(c)] = \{f(k) \mid k \in area_1(c)\}$*).*

Now let $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i, \kappa_i)$*,* $i = 1, 2$ *be two EGIs. Then* $f = f_V \dot\cup f_E \dot\cup f_{Cut}$ *is called* ISOMORPHISM*, iff* f *is an isomorphism for the underlying relational graphs with cuts* $\mathfrak{G} := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ *which respects the labeling, i.e. if* $\kappa_1(e) = \kappa_2(f_E(e))$ *for all* $e \in E_1$ *holds.*

Furthermore, again a notation of a partial isomorphism is needed.

Definition 12.12 (Partial Isomorphism). *For* $i = 1, 2$*, let a relational graph with cut* $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ *and a context* $c_i \in Cut_i \cup \{\top_i\}$ *be given. For* $i = 1, 2$*, we set*

1. $V_i' := \{v \in V_i \mid v \not\leq c_i\}$*, and*
2. $E_i' := \{e \in E_i \mid e \not\leq c_i\}$*,*
3. $Cut_i{}' := \{d \in Cut_i \cup \{\top_i\} \mid d \not< c_i\}$

Let $\mathfrak{G}_i'$ *be the restriction of* $\mathfrak{G}_i$ *to these sets, i.e., for* $area_i{}' := area_i\big|_{Cut_i{}'}$*, let* $\mathfrak{G}_i' := (V_i', E_i', \nu\big|_{E_i'}, \top_i, Cut_i', area_i')$*. If* $f = f_{V_1'} \dot\cup f_{E_1'} \dot\cup f_{Cut_1'}$ *is an isomorphism between* $\mathfrak{G}_1'$ *and* $\mathfrak{G}_2'$ *with* $f_{Cut}(c_1) = c_2$*, then* f *is called* (PARTIAL) ISOMORPHISM FROM $\mathfrak{G}_1$ TO $\mathfrak{G}_2$ EXCEPT FOR THE CONTEXTS $c_1 \in Cut_1 \cup \{\top_1\}$ AND $c_2 \in Cut_2 \cup \{\top_2\}$.

Analogously to the last definition, partial isomorphisms between EGIs are partial isomorphisms between the underlying relational graph with cuts which respect κ*.*

To this definition, the same remarks as for its counterpart in Alpha apply (see the remarks after Def. 7.11).

Finally, we have to define juxtapositions for EGIs.

Definition 12.13 (Juxtaposition). *For an* $n \in \mathbb{N}_0$ *and for each* $i = 1, \ldots, n$*, let* $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ *be a relational graph with cuts. The* JUXTAPOSITION OF THE $\mathfrak{G}_i$ *is defined to be the following graph* $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$*:*

- $V := \bigcup_{i=1,\ldots,n} V_i \times \{i\}$,
- $E := \bigcup_{i=1,\ldots,n} E_i \times \{i\}$,
- *for* $e = (v_1, \ldots, v_n) \in E$*, let* $\nu((e, i)) := ((v_1, i), \ldots, (v_n, i))$,
- $\top$ *arbitrary element,*

- $Cut := \bigcup_{i=1,\ldots,n} Cut_i \times \{i\}$,
- *area is defined as follows:* $area((c,i)) = area_i(c) \times \{i\}$ *for* $c \in Cut_i$*, and* $area(\top) = \bigcup_{i=1,\ldots,n} area_i(\top_i) \times \{i\}$.

Analogously to the last definitions, if $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ *is an EGI for* $i = 1, \ldots, n$*, the* JUXTAPOSITION OF THE $\mathfrak{G}_i$ *is defined to be the graph* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *where* $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$ *is the juxtaposition of the graphs* $(V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ *which respects* κ*, i.e. we have* $\kappa(e,i) := \kappa_i(e)$ *for all* $e \in E$ *and* $i = 1, \ldots, n$.

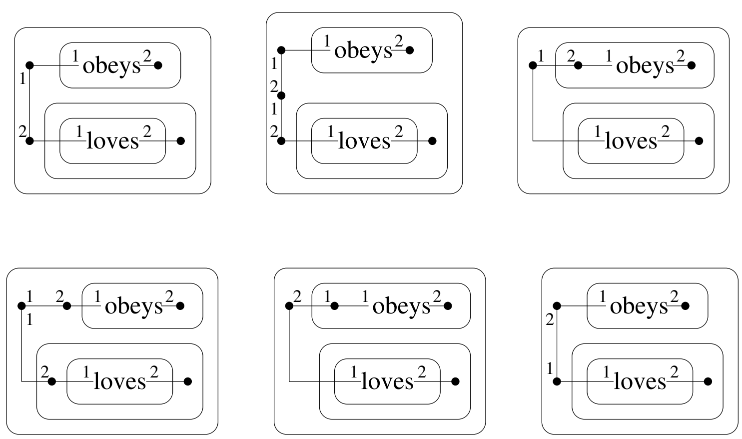
In the graphical notation, the juxtaposition of graphs $\mathfrak{G}_i$ *(relational graphs with cuts or EGIs) is simply noted by writing the graphs next to each other, i.e. we write:*

$$\mathfrak{G}_1 \ \mathfrak{G}_2 \ \ldots \ \mathfrak{G}_n \qquad .$$

Again, similar to Alpha, the juxtaposition of an empty set of EGIs yields the empty graph, i.e., $(\emptyset, \emptyset, \emptyset, \top, \emptyset, \emptyset, \emptyset)$.

12.4 Formal Existential Graphs

In Sect. 12.2, I have argued that EGIs can be represented by diagrams, and that we can reconstruct an EGI from a diagram up to isomorphism. Moreover, we have discussed how the diagrams of EGIs can be transformed into Peircean diagrams. Due to our discussion in the last chapter that the number of identity spots which form a LoI is not fixed, but up to our choice, it has to be expected that different EGIs may yield the same Peirce diagram as well. For example, the next EGIs yield the same Peircean diagram as $\mathfrak{G}_3$ from Sect. 12.2:



Neither the diagrams of EGIs, nor the Peircean graphs are defined mathematically, so we cannot *prove* any correspondence between them. But from the example above, it should be clear that two EGIs yield the same Peirce graph if they can mutually be transformed into each other by one or more applications of the following informally given rules:

- **isomorphism**

  An EGI may be substituted by an isomorphic copy of itself.

- **changing the orientation of an identity edge**

  If $e = (v_1, v_2)$ is an identity edge of an EGI, then its orientation may be changed, i.e., v_1 and v_2 are exchanged.

- **adding a vertex to a ligature**

  Let $v \in V$ be a vertex which is attached to a hook (e, i). Furthermore let c be a context with $ctx(v) \geq c \geq ctx(e)$. Then the following may be done: In c, a new vertex v' and a new identity-edge between v and v' is inserted. On (e, i), v is replaced by v'.

- **removing a vertex from a ligature**

  The rule 'adding a vertex to a ligature' may be reversed.

The rules 'changing the orientation of an identity edge', 'adding a vertex to a ligature' and 'removing a vertex from a ligature' will be summarized by 'transforming a ligature'.

In the next definition, a formal elaboration of these rules is provided. In this definition, the term 'fresh' is used for vertices, edges or cuts, similar to the following known use in logic for variables: Given a formula, a 'fresh' variable is a variable which does not occur in the formula. Analogously, given a graph $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, a vertex, edge or cut x is called FRESH if we have $x \notin V \dot\cup E \dot\cup Cut \cup \{\top\}$.

Definition 12.14 (Transformation Rules for Ligatures). *The transformation rules for ligatures for EGIs over the alphabet $\mathcal{R}$ are:*

- **isomorphism**

  Let $\mathfrak{G}, \mathfrak{G}'$ be EGIs such that there exists an isomorphism from $\mathfrak{G}$ to $\mathfrak{G}'$. Then we say that $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by SUBSTITUTING $\mathfrak{G}$ BY THE ISOMORPHIC COPY $\mathfrak{G}'$ OF ITSELF.

- **changing the orientation of an identity edge**

  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. Let $e \in E^{id}$ be an identity edge with $\nu(e) = (v_1, v_2)$. Let $\mathfrak{G} := (E, V, \nu', \top, Cut, area, \kappa)$ with

  $$\nu'(f) = \nu(f) \text{ for } f \neq e \text{ , and } \nu'(e) = (v_2, v_1)$$

  Then we say that $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by CHANGING THE ORIENTATION OF THE IDENTITY EDGE e.

- **adding a vertex to a ligature/removing a vertex from a ligature**
  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. Let $v \in V$ be a vertex which is attached to a hook (e, i). Let v' be a fresh vertex and e' be a fresh edge. Furthermore let c be a context with $ctx(v) \geq c \geq ctx(e)$. Now let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be the following graph:
  - $V' := V \dot{\cup} \{v'\}$,
  - $E' := E \dot{\cup} \{e'\}$,
  - $\nu'(f) = \nu(f)$ *for all* $f \in E$ *with* $f \neq e$, $\nu'(e') = (v, v')$, $\nu'(e)\big|_j := \nu(e)\big|_j$ *for* $j \neq i$, *and* $\nu'(e)\big|_i := v'$,
  - $\top' := \top$
  - $Cut' := Cut$
  - $area'(c) := area(c) \dot{\cup} \{v', e'\}$, *and for* $d \in Cut' \cup \{\top'\}$ *with* $d \neq c$ *we set* $area'(d) := area(d)$, *and*
  - $\kappa' := \kappa \dot{\cup} \{(e', \doteq)\}$.

  Then we say that $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by ADDING A VERTEX TO A LIGATURE *and $\mathfrak{G}$ is obtained from $\mathfrak{G}'$ by* REMOVING A VERTEX FROM A LIGATURE.

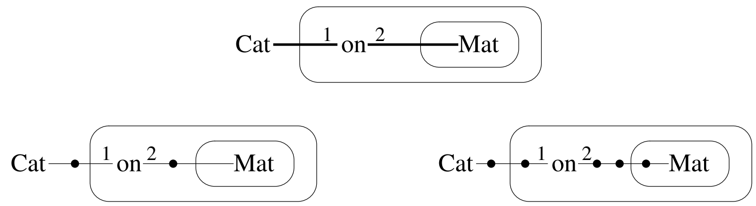
Definition 12.15 (Formal Existential Graphs). *Let $\mathfrak{G}_a, \mathfrak{G}_b$ be EGIs over a given alphabet $\mathcal{A}$. We set $\mathfrak{G}_a \sim \mathfrak{G}_b$, if $\mathfrak{G}_b$ can be obtained from $\mathfrak{G}_a$ with the four rules above (i.e., if there is a finite sequence $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ with $\mathfrak{G}_1 = \mathfrak{G}_a$ and $\mathfrak{G}_b = \mathfrak{G}_n$ such that each $\mathfrak{G}_{i+1}$ is derived from $\mathfrak{G}_i$ by applying one of the rules 'isomorphism', 'changing the orientation of an identity edge', 'adding a vertex to a ligature' and 'removing a vertex from a ligature').*

Each class $[\mathfrak{G}]_\sim$ is called an (FORMAL) EXISTENTIAL GRAPH *over $\mathcal{A}$. Formal existential graphs will usually be denoted by the letter $\mathfrak{E}$.*

To provide a simple example, a Peircean Beta graph and two elements (that is: EGIs) of the corresponding formal Beta graph are depicted in Fig. 12.4.

A factorization of formulas due to some specific transformations is folklore in mathematical logic. The classes of formulas are sometimes called *structures*, but using this term in this treatise may lead to misinterpretations (esp., as EGIs are already mathematical structures), thus we avoid this term.

There are no 'canonically given' discrete structures which formalize Peirce's Beta graphs, but for our mathematical elaboration of Peirce's Beta graphs, discrete structures are of course desirable. Factorizing the class of EGIs by the transformation rules for ligatures provides a means to formalize Peirce's graphs by classes of discrete structures. In the next chapters, nearly all mathematical work will be carried out with EGIs. Nonetheless, the relation $\sim$ can be understood in some respect to be an 'congruence relation', thus, it is – in some sense – possible to carry over the definitions for EGIs to definitions for formal



$$
\begin{aligned}
(&\{v_1, v_2\},\\
&\{e_1, e_2, e_3\},\\
&\{(e_1,(v_1)),(e_2,(v_1,v_2)),(e_3,(v_2))\},\\
&\top, \{c_1, c_2\},\\
&\{(\top, \{v_1, e_1, c_1\},\\
&\ \ (c_1, \{v_2, e_2, c_2\}),\\
&\ \ (c_2, \{e_3\})\},\\
&\{(e_1, \mathrm{Cat}), (e_2, \mathrm{on}), (e_3, \mathrm{Mat})\})
\end{aligned}
$$

$$
\begin{aligned}
(&\{v_1, v_2, v_3, v_4, v_5\},\\
&\{e_1, e_2, e_3, e_4, e_5, e_6\},\\
&\{(e_1,(v_1)),(e_2,(v_1,v_2)),\\
&\ \ (e_3,(v_2,v_3)),(e_4,(v_3,v_4))\},\\
&\ \ (e_5,(v_4,v_5)),(e_6,(v_5))\},\\
&\top, \{c_1, c_2\},\\
&\{(\top, \{v_1, e_1, c_1\},\\
&\ \ (c_1, \{v_2, v_3, v_4, e_2, e_3, e_4, c_2\}),\\
&\ \ (c_2, \{v_5, e_5, e_6\})\},\\
&\{(e_1, \mathrm{Cat}), (e_2, \doteq), (e_3, \mathrm{on})\\
&\ \ (e_4, \doteq), (e_5, \doteq), (e_6, \mathrm{Mat})\})
\end{aligned}
$$

Fig. 12.2. A Peirce graph and two corresponding EGIs

EGs. For example, we can say that $\mathfrak{E}_0$ is the juxtaposition of $\mathfrak{E}_1, \ldots, \mathfrak{E}_n$, if we have EGIs $\mathfrak{G}_0, \mathfrak{G}_1, \ldots, \mathfrak{G}_n$ with $\mathfrak{E}_i = [\mathfrak{G}_i]_\sim$ for each $0 \leq i \leq n$, and $\mathfrak{G}_0$ is the juxtaposition of $\mathfrak{G}_1, \ldots, \mathfrak{G}_n$. For more complicated definitions, we will have to elaborate further that $\sim$ is a 'congruence relation'. For example, in the next section, this will be done for the semantical entailment relation, that is, we will show that equivalent EGIs have the same meaning.

13

Semantics for Existential Graphs

The main treatises dealing with Peirce's Beta graphs provide a semantics for them either in an informal, naive manner (like [Rob73]) or by providing a mapping -let us call it Φ– of existential graphs to $\mathcal{FO}$-formulas (e.g. [Bur91a], [PS00], [Zem64], [Shi02a]). As Φ is a mapping of one syntactically given logical language into another syntactically given logical language, in my view the use of the term 'semantics' for Φ is not appropriate. Instead of this, a direct extensional semantics based on the relational structures known from Chpt. 18 is provided.

In the first section, it will be shown how EGIs are evaluated in relational structures. It should be expected that equivalent EGIs have the same meaning (i.e., their evaluation in arbitrary structures yield always the same result). This will be shown in the second section.

13.1 Semantics for Existential Graph Instances

EGIs are evaluated in the well-known *relational structures* as they are known from first order logic. We start this section with a definition of these structures.

Definition 13.1 (Relational Structures). *A* RELATIONAL STRUCTURE *or* RELATIONAL MODEL OVER AN ALPHABET $(\mathcal{R}, ar : \mathcal{R} \to \mathbb{N})$ *is a pair* $\mathcal{M} := (U, I)$ *consisting of a nonempty* UNIVERSE U *and a function* $I : \mathcal{R} \to \bigcup_{k\in\mathbb{N}} \mathfrak{P}(U^k)$ *such that* $I(R) \in \mathfrak{P}(U^k)$ *for* $ar(R) = k$*, and* $(u_1, u_2) \in I_{\mathcal{R}}(\doteq) \Leftrightarrow u_1 = u_2$ *for all* $u_1, u_2 \in U$*.*

The function I (the letter 'I' stands of course for 'interpretation') is the link between our language and the mathematical universe, i.e., it relates syntactical objects to mathematical entities.

When an EGI is evaluated in a relational structure (U, I), we have to assign objects of our universe of discourse U to its vertices. This is done – analogously to $\mathcal{FO}$ (see Def. 18.5) – by valuations.

Definition 13.2 (Partial and Total Valuations). *Let an EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given and let* (U, I) *be a relational structure over* $\mathcal{A}$*. Each mapping* $ref : V' \to U$ *with* $V' \subseteq V$ *is called a* PARTIAL VALUATION OF $\mathfrak{G}$*. If* $V' = V$*, then* ref *is called* (TOTAL) VALUATION OF $\mathfrak{G}$*. Let* $c \in Cut \cup \{\top\}$*. If* $V' \supseteq \{v \in V \mid v > c\}$ *and* $V' \cap \{v \in V \mid v \leq c\} = \emptyset$*, then* ref *is called* PARTIAL VALUATION FOR c*. If* $V' \supseteq \{v \in V \mid v \geq c\}$ *and* $V' \cap \{v \in V \mid v < c\} = \emptyset$*, then* ref *is called* EXTENDED PARTIAL VALUATION FOR c*.*

Now we can define whether an EGI is valid in a relational structure over $\mathcal{A}$. This shall be done in two ways. The first way is directly adopted from $\mathcal{FO}$ (see Def. 18.5). In $\mathcal{FO}$, we start with total valuations of the variables of the formula, i.e., an object is assigned to each variable. Whenever an $\exists$-quantifier is evaluated during the evaluation of the formula, the object which is assigned to the quantified variable is substituted (i.e., the total valuation is successively changed). As we adopt this classical approach of $\mathcal{FO}$ for EGIs, we denote the semantical entailment relation by '$\models_{class}$'. It is defined as follows:

Definition 13.3 (Classical Evaluation of Graphs). *Let an EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given and let* (U, I) *be a relational structure over* $\mathcal{A}$*. Inductively over the tree* $Cut \cup \{\top\}$*, we define*[1] $(U, I) \models_{class} \mathfrak{G}[c, ref]$ *for each context* $c \in Cut \cup \{\top\}$ *and every total valuation* $ref : V \to U$*:*

$(U, I) \models_{class} \mathfrak{G}[c, ref] :\Longleftrightarrow$

it exists a valuation $\overline{ref} : V \to U$ *with* $\overline{ref}(v) = ref(v)$ *for all* $v \in V \backslash area(c)$ *such that the following conditions hold:*

- $\overline{ref}(e) \in I(\kappa(e))$ *for each* $e \in E \cap area(c)$ *(edge condition))*
- $(U, I) \not\models_{class} \mathfrak{G}[d, \overline{ref}]$ *for each* $d \in Cut \cap area(c)$ *(cut condition and iteration over* $Cut \cup \{\top\}$*))*

If there is a total valuation ref *such that* $(U, I) \models_{class} \mathfrak{G}[\top, ref]$*, we write* $(U, I) \models_{class} \mathfrak{G}$*. If* $\mathfrak{H}$ *is a set of EGIs and if* $\mathfrak{G}$ *is an EGI such that* $(U, I) \models_{class} \mathfrak{G}$ *for each model* (U, I) *that satisfies* $(U, I) \models_{class} \mathfrak{G}'$ *for each* $\mathfrak{G}' \in \mathfrak{H}$*, we write* $\mathfrak{H} \models_{class} \mathfrak{G}$*.*

The second semantics we will provide is the formalization of Peirce's endoporeutic method to evaluate EGIs. He read and evaluated existential graphs

[1] In order to be in line with the notation '$\mathcal{M} \models_{val} f$' of $\mathcal{FO}$ (see Def.18.5), it would be preferable to use a notation '$(U, I) \models_{c,ref} \mathfrak{G}$' for EGIs. But as we have *two different* entailment relations in $\mathcal{EGI}$, we decided to distinguish them writing '$\models_{class}$' and '$\models_{endo}$' (see Def. 13.4). Thus, as the symbol '$\models$' is already indexed, we write the evaluated context c and the valuation ref in square brackets instead.

from the outside, hence starting with the sheet of assertion, and proceeded inwardly. During this evaluation, he assigned successively values to the lines of identity. The corresponding semantical entailment relation is denoted by '$\models_{endo}$'. But before we give a precise definition of '$\models_{endo}$', we exemplify it with the left graph of Fig. 12.4.

We start the evaluation of the graph on the sheet of assertion $\top$. As $\top$ contains v_1, e_1, c_1, $\mathfrak{G}$ is true if there is an object o_1 which is a cat, and the part of $\mathfrak{G}$ which is enclosed by c_1 is not true. As c_1 contains v_2, e_2, c_2, the part of $\mathfrak{G}$ enclosed by c_1 is true if there is an object o_2 such that o_1 is on o_2 (that is why the endoporeutic method proceeds *inwardly*: We cannot evaluate the inner cut c_1 unless we know which object is assigned to v_1. Please note that the assignment we have build so far is a partial valuation for the cut c_1.) and the graph enclosed by c_2 is not true, i.e., o_2 is not a mat. So $\mathfrak{G}$ is true if there is an object o_1 which is a cat, and it is not true that there is an object o_2 such that o_1 is on o_2 and o_2 is not a mat, i.e., there is a cat which is only on mats.

Definition 13.4 (Endoporeutic Evaluation of Graphs). *Let an EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be given and let (U, I) be a relational structure over $\mathcal{A}$. Inductively over the tree $Cut \cup \{\top\}$, we define $(U, I) \models_{endo} \mathfrak{G}[c, ref]$ for each context $c \in Cut \cup \{\top\}$ and every partial valuation $ref : V' \subseteq V \to U$ for c:*

$(U, I) \models_{endo} \mathfrak{G}[c, ref] :\Longleftrightarrow$

ref can be extended to an partial valuation $\overline{ref} : V' \cup (V \cap area(c)) \to U$ (i.e., $\overline{ref}$ is an extended partial valuation for c with $\overline{ref}(v) = ref(v)$ for all $v \in V'$), such that the following conditions hold:

- *$\overline{ref}(e) \in I(\kappa(e))$ for each $e \in E \cap area(c)$ (edge condition))*
- *$(U, I) \not\models_{endo} \mathfrak{G}[d, \overline{ref}]$ for each $d \in Cut \cap area(c)$ (cut condition and iteration over $Cut \cup \{\top\}$))*

For $(U, I) \models_{endo} \mathfrak{G}[\top, \emptyset]$ we write $(U, I) \models_{endo} \mathfrak{G}$. If $\mathfrak{H}$ is a set of EGIs and if $\mathfrak{G}$ is an EGI such that $(U, I) \models_{endo} \mathfrak{G}$ for each model (U, I) that satisfies $(U, I) \models_{endo} \mathfrak{G}'$ for each $\mathfrak{G}' \in \mathfrak{H}$, we write $\mathfrak{H} \models_{endo} \mathfrak{G}$.

Please note that the edge-condition for an edge e can only be evaluated when we have already assigned objects to all vertices being incident with e. This is assured because we only consider graphs with dominating nodes.

The main difference of the last definition to Def. 13.3 is the following: In Def. 13.3, we start with a total valuation of vertices which is, during the evaluation, successively changed. In the Def. 13.4, we start with the empty partial valuation which is, during the evaluation, successively completed. Unsurprisingly, these two definitions yield exactly the same entailment relations, as the following lemma shows.

Lemma 13.5 (Both Evaluations are Equivalent). *Let an EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given and let* $\mathcal{M} := (U, I)$ *be a relational structure over* $\mathcal{A}$. *Then we have*

$$\mathcal{M} \models_{class} \mathfrak{G} \quad \Longleftrightarrow \quad \mathcal{M} \models_{endo} \mathfrak{G}$$

Proof: We show inductively over $Cut \cup \{\top\}$ that the following conditions are satisfied for every context $c \in Cut \cup \{\top\}$ and every partial valuation $ref' : V' \subseteq V \to U$ for the context c:[2]

$$(U, I) \models_{endo} \mathfrak{G}[c, ref'] \tag{13.1}$$
$$\Longleftrightarrow (U, I) \models_{class} \mathfrak{G}[c, ref] \text{ for all extensions } ref : V \to U \text{ of } ref' \tag{13.2}$$
$$\Longleftrightarrow (U, I) \models_{class} \mathfrak{G}[c, ref] \text{ for one extension } ref : V \to U \text{ of } ref' \tag{13.3}$$

As $\mathfrak{G}$ has dominating nodes, we have

$$V_e \subseteq \{v \in V \mid v \geq c\} \text{ for each } e \in E \cap area(c) \tag{13.4}$$

Now the proof is done by induction over $Cut \cup \{\top\}$. Let c be a context such that (13.1) $\Longleftrightarrow$ (13.2) $\Longleftrightarrow$ (13.3) for each cut $d < c$.

We start with the proof of (13.1) $\Longrightarrow$ (13.2), so let $ref' : V' \subseteq V \to U$ be a partial valuation for c such that $(U, I) \models_{endo} \mathfrak{G}[c, ref]$. Hence there is a partial valuation $\overline{ref'}$ which extends ref' to $V' \cup (V \cap area(c))$ and which fulfills the properties in Def. 13.4. Furthermore, let $ref : V \to U$ be an arbitrary total valuation which extends ref' to V. We want to show $(U, I) \models_{class} \mathfrak{G}[c, ref]$. We set

$$\overline{ref} := ref\big|_{V \backslash area(c)} \cup \overline{ref'}\big|_{area(c)}$$

As ref is an extension of ref', we have $\overline{ref}\big|_{\{v \in V | v > c\}} = \overline{ref'}\big|_{\{v \in V | v > c\}}$, and by definition of $\overline{ref}$, we have $\overline{ref}\big|_{area(c)} = \overline{ref'}\big|_{area(c)}$. From this we conclude $\overline{ref}\big|_{\{v \in V | v \geq c\}} = \overline{ref'}\big|_{\{v \in V | v \geq c\}}$. Since all edge conditions hold for $\overline{ref'}$, (13.4) yields that all edge conditions hold for $\overline{ref}$, too. Furthermore we have $(U, I) \not\models_{endo} \mathfrak{G}[d, ref']$ for each $d < c$. Since $\overline{ref}$ is an extension of $\overline{ref'}$, the induction hypothesis (13.3) $\Longrightarrow$ (13.1) for d yields $(U, I) \not\models_{class} \mathfrak{G}[d, ref']$. So the proof for (13.1) $\Longrightarrow$ (13.2) is done.

The implication (13.2) $\Longrightarrow$ (13.3) holds trivially.

Finally, we show that (13.3) $\Longrightarrow$ (13.1) holds for c. So assume that (13.3) is true for c, i.e., there is an extension $ref : V \to U$ of ref' which satisfies $(U, I) \models_{class} \mathfrak{G}[c, ref]$. So there is a valuation $\overline{ref}$ with $\overline{ref}(v) = ref(v)$ for all $v \in V \backslash area(c)$ and which fulfills the properties in Def. 13.3. Now define

[2] The ongoing proof of Lem. 13.5 relies on the idea that a valuation does not determine which values are assigned to the vertices $v \in V \cap area(c)$. For this reason it is crucial to consider only partial valuations *for the context* c, i.e., partial valuations ref' which satisfy in particular $dom(ref') \cap \{v \in V \mid v \leq c\} = \emptyset$.

$$\overline{ref'} := ref' \cup \overline{ref}\big|_{area(c)}$$

Again, $\overline{ref}$ is an extension of $\overline{ref'}$, and $\overline{ref}\big|_{\{v \in V \mid v \geq c\}} = \overline{ref'}\big|_{\{v \in V \mid v \geq c\}}$. Similar arguments as in the proof for (13.1) $\Longrightarrow$ (13.2) yield that all edge conditions hold for $\overline{ref'}$. Furthermore, if $d < c$ is a cut, the cut-condition $(U, I) \not\models_{endo} \mathfrak{G}[d, \overline{ref'}]$ holds because of the induction hypothesis (13.1) $\Longrightarrow$ (13.2). So we have that $\overline{ref'}$ fulfills the properties in Def. 13.4, and the proof for the implication (13.3) $\Longrightarrow$ (13.1) is done. □

As Lem. 13.5 shows that Def. 13.3 and and Def. 13.4 yield the same entailment relation between models and graphs, we will write $\models$ instead of $\models_{endo}$ and $\models_{class}$. Nevertheless it will turn out that in some proofs it is more useful to use Def. 13.3 for $\models$, and in other proofs it is more useful to use Def. 13.4.

13.2 Semantics for Existential Graphs

In the next section, a calculus for EGIs will be presented. This calculus will have derivation rules, but recall that we already have four transformation rules on EGIs, from which the relation $\sim$ is obtained. In this section, we will show that these transformation rules are respected by the semantical entailment relation, i.e., that they are sound (that is, $\sim$ can be considered to be a 'congruence-relation' with respect to the semantics). Unsurprisingly, the proof-method is the same as the proof of the soundness of the forthcoming calculus in Chpt. 17.

We start with a simple definition.

Definition 13.6 (Partial Isomorphism Applied to Valuations). *Let EGIs* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ *be given and let* f *be an isomorphism between* $\mathfrak{G}$ *and* $\mathfrak{G}'$ *except for* $c \in Cut$ *and* $c' \in Cut'$. *Let* ref *be a partial valuation on* $\mathfrak{G}$ *with* $dom(ref) \cap \{v \in V \mid ctx(v) \leq c\} = \emptyset$. *Then we define* $f(ref)$ *on* $\{f(v) \mid v \in V \cap dom(ref)\}$ *by* $f(ref)(f(v)) := ref(v)$.

Now we can start with the proof of the soundness. Like in Alpha, we have a main theorem which is the basis for the soundness of nearly all rules, which will be presented in two forms.

Theorem 13.7 (Main Lemma for Soundness, Implication Version). *Let EGIs* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ *be given and let* f *be an isomorphism between* $\mathfrak{G}$ *and* $\mathfrak{G}'$ *except for* $c \in Cut$ *and* $c' \in Cut'$. *Set* $Cut_c := \{d \in Cut \cup \{\top\} \mid d \not< c\}$. *Let* $\mathcal{M}$ *be a relational model and let* $P(d)$ *be the following property for contexts* $d \in Cut_c$:

- *If d is a positive context and ref is a partial valuation for d which fulfills $\mathcal{M} \models_{endo} \mathfrak{G}[d, ref]$, then $\mathcal{M} \models_{endo} \mathfrak{G}'[f(d), f(ref)]$, and*
- *if d is a negative context and ref is a partial valuation for d which fulfills $\mathcal{M} \not\models_{endo} \mathfrak{G}[d, ref]$, then $\mathcal{M} \not\models_{endo} \mathfrak{G}'[f(d), f(ref)]$.*

If P holds for c, then P holds for each $d \in Cut_c$.

Particularly, If P holds for c, we have

$$\mathcal{M} \models_{endo} \mathfrak{G} \Longrightarrow \mathcal{M} \models_{endo} \mathfrak{G}' \quad .$$

Proof: Cut_c is a tree such that for each $d \in Cut_c$ with $d \neq c$, then all cuts $e \in area(d)$ are elements of Cut_c as well. Thus we can carry out the proof by induction over Cut_c. So let $d \in Cut_c$, $d \neq c$ (we know that c satisfies P) be a context such that $P(e)$ holds for all cuts $e \in area(d) \cap Cut$. Furthermore let ref be a partial valuation for d. We have two cases to consider:

- **First Case:** d is positive and $\mathcal{M} \models_{endo} \mathfrak{G}[d, ref]$.

  As we have $\mathcal{M} \models_{endo} \mathfrak{G}[d, ref]$, we can extend ref to a mapping $\overline{ref}$: $dom(ref) \cup (V \cap area(d)) \to U$ such that all edge-conditions in d hold and such that $\mathcal{M} \not\models_{endo} \mathfrak{G}[e, \overline{ref}]$ for all cuts $e \in area(d) \cap Cut$. Like in Def. 13.6, $f(ref)$ can be canonically extended to $\overline{f(ref)}$ such that we have $\overline{f(ref)} = f(\overline{ref})$. The isomorphism yields that all edge conditions hold for $\overline{f(ref)}$, and the induction hypothesis yields that $\mathcal{M} \not\models_{endo} \mathfrak{G}'[f(e), \overline{f(ref)}]$ for all cuts $e \in area(d) \cup Cut$. Hence we have $\mathcal{M} \models_{endo} \mathfrak{G}'[f(d), f(ref)]$.
- **Second Case:** d is negative and $\mathcal{M} \not\models_{endo} \mathfrak{G}[d, ref]$.

  Assume that we have $\mathcal{M} \models_{endo} \mathfrak{G}'[f(d), f(ref)]$, i.e., $f(ref)$ can be extended to $\overline{f(ref)}$ such that all edge conditions hold in $f(d)$ and such that $\mathcal{M} \not\models_{endo} \mathfrak{G}'[e', \overline{f(ref)}]$ for all cuts $e' \in area(f(d)) \cap Cut'$. Obviously there exists an extension $\overline{ref}$ of ref such that $f(\overline{ref}) = \overline{f(ref)}$. We conclude that all edge conditions hold for $\overline{ref}$ in d. Our assumption yields that we have $\mathcal{M} \not\models_{endo} \mathfrak{G}'[f(e), \overline{f(ref)}]$, i.e., $\mathcal{M} \not\models_{endo} \mathfrak{G}'[f(e), f(\overline{ref})]$ for all cuts $e \in area(d)$. By induction hypothesis we have $\mathcal{M} \not\models_{endo} \mathfrak{G}[e, \overline{ref}]$ for all cuts $e \in area(d)$. So $\overline{ref}$ is an extension of ref which fulfills all properties of Def. 13.4, hence we have $\mathcal{M} \models_{endo} \mathfrak{G}[d, ref]$, a contradiction.

From both cases we conclude that P holds for each $d \in Cut_c$. Finally we have

$$\begin{aligned} \mathcal{M} \models_{endo} \mathfrak{G}_a &\overset{\text{Def.} \models}{\Longleftrightarrow} \mathcal{M} \models_{endo} \mathfrak{G}_a[\top_a, \emptyset] \\ &\overset{\text{P}(\top_\text{a})}{\Longrightarrow} \mathcal{M} \models_{endo} \mathfrak{G}_b[\top_b, \emptyset] \\ &\overset{\text{Def.} \models}{\Longleftrightarrow} \mathcal{M} \models_{endo} \mathfrak{G}_b \end{aligned}$$

which yields the particular property for $\top$. $\Box$

This will be the main theorem to prove the soundness of those rules of the (forthcoming) calculus which weaken the 'informational content' of an EGI (e.g. erasure and insertion). The transformation rules of Def. 12.14 and some rules of the calculus (e.g. iteration and deiteration) do not change the informational content and may therefore be performed in both directions. For proving the soundness of those rules, a second version of this theorem is appropriate which does not distinguish between positive and negative contexts and which uses an equivalence instead of two implications for the property P.

Theorem 13.8 (Main Thm. for Soundness, Equivalence Version). *Let EGIs $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be given and let f be an isomorphism between $\mathfrak{G}$ and $\mathfrak{G}'$ except for $c \in Cut$ and $c' \in Cut'$. Set $Cut_c := \{d \in Cut \cup \{\top\} \mid d \not< c\}$. Let $\mathcal{M}$ be a relational model and let $P(d)$ be the following property for contexts $d \in Cut_c$:*

Every partial valuation ref for d satisfies

$$\mathcal{M} \models_{endo} \mathfrak{G}[d, ref] \quad \Longleftrightarrow \quad \mathcal{M} \models_{endo} \mathfrak{G}'[f(d), f(ref)]$$

If P holds for c, then P holds for each $d \in Cut_c$.

Particularly, If P holds for c, we have

$$\mathcal{M} \models_{endo} \mathfrak{G} \Longleftrightarrow \mathcal{M} \models_{endo} \mathfrak{G}' \quad .$$

Proof: Done analogously to the proof of Lemma 13.7. □

In the following, we will show that two EGIs $\mathfrak{G}, \mathfrak{G}'$ with $\mathfrak{G} \sim \mathfrak{G}'$ have in every model $\mathcal{M} := (U, I)$ the same meaning. For the isomorphism-rule, this is obvious. Next, let $\mathfrak{G}'$ is obtained from $\mathfrak{G}'$ with the rule 'changing the orientation of an identity-edge. Due to Defs. 13.1 and 13.4, the relation-name $\doteq$ is interpreted by the –symmetric– identity-relation on u. Thus it is trivial that $\mathfrak{G}$ holds in $\mathcal{M}$ if and only if $\mathfrak{G}'$ holds in $\mathcal{M}$. It remains to investigate the rules 'adding a vertex to a ligature' and 'removing a vertex from a ligature'. This will be done in the next lemma.

Lemma 13.9 (The Transformation Rules for Ligatures are Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are EGIs, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by applying one of the transformation rules ' adding a vertex to a ligature' and 'removing a vertex from a ligature', then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: We use the notation of Def. 12.14, i.e., we have a vertex $v \in V$, an edge $e \in E$ be an edge with $e\big|_i = w$ for an $i \in \mathbb{N}$, and a context c with $ctx(v) \geq c \geq ctx(e)$ such that in c, a new vertex v' and a new identity-edge between v and v' is inserted, and on e, $e\big|_i = v$ is substituted by $e\big|_i = v'$.

$\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except the cut c, where the isomorphism is the identity-mapping for $\{d \in Cut \cup \{\top\} \mid d \not< c\}$. Let ref be a partial valuation for the cut c.

First we suppose $\mathcal{M} \models_{endo} \mathfrak{G}[c, ref]$. That is, ref can be extended to $\overline{ref}$: $V' \cup (V \cap area(c)) \to U$ such that the all edge- and cut-conditions in c hold (for $\mathfrak{G}$). In $\mathfrak{G}'$, we have $area'(c) = area(c) \dot\cup \{v', e'\}$, i.e., we have –compared with $\mathfrak{G}$– the same edge- and cut-conditions in c plus an additional edge-condition for the new edge e'. Thus, for $\overline{ref}\,' := \overline{ref} \dot\cup \{(v', ref(v)\}$, we easily see that $\overline{ref}\,'$ is in $\mathfrak{G}'$ a partial valuation for the cut c which extends ref and which satisfies all edge- and cut-conditions in c (for $\mathfrak{G}'$). So we get $\mathcal{M} \models_{endo} \mathfrak{G}'[c, ref]$.

Now let us suppose $\mathcal{M} \models_{endo} \mathfrak{G}'[c, ref]$, i.e., , ref can be extended to $\overline{ref}\,'$: $V' \cup (V \cap area'(c)) \to U$ such that the all edge- and cut-conditions in c hold (for $\mathfrak{G}'$). Now we easily see that $\overline{ref} := \overline{ref}\,' \backslash \{v, \overline{ref}\,'(v')\}$ is in $\mathfrak{G}'$ a partial valuation for the cut c which extends ref and which satisfies all edge- and cut-conditions in c. So we get $\mathcal{M} \models_{endo} \mathfrak{G}[c, ref]$.

Both cases together yield the property $P(c)$ of Lemma 13.7, so we conclude $\mathcal{M} \models_{endo} \mathfrak{G} \Longleftrightarrow \mathcal{M} \models_{endo} \mathfrak{G}'$. □

As we now know that all transformation rules respect the relation $\models$, we immediately obtain the following theorem:

Theorem 13.10 ($\sim$ Preserves Evaluations). *Let $\mathfrak{G}_1, \mathfrak{G}_2$ be two EGIs with $\mathfrak{G}_1 \sim \mathfrak{G}_2$, and let $\mathcal{M}$ be a relational structure. Then we have:*

$$\mathcal{M} \models \mathfrak{G}_1 \quad \Longleftrightarrow \quad \mathcal{M} \models \mathfrak{G}_2 \quad .$$

Now we can carry over the semantics for EGIs to semantics for formal EGs:

Definition 13.11 (Semantics for Existential Graphs). *Let $\mathfrak{E}$ be an EG, $\mathfrak{G}$ be an representing EGI for $\mathfrak{E}$ (i.e., $\mathfrak{E} = [\mathfrak{G}]_\sim$) and $\mathcal{M}$ be a relational structure. We set:*

$$\mathcal{M} \models \mathfrak{E} \quad :\Longleftrightarrow \quad \mathcal{M} \models \mathfrak{G} \quad .$$

The last theorem and definition can be understood as 'a posteriori' justification for Peirce's understanding of LoIs as composed of an arbitrary number of identity spots, as well as for its formalization in form of EGIs and classes of EGIs.

14

Getting Closer to the Calculus for Beta

In Chpt. 11, we first have extensively investigated Peirce's understanding of the form and meaning of existential graphs. Afterwards, the mathematical definitions for syntax and semantics of formal existential graphs, which can be seen as a formal reconstruction of Peirce's view on his graphs, were provided. The same attempt is now carried out for the calculus. In this chapter, which corresponds to Sec. 8.2 in the Alpha-part, we will deeply discuss Peirce's rules for existential graphs, before the formal definition for the calculus is provided in Chpt. 15.

The set of rules for the beta-part is essentially the same as for alpha graphs, i.e., we have the rules erasure, insertion, iteration, deiteration and double cut. These rules are now extended to handle LoIs as well. The additionally handling of LoIs has to discussed in detail, particularly for the rules iteration and deiteration. Furthermore, it will turn out that a further rule for the insertion and erasure of single, heavy dots has to be added.

In different writings on existential graphs, Peirce provided different versions of his rules, which differ in details. Thus the elaboration of the rules which will provided in this chapter should be understood as *one possible* formulation which tries to cover Peirce's general understanding of his rules, which can be figured out from the different places where Peirce provides and discusses them. Roughly speaking, the following rules will be employed:

1. **Rule of Erasure and Insertion**

   In positive contexts, any subgraph may be erased, and in negative contexts, any subgraph may be inserted.

2. **Rule of Iteration and Deiteration**

   If a subgraph of a graph is given, a copy of this subgraph may be inserted into the same or a deeper nested context. On the other hand, if we already have a copy of the subgraph in the same or a deeper nested context, then this copy may be erased.

3. **Double Cut Rule**

   Two Cuts one within another, with nothing between them, except ligatures which pass entirely through both cuts, may be inserted into or erased from any context.

4. **Rule of Inserting and Deleting a Heavy Dot**

   A single, heavy dot may be inserted to or erased from any context.

The double cut rule can be understood as a pair of rules: Erasing a double cut and inserting a double cut. So we see that the rules of Peirce are grouped in pairs. Moreover, we have two kinds of rules.[1] The rules erasure and insertion (possibly) weaken the informational content of a graph. These rules will be called GENERALIZATION RULES. All other rules do not change the informational content of a graph. These rules will be called EQUIVALENCE RULES.

Each equivalence rule can be carried out in arbitrary contexts. Moreover, for each equivalence rule which allows to derive a graph $\mathfrak{G}_2$ from a graph $\mathfrak{G}_1$, we have a counterpart, a corresponding rule which allows to derive $\mathfrak{G}_1$ from $\mathfrak{G}_2$. On the other hand, generalization rules cannot be carried out in arbitrary contexts. Given a pair of generalization rules, one of the rules may only be carried out in *positive* contexts, and the other rule is exactly the opposite direction of the first rule, which may only be carried out *negative* contexts

This duality principle is essential for Peirce's rules.[2] In the following sections, we discuss each of the pairs of rules for Peirce's EGs in detail, before we elaborate their mathematical implementation for EGIs and formal EGs in the next chapter. Due to the duality principle, it is sufficient to discuss only one rule of a given pair in detail. For example, in the next section, where the pair erasure and insertion is considered, only the erasure-rule is discussed, but the results of the discussion can easily be transferred to the insertion-rule, too.

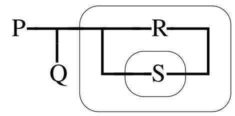
14.1 Erasure and Insertion

We have already discussed in Sec. 8.2 how subgraphs are erased and inserted in Alpha. For Beta, we have to discuss how these rules are extended for the handling of LoIs and ligatures. In 4.503, Peirce provides considers 20 graphs and explains how LoI are treated with them, and in 4.505, he summarizes: '*This rule permits any ligature, where evenly enclosed, to be severed, and any two ligatures, oddly enclosed in the same seps, to be joined.*'

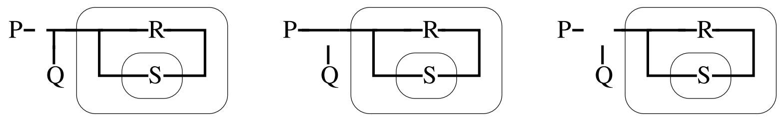
The rule of erasure shall here be exemplified with the following graph:

[1] This has already be mentioned in Sec.8.2.

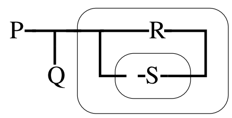
[2] This principle will be adopted in the part 'Extending the System', where the expressiveness of EGIs is extended and new rules are added to the calculus.

$\mathfrak{G} :=$ 

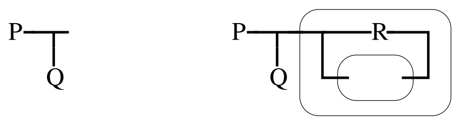
The following three examples are obtained from $\mathfrak{G}$ by erasing a part of the ligature on the sheet of assertion.



Analogously, we can erase parts of the ligature in the innermost cut. For example, we can derive

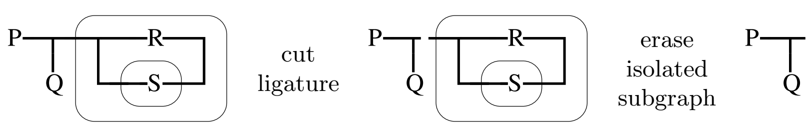


Similar to Alpha, whole subgraphs may be erased from positive contexts. This is how the following two graphs are obtained:

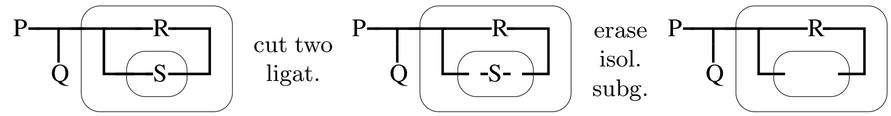


In the first graph, we have erased the whole subgraph of $\mathfrak{G}$ which consists of the outermost cut and all what it is scribed inside. In the second example, the subgraph —S— in the innermost cut has been erased.

In these examples, we have erased a subgraph which was connected to the remaining graph. In order to do this, we had to cut these connections. This is explained by Peirce in 4.505, where he explains: '*In the erasure of a graph by this rule, all its ligatures must be cut.*' For the ongoing mathematization of the rules, it is important to note that an erasure of a subgraph which is connected to the remaining graph can be performed in two steps: First cut all ligations which connect the subgraph with the remaining graphs (this has to be done in the positive context where the subgraph is placed) in order to obtain an isolated subgraph, then erase this isolated subgraph. For example, the deletion of the subgraph in the first example can be performed as follows:



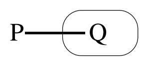
Analogously, the deletion of the subgraph in the second example can be performed as follows:



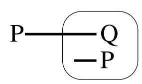
It will turn out that this consideration will be helpful for a mathematization of these rules for EGIs.

14.2 Iteration and Deiteration

We start with a very simple example of an application of the iteration-rule. Consider the following graph:



The next graph is derived from this left graph by iterating the subgraph P— into the cut.



This is a trivial application of the iteration-rule. But, of course, a crucial point in the iteration-rule for beta is the handling of ligatures. In fact, the iterated subgraph may be joined with the existing heavily drawn line. The question is: how?

In 4.506, Peirce writes that the iteration rule '*includes the right to draw a new branch to each ligature of the original replica inwards to the new replica.*' Consider the following two graphs, which have the same meaning:

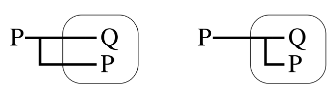
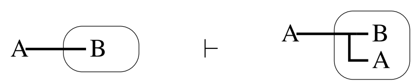


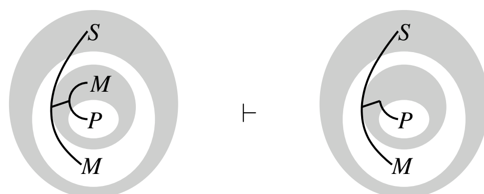
Fig. 14.1. two different existential graphs with the same meaning

As we have in the left graph of Fig. 14.1 a branch from the old replica of P— which goes inwardly to the copy of the iterated subgraph, one might think that this graph is the result from a correct application of the iteration-rule to our starting graph, while the right is not. But, then, it must be possible to show that the two graphs of Fig. 14.1 can be syntactically, i.e., with the rules of the calculus, transformed into each other. With the just given, first interpretation of the iteration rule, this is probably impossible. In fact, it turns out that the quotation of Peirce for the handling of LoIs might be misleading, and the right graph of Fig. 14.1 appears to be the result of an application of the iteration-rule as well.

In 4.386 Peirce provides an example how the alpha-rule of iteration is amended to beta. He writes: '*Thus,* $[A—(—B)]$ *can be transformed to* $[A\ \ (A—B)]$.' Peirce uses in this place a notation with brackets. It is crucial to note that the line of identity in the copy of A— is connected *inside* the cut with the already existing ligature. Thus, using cuts, the example can be depicted as follows:



A similar example can be found in [PS00], where Peirce uses the rule of deiteration to remove a copy of M— (like in one example we referred to in Chpt. 11, he uses shadings for representing cuts):

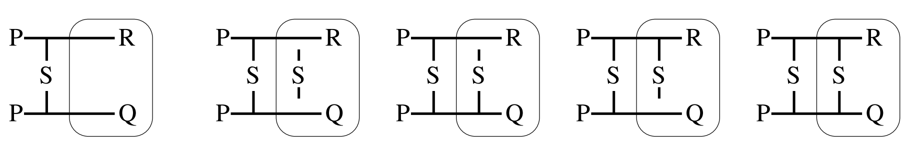


Again it is crucial to note that the copy of M— was connected in the *second innermost* cut with the ligature. For Gamma, Peirce provides 1906 in 4.566 another example with the same handling of ligatures when a subgraph is iterated.

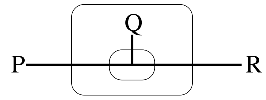
From these examples, I conclude the following understanding of handling ligatures in the iteration-rule:

> **Handling ligatures in the rule of iteration:** Assume that a subgraph S is iterated from a cut (or the sheet of assertion) c into a cut (or the sheet of assertion) d. Furthermore, assume that S is connected to a ligature which goes inwardly from c to d. Then the copy of S may be connected in d with this ligature.

In order to provide a (slightly) more sophisticated example for the handling of ligatures, consider the following graphs. Each of the four graphs on the right are resulting from a correct application for iterating the subgraph S of the leftmost graph into its cut:



The phrase "which goes inwardly from c to d" is crucial for the correct application of the iteration-rule. Consider the following graph:

$\mathfrak{G} :=$ 

The four graphs of Fig. 14.2 are results of a correct application of the iteration-rule to $\mathfrak{G}$. In fact, one can check that all graphs have the same meaning, which is the meaning of $\mathfrak{G}$.

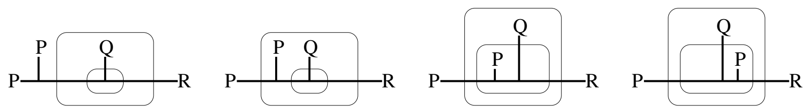


Fig. 14.2. Five results from a correct application of the iteration-rule to $\mathfrak{G}$

In the next two examples, the iterated subgraph is connected to a ligature which does not "go inwardly from c to d" (the ligature crosses some cuts more than once), so these examples are results of an *in-correct* application of the iteration-rule to $\mathfrak{G}$.

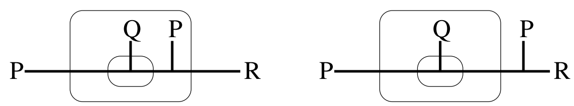


Fig. 14.3. two different results from a incorrect application of the iteration-rule

These graphs have *different* meanings than $\mathfrak{G}$. This can be seen if we consider the following models:

$\mathcal{M}_1 :=$

	P	R	Q
a	$\times$		$\times$
b		$\times$	

$\mathcal{M}_2 :=$

	P	R	Q
a	$\times$		
b		$\times$	

It can easily be seen that $\mathfrak{G}$ does not hold in $\mathcal{M}_1$, but the left graph of Fig. 14.3 holds in $\mathcal{M}_1$, thus they have different meanings. Analogously, $\mathfrak{G}$ holds in $\mathcal{M}_2$, but the right graph of Fig. 14.3 does not, thus these two graphs have different meanings as well.

The rule of iteration may be reversed by the rule of deiteration. Particularly, if we derive a graph from another graph by the rule of iteration, both graphs have the same meaning. Thus we now see that the graphs of Fig. 14.3 cannot be allowed to be derived by the rule of iteration from $\mathfrak{G}$.

With our understanding, the right graph of of Fig. 14.1 is the result of a correct application of the iteration-rule. We have now in turn to show how the left graph of Fig. 14.1 can be derived from $\mathfrak{G}$.

In 4.506, Peirce continues his explanation of the iteration-rule as follows:

> The rule permits any loose end of a ligature to be extended inwardly through a sep or seps or to be retracted outwards through a sep or seps. It permits any cyclical part of a ligature to be cut at its innermost part, or a cycle to be formed by joining, by inward extensions, the two loose ends that are the innermost parts of a ligature.

With our handling of ligatures in the iteration-rule, it is not clear why loose ends of ligatures can be be extended inwardly through cuts. For this reason, this extension of ligatures will be a separate clause in the version of the iteration-rule which will be used and formalized in this treatise. In other words, the mathematical definition of the iteration/deiteration rule in the next chapter will be a formalization of the following informal understanding of the rules:

Iteration:

1. Let $\mathfrak{G}$ be a graph, and let $\mathfrak{G}_0$ be a subgraph of $\mathfrak{G}$ which is placed in a cut (or the sheet of assertion) c. Let d be a a cut (or the sheet of assertion) such that $d = c$ or d is deeper nested than c. Then a copy of $\mathfrak{G}_0$ can be scribed on the area of d.

   If $\mathfrak{G}_0$ is connected to a ligature which goes inwardly from c to d and which crosses no cut more than once, then the copy of $\mathfrak{G}_0$ may be connected in d with this ligature.

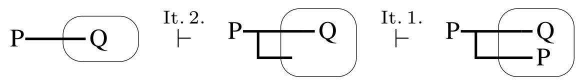
2. A branch with a loose end may be added to a ligature, and this loose end may be extended inwardly through cuts.[3]

Deiteration:

The rule of iteration may be reversed. That is: If $\mathfrak{G}'$ can be derived from $\mathfrak{G}$ with the rule of iteration, then $\mathfrak{G}$ can be derived from $\mathfrak{G}'$ with the rule of deiteration.

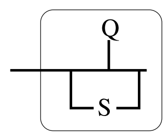
[3] In some places, extensions of ligatures, particularly the transformation carried out by part 2. of the iteration-rule is for Peirce a part of the insertion-rule, not of the iteration-rule. This understanding of this transformation is explained for example in 4.503, where Peirce needs to consider EGs where LoIs terminate on a cut. On the other hand, an extension of a ligature fits very well in the idea behind the iteration-rule, and in other places (e.g. 4.506), an extension of ligatures is encompassed by the iteration-rule. As we moreover dismissed EGs where LoIs terminate on a cut, it is convenient to subsume the extension of ligatures by the iteration-rule.

With this rule, clause 1., the right graph of Fig. 14.1 can be derived from $\mathfrak{G}$. Using both clauses one after the other, it is possible to derive the left graph of Fig. 14.1 from $\mathfrak{G}$ as well, as the following derivation shows:



Moreover, our understanding of the iteration-rule allows to rearrange ligatures in cuts in nearly arbitrary ways. In the following, this will be exemplified with some examples. In Chpt. 16, some lemmata are provided which capture mathematically the ideas behind these examples.

The main idea in all ongoing examples is to iterate or deiterate a portion of a ligature in a cut. If a portion of a ligature is iterated, the rule of iteration allows to connect arbitrary points of the iterated copy with arbitrary points of the ligature. For example, if we start with the graph



,

all the graphs in Fig. 14.4 can be derived by a single application of the iteration-rule:

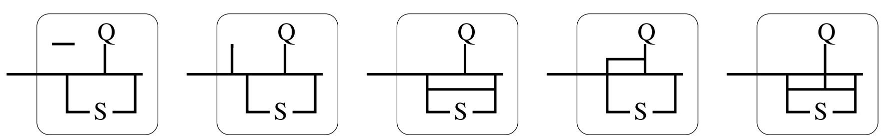
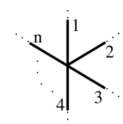


Fig. 14.4. Iterating a Ligature within a Context

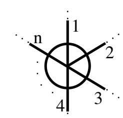
In the first example, a portion of the ligature is iterated without connecting the copy with the ligature. In the second example, a portion of the ligature is iterated, and one of its two ends is connected to the ligature (this could be an application of part 2. of the iteration-rule as well). In the third example, a portion of the ligature is iterated, and both ends are connected to the ligature. The same holds for the fourth example. Finally, in the last example, three points of the iterated copy are connected to the ligature (one of them to a branching point).

On page 102, I have already mentioned that Peirce did not consider EGs having branching points with more than three branches. An easy part of Peirce's reduction thesis is that we do not need to consider identity relations with an arity higher than three, that is, graphs having identity spots in which more than three LoIs terminate can be avoided without any loss in their expressivity. This is probably the reason we do not find in Peirce's works any EG where branching points with more than three branches occur. Nonetheless, in this treatise, due to convenience, branching points with more than three branches are allowed. But if a graph having spots in which more than three LoIs terminate is given, using the iteration/deiteration-rule, it can easily be transformed into a semantically equivalent graph in which no such spots occur. In order to see that, consider a branching point with more than three branches.

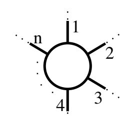
A ligature may contain branching points with more than three branches. A branching point with n branches can be sketched as follows (the small numbers are used to enumerate the branches):



With an n-fold application of the last lemma, we can add n lines of identity to this part of the ligature, which yields:

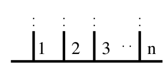


Now, again with the last lemma, we can remove all lines of identity within the circle. Then we obtain:



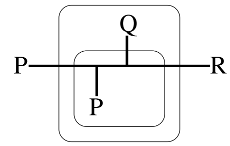
With this procedure, we can transform each identity spot in which more than three LoIs terminate into a 'wheel' on which only teridentity spots occur.

A very slight modification of this proof shows that each branching point with more than three branches can be converted into a 'fork-like' ligature as well, as it is depicted below (this is the kind of graphical device which Zeman normally uses in [Zem64] instead of branching points with n branches):

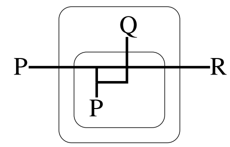


Next, I want to show that "branches of a ligature may be moved within a context along that ligature". In order to understand this phrase, consider the third, fourth and fifth graph of Fig. 14.2. These graphs differ only in the fact that the branch P— of the ligature in the innermost cut is attached at different positions to that ligature. These graphs can be easily transformed into each other. I will exemplify this with the third and forth graph of Fig. 14.2 (the shape and length of some lines of identity are slightly different to the graphs of Fig. 14.2, but "the shape and length [of some lines of identity] are matters of indifference" (4.416)).

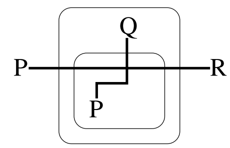
We start with the third graph of Fig. 14.2:



Now a line of identity which is part of the ligature is iterated into the same cut. Both ends of the copy of the line of identity are connected to the ligature as follows:

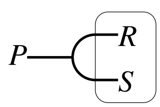


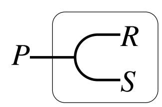
The portion of the ligature between the two branches above P could have been inserted by a similar application of the iteration-rule like in the last step. Thus we are allowed to remove it with the deiteration-rule. This yields the graph on the right, which is the fourth graph of Fig. 14.2:



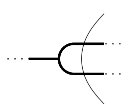
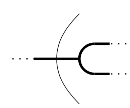
The proof uses only iteration and deiteration and maybe therefore be carried out in both directions.

In Chpt. 11 we have already seen that



and


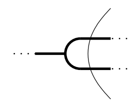
have the same meaning (see page 119). They differ only in the place of the branching point, which has so-to-speak moved from the sheet of assertion into the cut. In fact, moving a branching point inwardly can be done in arbitrary graphs for branching points in arbitrary context. That is, in Peirce's graph, we have:[4]

A devise like  can be replaced by  and vice versa

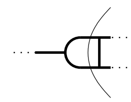
(in this representation, a segment of a cut-line is sketched, and we agree that the whole device is part of a graph, placed in an arbitrary context). With the rules iteration/deiteration and the possibility to move branches along ligatures, as is just has been discussed, we can now elaborate how these devices can syntactically be transformed into each other. This shall be done now.

[4] This result will be made precise and generalized with Def. 16.6 and Lemma 16.7. Burch obtained for his algebraic elaboration of Peirce's logic in ([Bur91a]) a corresponding result. It is Thm. 7.5. of [Bur91a], a theorem of which Burch claims to be 'of great importance for the presentation of the Representation theorem'.

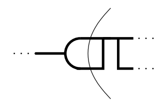
Let a graph be given where the left device occurs. I.e., we start with the device on the right, placed in an arbitrary context:



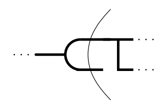
First we iterate a part of the ligature of the outer cut into the inner cut:



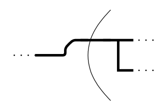
Now we move the branch in the inner cut:



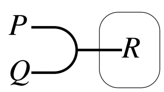
The iteration of the first step is reversed with the deiteration-rule:



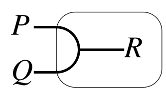
The 'loose' end of the ligature is retracted with the deiteration-rule. We obtain right device, which is the device we wanted to derive (drawn slightly different).



A more general and formally elaborated version of this transformation will be provided for EGIs in Chpt. 16. Finally, recall that we have already seen that the following graphs have different meanings (see page 119):



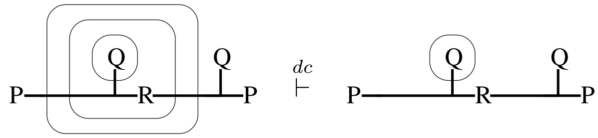
and



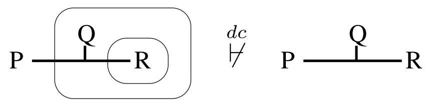
Thus it is *not* allowed to move the branching point in this example. Indeed, the above given proof cannot be applied to these graphs, as it relies on some applications of the iteration-rule to parts of the ligature in the outer cut.

14.3 Double Cuts

In Alpha, a double cut is a device of two nested cuts c_1, c_2 with $area(c_1) = \{c_2\}$). That is, the outer c_1 contains nothing directly except the inner cut c_2 (but, of course, it is allowed that c_2 in turn may contain other items, as these items are thus not directly contained by c_1). This understanding of a double cut has to be generalized in Beta. In an example for Beta in Chpt. 2, we have already seen an application of the double-cut rule where a ligature passes entirely through the double cut (see page 14). This handling of ligatures is described by Peirce's definition of the double cut rule for Beta in 4.567: '*Two Cuts one within another, with nothing between them, unless it be Ligatures passing from outside the outer Cut to inside the inner one, may be made or* *abolished on any Area.*' Let us first consider a valid example of the double-cut-rule. We have:

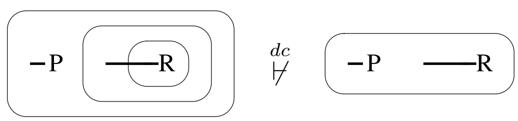


It is obvious that the area of the outer cut is not allowed to contain any relation-names. For example, although we have a ligature which passes through both cuts, the following is an invalid application of the double-cut rule (indicated by the crossing out of the symbol '$\vdash$'):



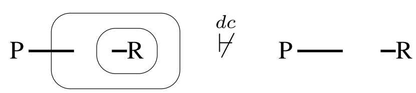
In fact, if we consider the relational structure $\mathcal{M}_1$ with a single element a, such that $P(a)$ and $R(a)$, but not $Q(a)$ holds, the left graph holds in $\mathcal{M}_1$, but the right graph does not.

Moreover, it is crucial that each ligature passes through both cuts: No ligature may start or end in the area of the outer cut. If a ligature starts in the area of the outer cut, we may obtain an invalid conclusion, as the following example shows:



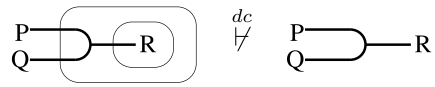
The left graph has the meaning 'if there is an object with property P, then there is an object which has not property R', while the meaning of the right graph is 'it is not true that there exists an object with the property P and an object with the property R', thus this conclusion is not semantically valid.

Strictly speaking, a ligature is not allowed to end in the area of the outer cut, neither, that is, the next example is again an invalid application of the double cut rule.



But in fact, the right graph can be derived from the left graph by first retracting the ligature with the deiteration-rule, and then by applying the double-cut rule. Both graphs are semantically equivalent.

'To pass through both cuts' has to be understood very rigidly: A ligature which passes through the cut, but which has branches in the area of the outer cut, may cause problems. The next example is another invalid application of the double-cut rule:



To see that this implication is not valid, consider the following relational structure $\mathcal{M}_2$:

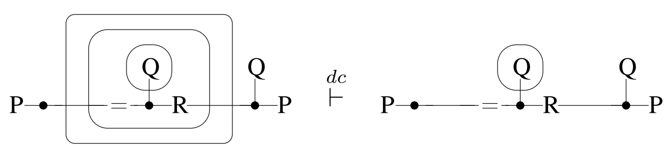
$\mathcal{M}_2 :=$

	P	Q	R
a	$\times$		
b		$\times$	

It is easy to check that the left graph holds in $\mathcal{M}_2$, but the right graph does not.

So, for Beta, two cuts c_1, c_2 are understood to be a double cut if c_2 is scribed in the area of c_1, and if c_1 contains except c_1 only ligatures which begin outside of c_1, pass completely through c_1 into c_2, and which do not have on the area of c_1 any branches.

It is a slightly surprising, but nice feature of the mathematical elaboration of existential graphs that we do not have to change our understanding of a double cut in EGIs if we go from Alpha to Beta. That is, for EGIs, which are the mathematical implementations of Peirce's Beta graphs, a double cut is indeed still a device of two cuts c_1, c_2 with $area(c_1) = \{c_2\}$. To see this, assume we consider a Peircean Beta graph with a double cut, such that there is a ligature which passes entirely through this double cut. In the mathematical reconstruction of this EG, i.e. in the corresponding EGI, we do not have to place a vertex in the area of the outer cut. This can be better seen if we 'translate' the example of the beginning of this section into EGIs. A application of the double cut rule of the corresponding EGIs is as follows (in order to indicate that neither any vertex, nor any identity-edge is placed in the area of the outer cut, the identity-edge is labeled with the relation-symbol '=' in the appropriate cut):



14.4 Inserting and Deleting a Heavy Dot

In our semantics, we consider, as usual in mathematics, only *non-empty* relational structures as models (see Def. 13.1). For this reason, it must be possible to derive that there is an object.

Surprisingly, neither in Peirce's manuscripts, nor in secondary literature, we find a rule which explicitely allows to derive a heavy dot or a line of identity.

But, in his 'Prolegomena to an Apology For Pragmaticism', he states in 4.567 '*that, since a Dot merely asserts that some individual object exists, and is thus one of the implications of the Blank, it may be inserted in any Area.*' This principle is not stated as an explicit rule, but as a principle '*the neglect of which might lead to difficulties.*' In order to provide a complete calculus, it is convenient to add this principle as a rule to it.

15

Calculus for Existential Graphs

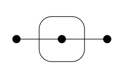
In this chapter, we will provide the formal definitions for the calculus for EGIs and EGs.

We start with the calculus for EGIs. Before its formal definition is given, we first have to introduce a simple relation on the set of vertices of an EGI. Recall that in the iteration-rule, where a subgraph is iterated from a context c into a context d, we had to consider 'ligatures which go inwardly from c to d'. This idea is formally captured as a relation Θ on the set of vertices.

Definition 15.1 (Θ). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. On V, a relation Θ is defined as follows: Let $v, w \in V$ be two vertices. We set $v\Theta w$ iff there exist vertices $v_1, \ldots, v_n$ ($n \in \mathbb{N}$) with*

1. *either $v = v_1$ and $v_n = w$, or $w = v_1$ and $v_n = v$,*
2. *$ctx(v_1) \geq ctx(v_2) \geq \ldots \geq ctx(v_n)$, and*
3. *for each $i = 1, \ldots, n-1$, there exists an identity edge $e_i = \{v_i, v_{i+1}\}$ between v_i and v_{i+1} with $ctx(e_i) = ctx(v_{i+1})$.*

It should be noted that Θ is trivially reflexive and symmetric, but usually not transitive (i.e., Θ is no equivalence relation). This can easily be seen with the following well-known graph:



The vertex in the cut is in Θ-relation with each of the two vertices on the sheet of assertion, but these two vertices are not in Θ-relation.

Now we are prepared to provide the definition for the calculus. Similar to Sect. 8.3, we will first describe the whole calculus using common spoken language. After this, we present mathematical definitions for the rules.

Definition 15.2 (Calculus for Existential Graphs Instances). *The calculus for EGIs over the alphabet $\mathcal{R}$ consists of the following rules:*

- **erasure**

  In positive contexts, any directly enclosed edge, isolated vertex, and closed subgraph may be erased.

- **insertion**

  In negative contexts, any directly enclosed edge, isolated vertex, and closed subgraph may be inserted.

- **iteration**

  – *Let* $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0)$ *be a (not necessarily closed) subgraph of* $\mathfrak{G}$ *and let* $c \leq ctx(\mathfrak{G}_0)$ *be a context such that* $c \notin Cut_0$*. Then a copy of* $\mathfrak{G}_0$ *may be inserted into* c*.*

    Furthermore, the following may be done: If $v \in V_0$ *with* $ctx(v) = ctx(\mathfrak{G}_0)$ *is a vertex, and if* $w \in V_0$ *with* $ctx(w) = c$ *is a vertex with* $v\Theta w$*, then an identity edge between* v *and* w *may be inserted into* c*.*

  – *If* $v \in V$ *is a vertex, and if* $c \leq ctx(v)$ *is a cut, then a new vertex* w *and an identity edge between* v *and* w *may be inserted into* c*.*

- **deiteration**

  If $\mathfrak{G}_0$ *is a subgraph of* $\mathfrak{G}$ *which could have been inserted by rule of iteration, then it may be erased.*

- **double cuts**

  Double cuts (two cuts c_1, c_2 *with* $area(c_1) = \{c_2\}$*) may be inserted or erased.*

- **erasing a vertex**

  An isolated vertex may be erased from arbitrary contexts.

- **inserting a vertex**

  An isolated vertex may be inserted in arbitrary contexts.

Next, the mathematical definition for the rules are provided.

- **double cuts**

  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI graph and $c_1, c_2 \in Cut$ such that $area(c_1) = \{c_2\}$. Let $c_0 := ctx(c_1)$ (i.e., $c_1 \in area(c_0)$) and set $\mathfrak{G}' := (V, E, \nu, \top, Cut', area', \kappa)$ with

  – $Cut' := Cut \backslash \{c_1, c_2\}$

  – $area'(d) := \begin{cases} area(d) \text{ for } d \neq c_0 \\ area(c_0) \cup area(c_2) \text{ for } d = c_0 \end{cases}$.

  Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING THE DOUBLE CUTS c_1, c_2 and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by INSERTING THE DOUBLE CUTS c_1, c_2.

- **erasure and insertion, erasing and inserting a vertex**

  First, general definitions for inserting and erasing vertices, edges and closed subgraphs are provided.

  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI with an edge $e \in E$, and let $\mathfrak{G}^{(e)} := (V^{(e)}, E^{(e)}, \nu^{(e)}, \top^{(e)}, Cut^{(e)}, area^{(e)}, \kappa^{(e)})$ be the following graph:

  – $V^{(e)} := V$
  – $E^{(e)} := E \backslash \{e\}$
  – $\nu^{(e)} := \nu\big|_{E^{(e)}}$
  – $\top^{(e)} := \top$
  – $Cut^{(e)} := Cut$
  – $area^{(e)}(d) := area(d) \backslash \{e\}$ for all $d \in Cut^{(e)} \cup \{\top^{(e)}\}$
  – $\kappa^{(e)} := \kappa\big|_{E^{(e)}}$

  Let $c := ctx(e)$. We say that $\mathfrak{G}^{(e)}$ is derived from $\mathfrak{G}$ by ERASING THE EDGE e FROM THE CONTEXT c, and $\mathfrak{G}$ is derived from $\mathfrak{G}^{(e)}$ by INSERTING THE EDGE e INTO THE CONTEXT c.

  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI which contains the closed subgraph $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0)$.

  Let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be the following graph:

  – $V' := V \backslash V_0$
  – $E' := E \backslash E_0$
  – $\nu' := \nu\big|_{E'}$
  – $\top' := \top$
  – $Cut' := Cut \backslash Cut_0$
  – $area'(d) := \begin{cases} area(d) quad d \neq \top_0 \\ area(d) \backslash (V_0 \cup E_0 \cup Cut_0) quad d = \top_0 \end{cases}.$
  – $\kappa' := \kappa\big|_{E'}$

  Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING THE SUBGRAPH $\mathfrak{G}_0$ FROM THE CONTEXT $\top_0$, and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by INSERTING THE GRAPH $\mathfrak{G}_0$ INTO THE CONTEXT $\top_0$.

  Now the rules of the calculus are restrictions of the general definitions:

  – **erasure and insertion**

    Let $\mathfrak{G}$ be an EGI and let k be an edge or a closed subgraph of $\mathfrak{G}$ with $c := ctx(k)$, and let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by erasing k from the context c. If c is positive, then $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING k FROM A POSITIVE CONTEXT, and if c is negative, then $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by INSERTING k INTO A NEGATIVE CONTEXT.

– **erasing and inserting a vertex**

  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI and let $v \in V$ be a vertex with $E_v = \emptyset$ and $\kappa(v) = \top$. Then $\mathfrak{G}_0 := (\{v\}, \emptyset, \emptyset, ctx(v), \emptyset, \emptyset, \emptyset)$ is a closed subgraph only consisting v. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by erasing v from the context $ctx(v)$. Then $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ERASING THE ISOLATED VERTEX v, and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by INSERTING THE ISOLATED VERTEX v.

- **iteration and deiteration**

  Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI with the subgraph $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0)$, and let $c \leq \top_0$ with $c \notin Cut_0$ be a context. Let $W_0 := V_0 \cap area(\top_0)$. For each vertex $v \in W_0$ let $W_v \subseteq V$ be a (possibly empty) set of vertices which satisfies $w \Theta v$ and $w \in area(c)$ for all $w \in W_v$. For each $v \in W_0$ and $w \in W_v$, let $e_{v,w}$ be a fresh edge. We set $F := \{e_{v,w} \mid v \in W_0 \text{ and } w \in W_v\}$.

  Now let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be the following graph:

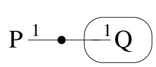
  – $V' := V \times \{1\} \ \cup \ V_0 \times \{2\}$
  – $E' := E \times \{1\} \ \cup \ E_0 \times \{2\} \ \cup \ F$
  – $\nu'(e') := \begin{cases} ((v_1, i), \ldots, (v_n, i)) \text{ for } e' = (e, i) \in E \times \{1\} \ \cup \ E_0 \times \{2\} \text{ and } \\ \qquad \nu(e) = (v_1, \ldots, v_n) \\ ((w, 1), (v, 2)) \text{ for } v \in W_0,\ w \in W_v \text{ and } e' = e_{v,w} \end{cases}$
  – $\top' := \top$
  – $Cut' := Cut \times \{1\} \ \cup \ Cut_0 \times \{2\}$
  – $area'$ is defined as follows:

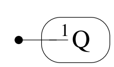
    for $(d, i) \in Cut' \cup \{\top'\}$ and $d \neq c$, let $area'((d, i)) := area(d) \times \{i\}$ and

    $area'((c, 1)) := area(c) \times \{1\} \ \cup \ area_0(\top_0) \times \{2\} \ \cup \ F$
  – $\kappa'(e') := \begin{cases} \kappa(e) \text{ for } e' = (e, i) \in E \times \{1\} \ \cup \ E_0 \times \{2\} \\ \doteq \text{ for } e' \in F \end{cases}$

  Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by ITERATING THE SUBGRAPH $\mathfrak{G}_0$ INTO THE CONTEXT c and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by DEITERATING THE SUBGRAPH $\mathfrak{G}_0$ FROM THE CONTEXT c.

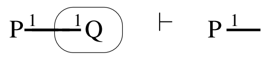
It should be noted that the iteration/deiteration-rule may be applied to arbitrary subgraphs, whilst the erasure/insertion-rule may only be applied to *closed* subgraphs. The reason is that the erasure (or insertion) of a subgraph which is not closed does not yield a well-formed EGI. This causes no troubles, as we already informally discussed in Sect. 14.1 how non-closed subgraphs can be erased. To see an simple example for EGI, consider



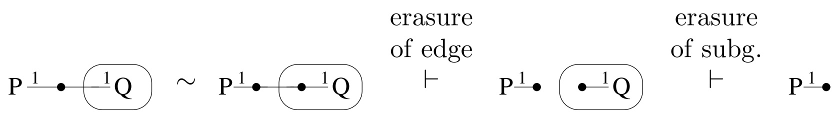
with its non-closed subgraph


.

It is not possible to erase the subgraph, as the edge labeled with P is incident with a vertex of the subgraph. but for the corresponding EGs, the following derivation is allowed:



This derivation is for EGIs performed as follows:



This idea can easily be transferred to more complex examples, thus, informally speaking, it is possible to erase non-closed subgraphs with the erasure-rule as well (and, analogously, to insert non-closed subgraphs with the insertion-rule).

Proofs are essentially defined like for Alpha. For Beta, we have to take both the transformation-rules and the rules of the calculus into account.

Definition 15.3 (Proof). *Let $\mathfrak{G}_a$, $\mathfrak{G}_b$ be EGIs. Then $\mathfrak{G}_b$* CAN BE DERIVED FROM *$\mathfrak{G}_a$ (which is written $\mathfrak{G}_a \vdash \mathfrak{G}_b$), if there is a finite sequence $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ with $\mathfrak{G}_1 = \mathfrak{G}_a$ and $\mathfrak{G}_b = \mathfrak{G}_n$ such that each $\mathfrak{G}_{i+1}$ is derived from $\mathfrak{G}_i$ by applying one of the rules of the calculus or one of the transformation rules. The sequence is called* A PROOF FOR *$\mathfrak{G}_a \vdash \mathfrak{G}_b$. Two graphs $\mathfrak{G}_1, \mathfrak{G}_2$ with $\mathfrak{G}_1 \vdash \mathfrak{G}_2$ and $\mathfrak{G}_2 \vdash \mathfrak{G}_1$ are said to be* PROVABLY EQUIVALENT *or* SYNTACTICALLY EQUIVALENT*.*

If $\mathfrak{H} := \{\mathfrak{G}_i \mid i \in I\}$ is a (possibly empty) set of EGIs, then A GRAPH $\mathfrak{G}$ CAN BE DERIVED FROM $\mathfrak{H}$ *if there is a finite subset $\{\mathfrak{G}_1, \ldots, \mathfrak{G}_n\} \subseteq \mathfrak{H}$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \vdash \mathfrak{G}$.*

Similar to the semantics, the calculus for EGIs can be transferred to a calculus for existential graphs.

Definition 15.4 (Calculus for Existential Graphs). *Let $\mathfrak{E}_a$, $\mathfrak{E}_b$ be EGs. We say that $\mathfrak{E}_b$ can be derived from $\mathfrak{E}_b$ with one of the rules of the calculus for EGIs (see Def. 15.2, if this holds for some representing EGIs $\mathfrak{G}_a \in \mathfrak{E}_a$ and $\mathfrak{G}_b \in \mathfrak{E}_b$.*

Moreover, we say that $\mathfrak{E}_b$ CAN BE DERIVED FROM *$\mathfrak{E}_a$ (which is written $\mathfrak{E}_a \vdash \mathfrak{E}_b$), if there is a finite sequence $(\mathfrak{E}_1, \mathfrak{E}_2, \ldots, \mathfrak{E}_n)$ of existential graphs with $\mathfrak{E}_1 = \mathfrak{E}_a$ and $\mathfrak{E}_b = \mathfrak{E}_n$ such that each $\mathfrak{E}_{i+1}$ is derived from $\mathfrak{E}_i$ by applying one of the rules of the calculus. The sequence is called* A PROOF FOR *$\mathfrak{E}_a \vdash \mathfrak{E}_b$.*

If $\mathfrak{H} := \{\mathfrak{E}_i \mid i \in I\}$ is a (possibly empty) set of existential graphs, then A GRAPH $\mathfrak{E}$ CAN BE DERIVED FROM $\mathfrak{H}$ *if there is a finite subset $\{\mathfrak{E}_1, \ldots, \mathfrak{E}_n\} \subseteq \mathfrak{H}$ with $\mathfrak{E}_1 \ldots \mathfrak{E}_n \vdash \mathfrak{E}$.* [1]

[1] Recall that the juxtaposition of existential graphs is carried over from the definition of the juxtaposition for EGIs, as it has been explained directly after Def. 12.15.

If we have a proof $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ for two EGIs $\mathfrak{G}_a$, $\mathfrak{G}_b$ with $\mathfrak{G}_a \vdash \mathfrak{G}_b$, then this proof immediately yields a proof for $[\mathfrak{G}_a]_\sim \vdash [\mathfrak{G}_b]_\sim$: We start with the proof $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$. From the sequence $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$, we remove all graphs $\mathfrak{G}_i$ which are derived from $\mathfrak{G}_{i-1}$ with a transformation-rule. From the remaining subsequence $(\mathfrak{G}_{i_1}, \mathfrak{G}_{i_2}, \ldots, \mathfrak{G}_{i_k})$, we obtain the proof $([\mathfrak{G}_{i_1}, [\mathfrak{G}_{i_2}]_\sim, \ldots, [\mathfrak{G}_{i_k}]_\sim)$ for $[\mathfrak{G}_a]_\sim \vdash [\mathfrak{G}_b]_\sim$. First examples for this principle will be given in the next chapter. A thoroughly discussion on how the calculus for EGIs is be used for formal existential graphs will be provided in Chpt. 21.

16

Improving the Handling of Ligatures

Before the soundness of the calculus for EGIs is proven the next chapter, we will investigate more deeply the understanding and handling of ligatures. This will be first done for EGIs, and the results for EGIs will then be transferred to EGs. In the first section of this chapter, we will derive some rules which allow to rearrange ligatures in various ways. In the second section, we will use the results of the first section to investigate how ligatures can be better understood.

16.1 Derived Rules For Ligatures

In Sect. 14.2, we had already discussed some examples which show that the iteration/deiteration-rule allows to rearrange ligatures in Peirce's graphs. The ideas behind these examples are mathematically elaborated in this section.

We start with the mathematical elaboration of 'moving branches along a ligature' for EGIs, as it had been discussed for Peirce's graphs on page 157 ff. Let an EGI $\mathfrak{G}$ with a ligature in a cut c be given. Then any branch of this ligature may be disconnected from that ligature and connected to it again in an arbitrary other place on that ligature. This is formally captured and proven in the next lemma.

Lemma 16.1 (Moving Branches along a Ligature in a Context). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI, let v_a, v_b be two vertices with $c := ctx(v_a) = ctx(v_b)$ and $v_a \Theta v_b$, and let e be an edge such that the hook (e, i) is attached to v_a. Let $\mathfrak{G}' := (V, E, \nu', \top, Cut, area, \kappa)$ be obtained from $\mathfrak{G}$ by replacing v_a by v_b on the hook (e, i). Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

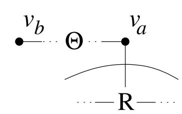
Proof: The proof of this lemma is quite easy. To exemplify the graphical notation for proofs, this proof provided in two different ways. In each application of a rule of the calculus, the result is given as mathematical structure as well as a digram representing the mathematical structure. In order to keep the proof readable, we use in both notations some further notational conventions, which shall be explained first.

In the proof, the graph $\mathfrak{G}$ will be modified only in $area(v_a)$ ($=area(v_b)$), and the edge e (more precisely: the value $\nu(e)$ of the function ν) will be changed. For this reason, we set $\nu_0 := \nu\backslash\{(e,\nu(e))\}$ and $area_0 := area\backslash\{(c, area(v_a))\}$. Moreover, as only the hook (e,i) is changed, we will write '$(e,(\ldots,v,\ldots))$' to indicate that we have $\nu(e)\big|_i = v$.

In the graphical notation, only the part of the graph $\mathfrak{G}$ are shown which are relevant for the proof. All vertex-spots and edges-lines are labeled with their corresponding vertices and edges. The relation $v_a \Theta v_b$ is depicted in the graphical notation with the symbol Θ between the vertex-spots of v_a and v_b. To indicate that e might be placed in a deeper context than $ctx(v_a)$, the relation-symbol (we have chosen the letter R) of the edge e is placed in the segment of a cut-line.

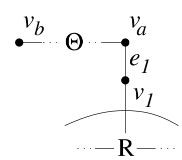
We start the proof with the graph $\mathfrak{G}$:

$(V, E,$
$\nu_0 \dot\cup \{(e,(\ldots,v_a,\ldots))\},$
$\top, Cut,$
$area_0 \dot\cup \{(c, A)\},$
$\kappa\ )$



We add a vertex v_1 (and an identity-edge e_1) between v_a and the hook (e,i) with the transformation-rule 'adding a vertex'. This yields:

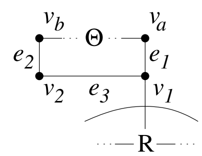
$(V \dot\cup \{v_1\},$
$E \dot\cup \{e_1\},$
$\nu_0 \dot\cup \{(e,(\ldots,v_1,\ldots)),(e_1,(v_a,v_1))\},$
$\top, Cut,$
$area_0 \dot\cup \{(c, A \dot\cup \{v_1, e_1\})\},$
$\kappa \dot\cup \{(e_1, \doteq)\}\ )\}$



We iterate the vertex v_1. The copy of v_1 is named v_2, and an identity-edge between v_2 and v_1 and an identity-edge between v_2 and v_b is added. This yields the following graph:[1]

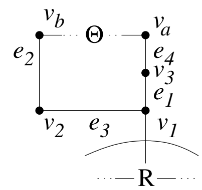
[1] This graph is not the 'direct' result of an application of the iteration-rule, as we considered in the definition of the rule the product $V \times \{1\}$ and $V_0 \times \{2\}$. But we consider graphs only up to isomorphism, thus the next graph, which is isomorphic to the 'direct' result of the iteration-rule, is considered to be a result of the iteration-rule as well.

$(V \dot{\cup} \{v_1, v_2\}$
$E \dot{\cup} \{e_1, e_2, e_3\},$
$\nu_0 \dot{\cup} \{(e, (\ldots, v_1, \ldots)), (e_1, (v_a, v_1)),$
$\quad (e_2, (v_2, v_b)), (e_3, (v_2, v_1))\},$
$\top, Cut,$
$area_0 \dot{\cup} \{(c, A \dot{\cup} \{v_1, v_2, e_1, e_2, e_3\})\},$
$\kappa \dot{\cup} \{(e_1, \doteq), (e_2, \doteq), (e_3, \doteq)\})$



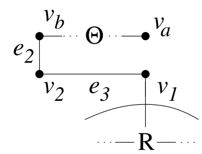
With the transformation-rule 'adding a vertex', we add a vertex v_3 (and an identity-edge e_4) between v_a and v_1 as follows:

$(V \dot{\cup} \{v_1, v_2, v_3\}$
$E \dot{\cup} \{e_1, e_2, e_3, e_4\},$
$\nu_0 \dot{\cup} \{(e, (\ldots, v_1, \ldots)), (e_1, (v_3, v_1)),$
$\quad (e_2, (v_2, v_b)), (e_3, (v_2, v_1)), (e_4, (v_a, v_3))\},$
$\top, Cut,$
$area_0 \dot{\cup} \{(c, A \dot{\cup} \{v_1, v_2, v_3, e_1, e_2, e_3, e_4\})\},$
$\kappa \dot{\cup} \{(e_1, \doteq), (e_2, \doteq), (e_3, \doteq), (e_4, \doteq)\})$



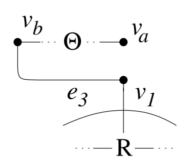
The vertex v_3 and the two identity-edges e_4 and e_1 could be the result of an iteration of v_2. Thus they can be erased with the rule of deiteration.

$(V \dot{\cup} \{v_1, v_2\}$
$E \dot{\cup} \{e_2, e_3\},$
$\nu_0 \dot{\cup} \{(e, (\ldots, v_1, \ldots)), (e_2, (v_2, v_b)), (e_3, (v_2, v_1))\},$
$\top, Cut,$
$area_0 \dot{\cup} \{(c, A \dot{\cup} \{v_1, v_2, e_2, e_3\})\},$
$\kappa \dot{\cup} \{(e_2, \doteq), (e_3, \doteq)\})$



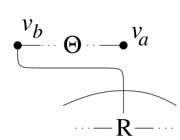
The vertex v_2 and the edge e_2 are removed with the transformation-rule 'removing a vertex'. This yields:

$(V \dot{\cup} \{v_1\},$
$E \dot{\cup} \{e_3\},$
$\nu_0 \dot{\cup} \{(e, (\ldots, v_1, \ldots)), (e_3, (v_b, v_1))\},$
$\top, Cut,$
$area_0 \dot{\cup} \{(c, A \dot{\cup} \{v_1, e_3\})\},$
$\kappa \dot{\cup} \{(e_3, \doteq)\})$



Similar to the last step, the vertex v_1 and the edge e_3 are removed with the transformation-rule 'removing a vertex'. This yields:

$(V, E,$
$\nu_0 \dot{\cup} \{(e, (\ldots, v_b, \ldots))\},$
$\top, Cut,$
$area_0 \dot{\cup} \{(c, A)\},$
$\kappa)$



This is $\mathfrak{G}'$, thus we are done. □

The proof should not only prove the lemma, but also show how the modifications of the mathematical structures are represented by the diagrams. As the rules of the calculus correspond to modifications of the diagrams, the graphical notation is easier to understand than the symbolic notation. For this reason in all ongoing proofs, we will mainly use the graphical representations of EGIs.

For Peirce's graphs, we have seen on page 156 in Fig. 14.4 that some simple applications of the iteration-rule can add new branches to a given ligature, and these branches may be connected several times to the that ligature. This will now be elaborated for EGIs.

Informally speaking, we can do the following: If in an EGI an arbitrary ligature is given, we can pick out a vertex v of this ligature, and the rule of iteration allows iterate v, i.e., to add a new vertex v' and new identity-links between v' and the given ligature. This process can be applied to the new vertices as well, and by an iterative application of the iteration-rule, we can finally add a new, arbitrarily connected network of LoIs, i.a., a ligature, to our given ligature. This idea is captured by the next lemma.

Lemma 16.2 (Extending or Restricting a Ligature in a Context). *Let a EGI $\mathfrak{G}$ be given with a vertex v. Let V' be a set of fresh vertices and E' be be a set of fresh edges. Let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be obtained from $\mathfrak{G}$ such that all fresh vertices and edges are placed in the context $ctx(v)$, and all fresh edges are identity edges between the vertices of $\{v\} \dot\cup V'$ such that we have $v\Theta v'$ for each $v' \in V'$. That is, $\mathfrak{G}'$ satisfies:*

- $V' := V \dot\cup V'$ *and* $E' := E \dot\cup E'$
- $\nu' \supseteq \nu$ *and* $\nu'(e') \in (\{v\} \dot\cup V')^2$ *for each* $e' \in E'$
- $\top' := \top$ *and* $Cut' := Cut$
- $area'(d) := \begin{cases} area(d) \text{ for } d \neq ctx(v) \\ area(d) \dot\cup E' \dot\cup V' \text{ for } d = ctx(v) \end{cases}$.
- $\kappa' := \kappa \ \dot\cup \ E' \times \{\doteq\}$

Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.

Proof: For each $v' \in V'$ let $d(v')$ be the length of the shortest path (in $\mathfrak{G}'$) from v' to v along edges in E', and we set $V'_i := \{v' \in V' \mid d(v') = i\}$. Let m be the maximal number with $V'_m \neq \emptyset$. Now, V' and E' can be inductively be constructed with the rule of iteration, applied to v: In the kth step, we can add all vertices of V'_k and all fresh edges which are (in $\mathfrak{G}'$) incident in one place with a vertex of V'_k and in the other place only with a vertex in $\bigcup_{i=1}^{k-1} V'_i$. If this induction is done, it remains to add all fresh edges who are in both places incident with one vertex of V'_n. Let e' be such an fresh edge. This edge can be added as follows: The rule of iteration is applied, and a new vertex w and two identity-links between w and $e'\big|_1$ resp. between w and $e'\big|_2$ is added, and afterwards, the transformation-rule 'removing a vertex' is applied to v' and one of the just added identity-links. The remaining-identity-link is incident with $e\big|_1$ and $e\big|_2$, thus is can be replaced by e.

As we have constructed $\mathfrak{G}'$ from $\mathfrak{G}$ only with the iteration-rule and the transformation-rule 'removing a vertex', $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent. □

With the last two lemmata, it is possible to 'retract' a ligature in a context to a single vertex, and vice versa, which is the subject of the next lemma.

Lemma 16.3 (Retracting a Ligature in a Context). *Let $\mathfrak{G}$ be an EGI. Let (W, F) be a ligature which is placed in a context c, i.e., $ctx(w) = c = ctx(f)$ for all $w \in W$ and $f \in F$, and let $w_0 \in W$. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ as follows: The ligature (W, F) is* RETRACTED TO *w_0, i.e., all vertices of $W \backslash \{w_0\}$ and all edges of F are removed from c, and if an edge $e \in E \backslash F$ was incident with a vertex $w \in W$ with $w \neq w_0$, it is connected to the vertex w_0. That is, $\mathfrak{G}'$ is defined as follows:*

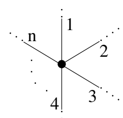
- $V' := V \backslash W \mathbin{\dot\cup} \{w_0\}$
- $E' := E \backslash F$
- *We define ν' as follows: For $\nu(e) = (v_1, \ldots, v_n)$, let $\nu'(e) = (v'_1, \ldots, v'_n)$ with* $v'_i := \begin{cases} v_i \text{ for } v_i \notin W \\ w_0 \text{ for } v_i \in W \end{cases}$.
- $\top' := \top$ *and* $Cut' := Cut$
- $area'(d) := \begin{cases} area(d) \text{ for } d \neq c \\ area(c) \backslash (W \mathbin{\dot\cup} F) \mathbin{\dot\cup} \{w_0\} \text{ for } d = c \end{cases}$.
- $\kappa' := \kappa\big|_{E'}$

Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.

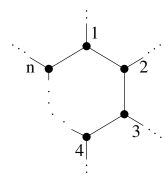
Proof: Let (e, i) be an arbitrary hook with $e \notin F$ and which is attached to a vertex $w \in W$ with $w \neq w_0$. Using Lem. 16.1, we can replace w by w_0 on the hook (e, i). If this is done for all such hooks, we obtain a graph $\mathfrak{G}''$ which is syntactical equivalent to $\mathfrak{G}$, and for which no vertex $w \in W$ with $w \neq w_0$ is incident with an edge $e \notin F$. Thus we can remove all vertices in $W \backslash \{w_0\}$ and all edges in F with one application of Lem. 16.2. The graph $\mathfrak{G}'$ we obtain is the desired graph, thus we are done. □

It should be noted that the last lemma can be used in both directions: It does not only allow to retract a ligature in a context to a single vertex; vice versa, it allows to replace a single vertex by a ligature. An important consequence of this direction is the possibility to avoid branching points with more than three branches. For Peirce's graphs, it has already been discussed on page 156 how such branching points can be converted into ligatures having only branching points with three branches. Now this can be be done for EGIs as well.

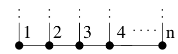
Lemma 16.3 allows to convert a single vertex which is attached to more than three hooks into a ‘wheel’ or a ’fork’, i.e.,



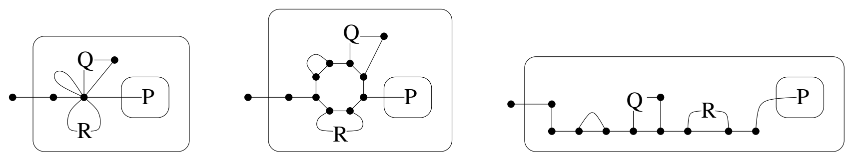
can be converted into



or



For example, this application of Lem. 16.3 yields that the following three graphs are provably equivalent:



Finally, we can use Lem. 16.3 to convert complete ligatures within the area of a given context. This shall first be formally defined.

Definition 16.4 (Rearranging Ligatures in a Context). *Let $\mathfrak{G}$ be an EGI. Let (W, F) be a ligature which is placed in a context c, i.e., $ctx(w) = c = ctx(f)$ for all $w \in W$ and $f \in F$. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ as follows: The ligature (W, F) is replaced by a new ligature (W', F'), i.e., all vertices of W and all edges of f are removed from c, the vertices of W' and edges of E' are inserted into c, and if an edge $e \in E \backslash F$ was incident with a vertex $w \in W$ of the ligature, it is now connected to a vertex $w' \in W'$ of the new ligature. That is, $\mathfrak{G}'$ satisfies:*

- $V' := V \backslash W \,\dot\cup\, W'$
- $E' := E \backslash F \,\dot\cup\, F'$
- *For ν', it holds the following: For $e \in E \backslash F$, $\nu(e) = (v_1, \ldots, v_n)$, and $\nu'(e) = (v'_1, \ldots, v'_n)$, we have $v'_i := v_i$, if $v_i \notin W$, and $v'_i \in W'$, if $v_i \in W$*
- $\top' := \top$ *and* $Cut' := Cut$
- $area'(d) := \begin{cases} area(d) \text{ for } d \neq c \\ area(c) \backslash (W \,\dot\cup\, F) \,\dot\cup\, (W' \,\dot\cup\, F') \text{ for } d = c \end{cases}.$
- $\kappa' := \kappa\big|_{E \backslash F} \,\dot\cup\, F' \times \{\doteq\}$

Then we say that $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by REARRANGING THE LIGATURE (W, F) (TO (W', F')).

If $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by rearranging a ligature, it can easily seen that these graphs are syntactically equivalent: First we can retract with Lem. 16.3

the ligature (W, F) to a single vertex $w_0 \in W$, afterwards w_0 can be extended with the opposite direction of Lem. 16.3 to the new ligature (W', F'). I.e., we have:

Corollary 16.5 ([Rearranging Ligatures in a Context). *If $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by rearranging a ligature in a context, then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

To summarize this result in sloppy way: A ligature in a context may be arbitrarily changed, as long as it keeps connected.

So far in this section, we have only modified ligatures which are wholly contained in the area of a given context. At the end of Sec. 14.2, we have already discussed for Peirce's graphs how branching points of a ligature can be moved across cuts. The idea of moving branching points can now for EGIs be mathematically elaborated and canonically be generalized.

In Sec. 14.2, we have moved a branching point into a cut for which two branches go from the branching point into the cut, but instead of exactly two, we may consider an arbitrary number of such edge-lines. Moreover, the branching point may move more than one cut inwardly. This generalization is mathematically captured by the following definition.

Definition 16.6 (Splitting a Vertex/Merging two Vertices). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI, let $v \in V$. Let $(e_1, i_1), \ldots, (e_n, i_n)$ be an enumeration of some (not necessarily all) hooks v is attached to. Let c be a context with $ctx(v) \geq c$ and $c \geq ctx(e_k)$ for all $1 \leq k \leq n$, and let v' be a fresh vertex and e' be a fresh edge. Now let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be obtained from $\mathfrak{G}$ as follows:*

- $V' := V \dot{\cup} \{v'\}$
- $E' := E \dot{\cup} \{e'\}$
- *ν' is defined as follows: $\nu'(e') = (v, v')$, and for $f \neq e'$, we set*
$$\nu'(f)\big|_j := \begin{cases} v' \text{ for } (f, j) = (e_i, n_i),\ 1 \leq i \leq n \\ \nu(f)\big|_j \text{ for else} \end{cases},$$
- $\top' := \top$ *and* $Cut' := Cut$
- *$area'(d) := area(d)$ for $d \neq c$, and $area'(c) := area(c) \dot{\cup} \{v', e'\}$, and*
- $\kappa' := \kappa \dot{\cup} \{(e', \doteq)\}$.

Then we say that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by MERGING v_1 INTO v_2 *and $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by* SPLITTING v_1.

Slightly more informally, but easier to understand, the rules can be described as follows:

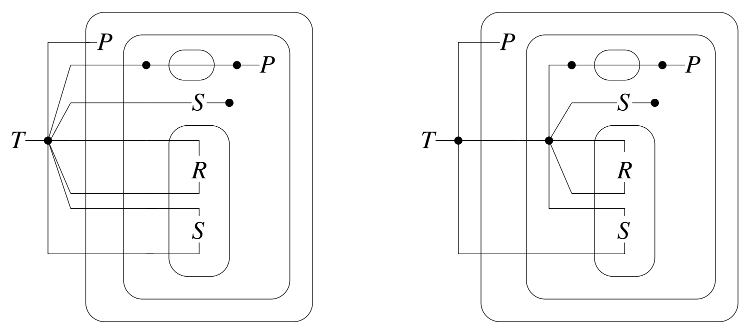
- **splitting a vertex**

  Let v be a vertex in the context c_0 attached to hooks $(e_1, i_1), \ldots, (e_n, i_n)$, placed in contexts $c_1, \ldots, c_n$, respectively. Let c be a context such that $c_1, \ldots, c_n \leq c \leq c_0$. Then the following may be done: In c, a new vertex v' and a new identity-link between v and v' is inserted. On the hooks $(e_1, i_1), \ldots, (e_n, i_n)$, v is replaced by v'.

- **merging two vertices**

  Let $e \in E^{id}$ be an identity edge with $\nu(e) = (v_1, v_2)$ such that $ctx(v_1) \geq ctx(e) = ctx(v_2)$. Then v_2 may be merged into v_1, i.e., v_2 and e are erased and, for every edge $e \in E$, $e\big|_i = v_1$ is replaced by $e\big|_i = v_2$.

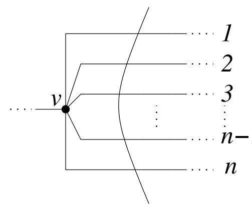
In the next example, the right graph is obtained from the left graph by splitting the outermost vertex. Please note that if a vertex v is incident more than once with an edge, arbitrary occurrences of v are substituted by its copy: For R, both occurrences are substituted, while for S, only one occurrence is substituted.



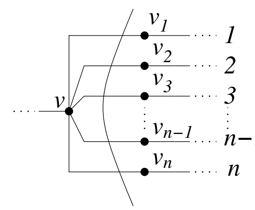
If the rule 'splitting a vertex' is applied such that exactly one occurrence of v on an edge is substituted by its copy v', then this corresponds to an application of the transformation-rule 'adding a vertex'. Thus, the rules 'splitting a vertex' and 'merging two vertices' can be understood to be a generalization of the transformation-rules 'adding a vertex/removing a vertex'. These rules can from derived from the calculus, as the next lemma shows.

Lemma 16.7 (Splitting a Vertex/Merging two Vertices). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI, let v in V, and let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by splitting the vertex v. Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

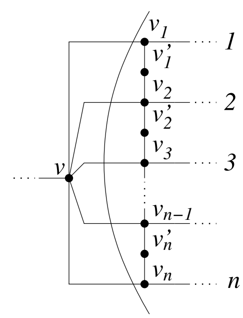
Proof: In diagrams for EGIs, each hook (e_k, i_k) corresponds to an edge-line between the vertex-spot of v to the symbol of the relation $\kappa(e)$. In the graphical notation for this proof, each hookline (e_k, i_k) is labeled with n, the relation-symbols of the edges e_k are not shown. In order to indicate that we might have $c < ctx(v)$, a cut-segment is drawn, but keep in mind that this cut-segment may stand for more than one cut, or even no cut (roughly speaking, the proof would work as well if we had no or many cut-lines instead of exactly one). The area right from the cut-line shall represent the area of c. So the starting graph $\mathfrak{G}$ is depicted as follows:



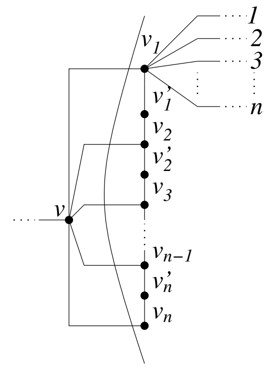
With the transformation-rule 'adding a vertex', for each $1 \leq k \leq n$, a vertex v_i (and an identity-edge id_i) is added between v and the hook (e_k, i_k), and v_i is placed in $area(c)$.



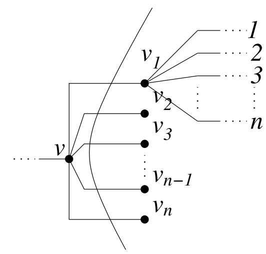
Now the vertex v is $n-1$-times iterated. The ith copy of v is placed in the context c and shall be called v_i'. In the ith iteration, an identity-edge between v_i' and v_i and an identity-edge between v_i' and v_{i+1} is inserted.



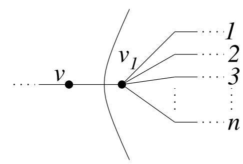
The part of the ligature inside the context c can now be rearranged.



The iteration of the last but one step of this proof can be undone with the rule of deiteration.



Again with the rule of deiteration, we can now erase the vertices v_i and the identity-edges id_i between v and v_i.

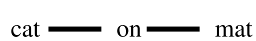


The last part of the graph is the desired result. Note that the proof only needs the rules of iteration and deiteration and the transformation-rules, thus it can be carried out in both directions. □

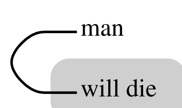
16.2 Improving the Reading of Ligatures

Based on the results of the last sections, we can now derive some methods which ease the understanding and handling of ligatures.

In Sec. 11.1, we have seen that, strictly speaking, only identity spots are used to denote objects. Due to Peirce's convention No. 6, a line of identity denotes the identity of the objects denoted by its two extremities, and, as we have seen, this convention turned out to be a conclusion from Peirce's deeper understanding of the inner structure of LoIs. Nonetheless, in the common reading, a line of identity is usually read as if they simply denote a single object. The very first example of a beta graph in this treatise was



The graph is read as follows: A cat is on a mat. It contains two lines of identity, each of them is understood to denote a single object. Even when a heavy line crosses a cut, it can be interpreted this way. On page 100, we have already seen the graph

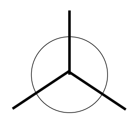


This graph can be read as follows: There is a man who will not die. That is, the whole heavy line is understood to denote a single object, namely an immortal man. Analogously, the ligatures in the (equivalent) graphs of Fig. 14.1

on page 152 stand for a single object. Recall that Peirce admits that a more complex reading, unfolding the understanding that a LoI is composed of overlapping identity spots, is '*unspeakably trifling, – not to say idiotic*'.

In Peirce's graphs, heavily drawn lines crossing a cut or networks of heavily drawn lines are called ligatures. Ligatures with branches can, similar to the last two examples, usually be understood to stand for a single object. For example, the complex left ligature in Fig. 11.1 on page 102 stands for Aristotle.

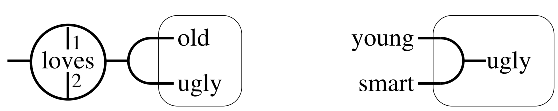
Nonetheless, other examples show that this interpretation of ligatures is not in every case that simple. A ligature may stand for more than one object. The most simple example where such a ligature occurs is the well-known graph $\mathfrak{E}_{twothings}$, where a single, heavy line traverses a cut (see page 97). This graph has the meaning 'there are at least to things', i.e., in this graph, the ligature does not stand for a single object. Analogously, we have already seen on page 116 that in the graph of Fig.118 in 4.469, i.e.,



,

the ligature cannot be understood to denote one object (but it cannot be understood to stand for three objects, neither, as the meaning of this graph is 'there are three things which are not all identical'). So, naturally the following question arises: When can a ligature be interpreted to denote a single object? With the results of the preceeding section, this question can now be answered.

The clue to the answer can already been found in the discussed examples. We have already seen that a heavy line traversing a cut denotes *non-identity*, for this reason, such a heavy line cannot be understood to denote one object. The same holds for the graph of Fig.118 in 4.469: It denotes that the three ends of the ligature do not stand for a single object. But ligatures which do not traverse[2] any cut. In the following, we will formal define ligatures which do not traverse any cut. As we will prove that this ligatures can be understood to denote a single object, they are called *single-object ligatures*. For example, the ligature in the left graph below is a single-object ligature, while the ligature in the right graph is not.



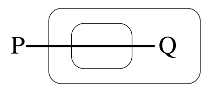
Now we have to formally define single-object ligatures for EGIs.

[2] A ligature traverses a cut c if there is a heavily drawn line l being a part of the ligature both such that both endpoints of l are placed on c and the remainder of l is enclosed by c.

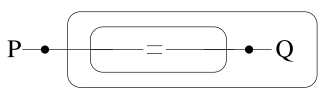
Definition 16.8 (Single-Object Ligatures). *Let (W, F) be a ligature of the EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$. Then it is called a* SINGLE-OBJECT LIGATURE*, iff*

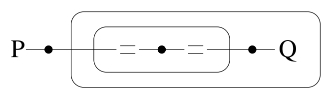
- *there are no (not necessarily different) $w_1, w_2 \in W$ and an identity-link $f \in F$ with $w_1\, f\, w_2$, $f < w_1$ and $f < w_2$,*
- *there are no $w_1, w_2, w \in W$ and $f_1, f_2 \in F$ with $w_1 \neq w_2$, $w_1\, f_1\, w\, f_2\, w_2$, $w < w_1$ and $w < w_2$, and*
- *there are no $w_1, w_2 \in W$ which are part of a cycle in (W, F) and for which we have $w_2 < w_1$.*

Recall how the graphical representations of EGIs are converted to Peirce's Beta graphs, as it has been described on page 133. A ligature in an EGI is converted to a network of heavy lines, and this network crosses each cut-line almost once if and only if the ligature is a single-object ligature. To see an example for this definition, consider the following Peircean Beta graph:



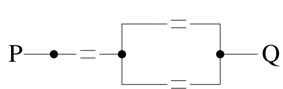
This graph has the meaning 'there is an object (with property) P which is distinct from all all objects (with property) Q'. The ligature traverses the inner cut. The following two EGIs are possible formalizations of this graph:

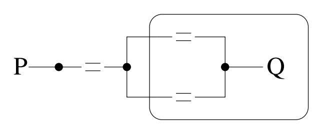




In the left graph, the ligature is not a single-object ligature, as it violates the first condition of Def. 16.8. Similarly, in the right graph, the ligature violates the second condition of Def. 16.8.

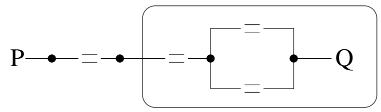
To see why the third condition of Def. 16.8 is needed, consider the following two examples:

$\mathfrak{G}_1 :=$ 

$\mathfrak{G}_2 :=$ 

A single-object ligatures still may contain cycles, as long as all vertices and edges of a cycle are placed in the area of single context. The ligature of $\mathfrak{G}_1$ is a single-object ligature, as all vertices and edges are placed on the sheet of assertion, i.e., the are placed in the same context. Even this ligature denotes a single object. But $\mathfrak{G}_2$ violates the third condition, and indeed, in the corresponding Peircean Beta graph, the ligature crosses the cut-line twice.

Nonetheless, this graph can be transformed with the rule 'splitting a vertex' into the semantically equivalent graph



,

thus we see that in $\mathfrak{G}_2$, the ligature denotes a single object as well. Being a single-object ligature is thus a sufficient, but not necessary condition to stand for a single object.

Let us first fix a simple fact about single-object ligatures. Let (W, F) be a ligature in an EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$. Let w_a, w_b be two vertices of W. As (W, F) is connected, there are $w_1, w_2, \ldots, w_n \in W$ and $f_1, f_2, \ldots, f_{n-1} \in F$ with $w_a = w_1\, f_1\, w_2\, f_3\, \ldots\, w_{n-1}\, f_{n-1}\, f_n = w_b$. Due to Def. 16.8, there can be no $k \in \{w_1, w_2, \ldots, w_n, f_1, f_2, \ldots, f_{n-1}\}$ with $k < w_a$ and $k < w_b$. From this we immediately obtain that the set of context $\{ctx(w) \mid w \in W\}$ contains a greatest element. In the following, this element is denoted by $ctx(W, F)$.

In Lem. 16.3 we have proven that ligatures which are wholly placed in one context can be retracted to a single vertex. Thus, in the light of Lem. 16.3, such ligatures can be understood to denote a single object. Now we can extend this lemma to the following lemma which states that even single-object ligatures can retracted to a single vertex. Thus this lemma, which is an extension of Lem. 16.3, elaborates mathematically why and how single-object ligatures can be understood to denote a single object.

Lemma 16.9 (Retracting a Single-Object Ligature). *Let a EGI $\mathfrak{G}$ be given and let (W, F) be a single-object ligature of $\mathfrak{G}$. Let $w_0 \in W$ be a vertex with $ctx(w_0) = ctx(W, F)$. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by retracting (W, F) to w_0 (the formal definition of $\mathfrak{G}'$ is exactly like in Lem. 16.3, where the condition that (W, F) is placed in a single context is dropped). Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

Proof: The proof is carried out in two steps.

Assume first that (W, F) contains a ligature $(\overline{W}, \overline{F})$ (i.e. $(\overline{W}, \overline{F})$ is a subgraph of (W, F)) which is wholly placed in a context c. Then, with Lem. 16.3, we can retract $(\overline{W}, \overline{F})$ to a single vertex $\overline{w}$ (if we have $w_0 \in \overline{W}$, we have to choose $w := w_0$, as we want to finally retract (W, F) to w_0). As this transformation does not change the context of $\overline{w}$ or any other vertex or edge which does not belong to $\overline{W} \cup \overline{F}$, the remaining ligature $(W \backslash \overline{W} \cup \{\overline{w}\}, F \backslash \overline{F})$ is still a single-object ligature in the graph we obtain. We perform this transformation as often as necessary until we obtain a graph $\mathfrak{G}_1 := (V_1, E_1, \nu_1, \top_1, Cut_1, area_1, \kappa_1)$ and single-object ligature (W_1, F_1) which does not contain any non-trivial ligatures (i.e., ligatures which contain an edge) which are placed within a single context.

The ligature (W_1, F_1) does not contain any cycles: Circles contained in one context have just been retracted, and circles which are not completely contained in one context cannot occur due to the third condition of Def. 16.8. Analogously (now with the first condition of Def. 16.8), the ligature (W_1, F_1) does not contain any loops. Thus (W_1, F_1) is a tree (in the graph-theoretical sense). Particularly, we we have $|W_1| = |F_1| + 1$.

The tree (W_1, F_1) can be understood as a tree in the order-theoretical understanding as well, which now shall be elaborated. For two vertices $w, w' \in W$, we set

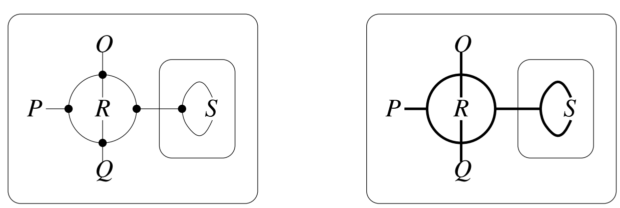
$$w \sqsubseteq w' :\Longleftrightarrow \begin{array}{l} \text{if there are } w_1, \ldots, w_n \in W_1, f_1, \ldots, f_{n-1} \in F_1 \\ \text{with } w = w_1\, f_1\, w_2 \ldots w_{n-1}\, f_{n-1}\, w_n = w' \\ \text{and } ctx(w_1) \geq ctx(f_1) \geq ctx(w_2) \geq \ldots \geq ctx(w_n) \end{array}$$

This relation is trivially reflexive, it is anti-symmetric, as we have to two different vertices which are placed in the same context and which are linked with an identity edge, and at two comparable elements, they are graph-theoretically connected in the tree (W_1, F_1), the order $\sqsubseteq$ is a tree with g_0 as greatest element.

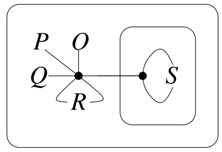
Assume $F_1 \neq \emptyset$ (otherwise we are finished). Let w be a leaf of (W_1, F_1) (with respect to $\sqsubseteq$). Then, there exists exactly one vertex $w' \in W_1$ and exactly one identity edge $f \in F_1$ with $w'\, f\, w$. Moreover, we have $w' > w$. Thus, we can merge w into w' with the rule 'merging two vertices', and this transformation rule removes w and f. The remaining ligature is still a tree, in the graph-theoretical as well as in the order-theoretical way. Thus, we can successively merge the leafs of tree to decrease its size, until there is only one leaf left. This must be the vertex w_0, i.e. the graph we obtain is the desired graph $\mathfrak{G}'$.

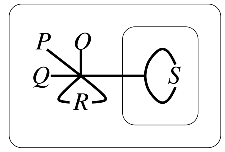
Both transformations we used in this proof (applying Lem. 16.3 and merging two vertices) can be carried out in both directions, hence $\mathfrak{G}$ and $\mathfrak{G}'$ are equivalent. $\square$

In the following, an example for Lem. 16.9 and its proof shall be provided. Consider the following EGI and the corresponding Peirce graph:

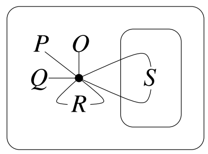


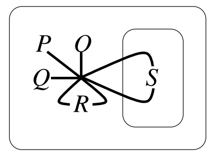
The ligature in the EGI is drawn unnecessarily complicated, but it is easy to see that it is a single-object ligature. In the first step of the proof, the part of the ligature which is placed in the outermost cut is retracted. We obtain:





In the next step, the remaining vertices (it is only one) are merged into the vertex of the ligature which is placed in the outermost cut. In this step, we get:





After the last step, the whole ligature is retracted to a single vertex. So we see that the graph we obtain, thus the starting graph as well, can be read as follows: 'Each object which has the properties O, P and Q and which stands in relation R to itself stands in relation S to itself as well'.

If a ligature is not a single-object ligature, it can be divided into single-object ligatures. This shall be elaborated now.

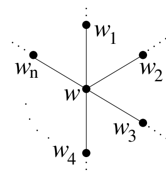
To separate a non-single-object ligature into several single-object ligatures, a simple technical trick is used: We add further identity-relations to our alphabet, and the relation symbols will be used to break up a ligature.

Let $\mathcal{A}$ be an arbitrary alphabet. For each $k \geq 2$, we add k-ary identity-relation $\doteq_k$ to $\mathcal{A}$. The resulting alphabet shall be denoted by $\mathcal{A}^{=}$. The symbol $\doteq_k$ shall be interpreted in the models as the k-ary identity-relation Id_k^U on U, i.e. we extend the relational structures (see Def. 13.1) as follows: For the interpretation function I and each $k \geq 2$, we set

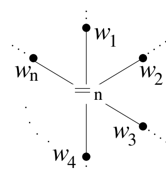
$$Id_k^U := \{(u_1, u_2, \ldots, u_k) \in U^k \mid u_1 = u_2 = \ldots = u_k\} \quad \text{and} \quad I_{\mathcal{R}}(\doteq_k) := Id_k^U \quad .$$

Of course, the relations we assign to $\doteq$ and $\doteq_2$ coincide. The mere difference is that in EGs, the symbol $\doteq_2$ is treated like any other relation symbol, i.e., it well explicitely written in any diagrammatic representation of an EGI (this will become clear immediately).

A STAR OF ARITY n WITH CENTER v IN THE CONTEXT c is a ligature (W, F) with $W = \{w, w_1, \ldots, w_n\}$ such that all $w, w_1, \ldots, w_n$ are pairwise distinct, $F = \{f_1, \ldots, f_n\}$, $w\, f_1\, w_1$, $w\, f_2\, w_2$, $\ldots$,$w\, f_n\, w_n$, $c = ctx(w) = ctx(f_1) = \ldots = ctx(f_n)$, $c \leq ctx(w_1), \ldots, ctx(w_n)$ and w is in the graph only incident with $f_1, \ldots, f_n$. If we omit the cut-lines from possibly existing cuts between c and the contexts of the vertices w_i, a star can graphically be depicted as follows:



The basic idea to break up ligatures is to replace the center of an n-ary star by an n-ary identity relation $\dot{=}_n$. That is, we remove w_0 and the identity edges $f_1, \ldots, f_n$ from c, and we insert a new $n - ary$ edge f into c with $\nu(f) := (w_1, \ldots, w_n)$ and $\kappa(f) := \dot{=}_n$. The resulting graph can be depicted as follows:



In the following, this transformation will be called SEPARATING A LIGATURE AT THE VERTEX v It may be carried out in the other direction as well: Then we will say that we JOIN LIGATURES AT THE EDGE f'. The soundness of these transformations is easy to see, formally it it proven analogously to the soundness of the transformation rules for ligatures (see Lem. 13.9). Moreover, with this transformation, it is possible to transform each graph over an alphabet $\mathcal{A}^=$ into an semantically equivalent graph over $\mathcal{A}$ as follows: If an edge $f = (w_1, \ldots, w_n)$ labeled with $\dot{=}_n$ is given, using the transformation rule 'adding a vertex to a ligature', we insert a new vertex and a new identity edge between each vertex w_i and the corresponding hook (f, i); then we can join the ligatures at the edge f. For this reason, once we have proven the soundness and completeness of the calculus for EGIs over $\mathcal{A}$, the calculus together with the transformation rules 'separating a ligature at the vertex v' and 'joining ligatures at the edge f' is sound and complete for EGIs over $\mathcal{A}^=$.

Introducing the relation-symbols $\dot{=}_n$ is a simple technical means to make clear how a non-single-object ligature is separated into several single-object ligatures. Particularly, this means helps to better understand the meaning of Peirce's graphs. This shall be exemplified now. We start with the well known graphs of Fig. 16.1, whose meanings have already been investigated in Chpt. 11, p. 11.3 ff.

We have already discussed in Chpt. 11 that the left EGI of the Fig. 16.2 is the best readable formalization of the left Peirce graph in Fig. 16.1. Moreover, in the investigation of the right graph of Fig. 16.2 (see p. 116,) I have written that 'if we had a symbol $\dot{=}_3$ for teridentity the graph could even simpler be formalized as follows:', and provided the right graph of Fig. 16.2.

Fig. 16.1. Two Peirce graphs with non-single-object ligatures

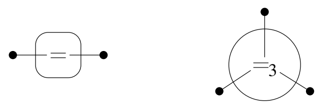


Fig. 16.2. Two EGIs where the ligatures are split into single-object ligatures

The introduction of the new identity symbols $\dot{=}_3$ is now a precise means for expressing teridentity as a relation. This means can be used for the corresponding EGs as well, i.e., the Peirce graphs of Fig. 16.1 are equivalent to the Peirce graphs of Fig. 16.3. In these graphs, all ligatures are single object ligatures, thus they are probably easier to understand.

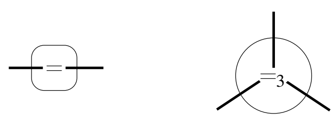


Fig. 16.3. Separating the ligatures of the graphs of Fig. 16.2

It should be noted that a separation of a non-single-object ligature into several single-object ligatures is not uniquely determined. In order to see this, consider the graph of Fig. 16.4.

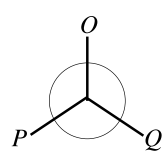


Fig. 16.4. A further Peirce graph with a non-single-object ligature

The ligature of the graph of Fig. 16.4 can be separated in different ways. Five possible separations are provided in Fig. 16.5.

These different separations yield the following different readings of the graph of Fig. 16.4:

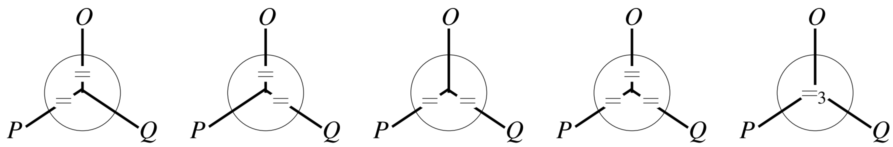


Fig. 16.5. Possible separations of the ligature of the graph of Fig. 16.4

1. There are objects O, P, Q such that Q is not (simultaneously) identical to O and P.
2. There are objects O, P, Q such that P is not (simultaneously) identical to O and Q.
3. There are objects O, P, Q such that O is not (simultaneously) identical to P and Q.
4. There are objects O, P, Q such that there is no forth object which is (simultaneously) identical to O, P and Q.
5. There are objects O, P, Q which are not all identical to each other.

We see that different separations of a non-single-object ligature yield different, semantically equivalent readings of the graph.[3]

Let us consider a more complex example, provided in Fig. 16.6:

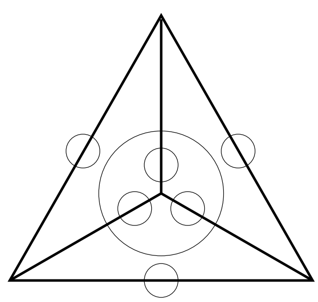


Fig. 16.6. A Peirce graph with the meaning 'there are exactly three things'

The ligature in this graph traverses six cuts completely, thus it is to a large extent a non-single-object ligature. If we separate this ligature in each cut it traverses, we obtain the graph of Fig. 16.7.

[3] So-called 'multiple readings' of a graph are extensively discussed by Shin in [Shi02a], where she argues that this is one of the main features of diagrams humans benefit from.

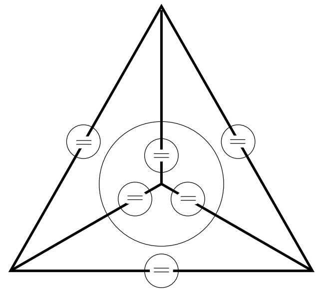


Fig. 16.7. Separating the ligature of the graph of Fig. 16.6

Now we see that the graph of Fig. 16.6 can be read as follows: There are three pairwise distinct things, but there does not exists a fourth object which is distinct to all of these three things. In short: There are exactly tree things.

The examples above show how Lem. 16.9 and the transformation rules 'separating a ligature at the vertex v' and 'joining ligatures at the edge f' can be used to read and understand complex ligatures in an EGI or in a Peirce graph by separating them into several single-object ligatures.

Finally, with the results of this chapter, it is possible to link this treatise, esp. the formalization of EGIs, to a different kind of formalizations of diagrammatic logic systems, which has carried out by some authors. We introduced new relation symbols $\doteq_n$ which were used to separate ligatures. It is important to note that in transformation rule 'separating a ligature at the vertex v', a vertex is, roughly speaking, replaced by an edge. If we had added a relation symbol $\doteq_1$ as well, we could even replace a vertex which is incident only with one identity edge by an edge. This gives raise to the following observation: It is possible to provide a different formalization for EGIs which switches the role of edges and vertices. That is, it is possible to provide a technical definition for EGIs where vertices stand for the relations in a Peirce graph, and the edges model only the heavily drawn lines of a Peirce graph. As it has already been mentioned in 11, approaches of this kind have been undertaken by Pollandt in [Pol02] or Hereth Correia and Pöschel in [HCP04, HCP06], where Peirce's relation graphs are investigated, or by Chein and Mugnier for conceptual graphs in [CM92, CM95], which are a diagrammatic logic system based on Peirce's beta graphs (but which do not include a means to express negation).

17

Soundness of the Calculus

Like in the proof for the soundness of the calculus in the Alpha part of this treatise, the main concept for this proof is the concept of an isomorphism except a context (see Defs. 7.11 and 12.12).

Lemma 17.1 (Erasure and Insertion are Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two EGIs, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by applying the rule 'erasure' or 'insertion', then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: We only show the soundness of the erasure-rule; the proof for the insertion-rule is done analogously.

Let $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0)$ be the subgraph which is erased. $\mathfrak{G}_0$ is erased from the area of the positive context $c := \top_0$. Obviously, $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for the context c by the (trivial) identity mapping. Let ref be a partial valuation for c such that $\mathcal{M} \models_{endo} \mathfrak{G}[c, ref]$. It is easy to see that we have $\mathcal{M} \models_{endo} \mathfrak{G}'[f(c), ref]$. So the property P of Lemma 13.7 holds for c, hence Lemma 13.7 can be applied. This yields the proposition. $\Box$

Next, we show the soundness of the double-cut rule.

Lemma 17.2 (Double Cut is Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two EGIs, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by applying the rule 'double cut', then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}$ and $\mathfrak{G}'$ be two EGIs such that $\mathfrak{G}$ is derived from $\mathfrak{G}'$ by erasing two cuts c_1, c_2 with $area(c_1) = \{c_2\}$. We set $c := ctx(c_1)$. We want to apply Lemma 13.8 and therefore have to show that property P of Lemma 13.8 is valid for $c \in \mathfrak{G}$ and $c \in \mathfrak{G}'$. We have

We have

$$area'(c) = (area(c) \cup area(c_2)) \backslash \{c_1\} \qquad (*)$$

Let ref be a partial valuation for c. With $(*)$ we get

$\mathcal{M} \models \mathfrak{G}[c, ref]$

$\overset{\text{Def.}\models}{\Leftrightarrow}$ ref can be extended to an extended partial valuation $\overline{ref}$ for c such that $\overline{ref}$ fulfills all edge- and cut-conditions of $area(c)$

$\Leftrightarrow$ ref can be extended to an extended partial valuation $\overline{ref}$ for c such that $\overline{ref}$ fulfills all edge- and cut-conditions of $area(c)\backslash\{c_1\}$ and $\mathcal{M} \not\models \mathfrak{G}[c_1, \overline{ref}]$

$\Leftrightarrow$ ref can be extended to an extended partial valuation $\overline{ref}$ for c such that $\overline{ref}$ fulfills all edge- and cut-conditions of $area(c)\backslash\{c_1\}$ and $\mathcal{M} \models \mathfrak{G}[c_2, \overline{ref}]$

$\Leftrightarrow$ ref can be extended to an extended partial valuation $\overline{ref}$ for c such that $\overline{ref}$ fulfills all edge- and cut-conditions of $area(c)\backslash\{c_1\}$ and $\overline{ref}$ can be extended to an extended partial valuation $\overline{\overline{ref}}$ for c_2 such that $\overline{\overline{ref}}$ fulfills all edge- and cut-conditions of $area(c_2)$

$\overset{(*)}{\Leftrightarrow}$ ref can be extended to an extended partial valuation $\overline{\overline{ref}}$ for c such that $\mathcal{M}$ fulfills all edge- and cut-conditions of $area'(c)$

$\overset{\text{Def.}\models}{\Leftrightarrow}$ $\mathcal{M} \models \mathfrak{G}'[c, ref]$

Now Lemma 13.8 yields that we have $\mathcal{M} \models \mathfrak{G} \iff \mathcal{M} \models \mathfrak{G}'$. □

Unfortunately, the proof for the soundness of the rules iteration and its counterpart deiteration is much more complex than in the Alpha-part of this treatise. The main reason is the following: Let $\mathfrak{G}_0$ be a subgraph of a graph $\mathfrak{G}$ which is iterated into a context c. Particularly, new vertices are added to c. When we now evaluate the graph in a model with the endoporeutic method of Peirce, we have to assign objects to these vertices. The assignment of the 'right' objects will depend on which objects we have already assigned to vertices which are placed in the same or higher context as the new vertices. But the new vertices are copies of already existing vertices (their origins of $\mathfrak{G}_0$). Each time when we reach the new vertices while performing the endoporeutic method, we have already assigned objects to their origins. It turns out that we should assign the same objects to the old and new vertices to gain that the old and the new graph become equivalent. This idea is worked out in the proof for the following lemma.

Lemma 17.3 (Iteration and Deiteration are Sound). *If $\mathfrak{G}_a$ and $\mathfrak{G}_b$ are two EGIs, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}_a$ and $\mathfrak{G}_b$ is derived from $\mathfrak{G}_a$ by applying the rule 'iteration' or 'deiteration', then $\mathcal{M} \models \mathfrak{G}_b$.*

Proof: In this proof, we use the notation of the formal definition of the iteration/ deiteration-rule on page 166. Particularly, we assume that $\mathfrak{G}_b$ is derived from $\mathfrak{G}_a$ by iterating the subgraph $\mathfrak{G}_0$ from the cut c_0 into the cut c.

First of all, if we consider the canonical mapping f which is defined by $f(k) = (k, 1)$ for $k \in E_a \cup Cut_a$, then $\mathfrak{G}_a$ and $\mathfrak{G}_b$ are isomorphic up to $c_0 := \top_0$ and $(c_0, 1)$ by f. They are even isomorphic up to c and $(c, 1)$.

The proof of this lemma will be based on Lemma 13.8. We start with some definitions which are needed in the proof.

1. Let $ref_a : V'_a \to U$ be a partial valuation for $\mathfrak{G}_a$. Then ref_a can canonically be transferred to a partial valuation $ref_{a\to b}$ for $\mathfrak{G}_b$ as follows: We set $V'_{a\to b} := \{(v, 1) \in V_b \mid v \in V'_a\} \dot{\cup} \{(v, 2) \in V_b \mid v \in V'_a \cap V_0\}$ and
$$ref_{a\to b} : \begin{cases} V'_{a\to b} \to U \\ (v, i) \mapsto ref_a(v) \end{cases}$$
Please note that this can be considered to be an extension of Def. 13.6. Particularly, if ref_a is a partial valuation for a context d with $d \not\leq c$, we have $ref_{a\to b} = f(ref_a)$.
Analogously, for a partial valuation $ref_b : V'_b \to U$ for $\mathfrak{G}_b$, we set $V'_{b\to a} := \{v \in V_a \mid (v, 1) \in V'_b\}$ and
$$ref_{b\to a} : \begin{cases} V'_{b\to a} \to U \\ \quad v \mapsto ref_b((v, 1)) \end{cases}$$
Again, if ref_b is a partial valuation for a context $(d, 1)$ (with $d \in Cut_a$, hence $(d, 1) \in Cut_b$) with $(d, 1) \not\leq (c, 1)$, we have $ref_{b\to a} = f^{-1}(ref_b)$.
2. To ease the terminology in this proof, we will use the notation of entailment for extended partial valuations for a context c as well, i.e., if c is a context and ref is an extended partial valuation for c, we write
$\mathcal{M} \models_{endo} \mathfrak{G}[c, ref] :\Longleftrightarrow$
   - $ref(e) \in I(\kappa(e))$ for each $e \in E \cap area(c)$ (edge condition))
   - $\mathcal{M} \not\models_{endo} \mathfrak{G}[d', ref]$ for each $d' \in Cut \cap area(d)$ (cut condition))

Now we are prepared to start with the proof.

First we will show the following:

$$\begin{aligned} &\text{If } ref_a \text{ is an extended partial valuation for } c_0 \text{ with} \\ &\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_a, c_0] \text{ , then } \mathcal{M} \models_{endo} \mathfrak{G}_b[ref_{a\to b}, (c_0, 1)] \qquad (17.1) \\ &\text{If } ref_b \text{ is an extended partial valuation for } (c_0, 1) \text{ with} \\ &\mathcal{M} \models_{endo} \mathfrak{G}_b[ref_a, (c_0, 1)] \text{ , then } \mathcal{M} \models_{endo} \mathfrak{G}_a[ref_{b\to a}, c_0] \qquad (17.2) \end{aligned}$$

Assume that Eqn. (17.1) and Eqn. (17.2) hold. We want to apply Lemma 13.8 to $\mathfrak{G}_a$, c_0 and $\mathfrak{G}_b$, $(c_0, 1)$, so let ref be a partial valuation for c_0. If we have $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref, c_0]$, then ref can be extended to a extended partial valuation ref_a with $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_a, c_0]$. Now Eqn. (17.1) yields

$\mathcal{M} \models_{endo} \mathfrak{G}_b[ref_{a\to b}, (c_0, 1)]$. Furthermore we have that $ref_{a\to b}$ is an extension of $f(ref)$. Thus we conclude $\mathcal{M} \models_{endo} \mathfrak{G}_b[f(ref), (c_0, 1)]$. Vice versa, using Eqn. (17.2), we have the following: If we have $\mathcal{M} \models_{endo} \mathfrak{G}_b[f(ref), (c_0, 1)]$, we obtain $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref, c_0]$. Both cases together yield that property P of Lemma 13.8 holds for c_0. Now Lemma 13.8 yields

$$\mathcal{M} \models_{endo} \mathfrak{G} \quad \Longleftrightarrow \quad \mathcal{M} \models_{endo} \mathfrak{G}' \quad ,$$

which shows the soundness of both the iteration- and deiteration-rule.

It remains to show that Eqn. (17.1) and Eqn. (17.2) hold. The proof for this is carried out by distinguish two cases.

First case: $c = c_0$

We start with the proof of Eqn. (17.1), so let ref_a be an extended partial valuation for c_0 with $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_a, c_0]$. In the context $(c_0, 1)$ of $\mathfrak{G}_b$ we have

1. the edges $(e, 1)$ and cuts $(d, 1)$ which correspond to the edges and cuts in the context c_0 of $\mathfrak{G}_a$,
2. the edges $(e, 2)$ and cuts $(d, 2)$ which correspond to the edges and cuts in the context c_0 of $\mathfrak{G}_0$, and
3. further identity edges $e_{v,w} = ((w, 1), (v, 2))$ for vertices $v \in V_0$, $w \in area(c_0)$ with $w \Theta v$.

Let $e = (v_1, \ldots, v_n) \in area_a(c_0)$ be an edge. Due to $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_a, c_0]$, we have $ref_a(e) = (ref_a(v_1), \ldots, ref_a(v_n)) \in I(\kappa_a(e))$. Then we have

$$\begin{aligned} ref_{a\to b}((e,1)) &= (ref_{a\to b}((v_1,1)), \ldots, ref_{a\to b}((v_n,1))) \\ &\overset{\text{Def.}ref_{a\to b}}{=} (ref_a(v_1), \ldots, ref_a(v_n)) \\ &\in I(\kappa_a(e)) \\ &= I(\kappa_b((e,1))) \end{aligned}$$

Thus we see that the edge-conditions for edges $(e, 1)$ which correspond to the edges in the context c_0 of $\mathfrak{G}_a$ are fulfilled. Analogously, it is easy to see that the edge-conditions for edges $(e, 2)$ which correspond to the edges in the context c_0 of $\mathfrak{G}_0$ the cut-conditions for cuts $(d, 1)$ which correspond to the cuts in the context c_0 of $\mathfrak{G}_a$, and the cut-conditions for cuts $(d, 2)$ which correspond to the cuts in the context c_0 of $\mathfrak{G}_0$, are fulfilled as well.

It remains to show that the edge-conditions for the further identity links $e_{v,w}$ are satisfied, too. Let $e_{v,w} = ((w, 1), (v, 2))$ with $v \in V_0$, $w \Theta v$ and $w \in area(c)$ be such a link. Then there are (in $\mathfrak{G}_a$) vertices $v_1, \ldots, v_n$ with $w = v_1$, $v_n = v$, $ctx(v_1) = ctx(v_2) = \ldots ctx(v_n) = c_0$, and for each $1 \leq i \leq n-1$, there is an identity edge e_i from v_i to v_{i+1}. Let $1 \leq i \leq n-1$. As $e_i \in area(c_0)$, we have – again due to $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_a, c_0]$– $ref_a(v_i) = ref_a(v_{i+1})$. Analogously to the argumentation above, we conclude $ref_{a\to b}((v,1)) = ref_{a\to b}((w,1))$. Moreover, we have $ref_{a\to b}((v,1)) = ref_{a\to b}((v,2))$ by definition of $ref_{a\to b}$. Thus we get $ref_{a\to b}((w,1)) = ref_{a\to b}((v,2))$, i.e., the edge-condition for the further identity-link $e_{v,w}$ in $\mathfrak{G}_b$ is satisfied as well, which finally proves Eqn. (17.1).

The proof of Eqn. (17.2) is now obvious: Let ref_b is an extended partial valuation for $(c_0,1)$ in $\mathfrak{G}_b$ with $\mathcal{M} \models_{endo} \mathfrak{G}_b[ref_a,(c_0,1)]$. To each cut d and edge e in $area_a(c_0)$ corresponds the edge $(e,1)$ and $(c,1)$ in $area_b((c_0,1))$, and it can analogously to the proof of Eqn. (17.1) shown that $ref_{b\to a}$ satisfies the edge- and cut-conditions for these edges and cuts in $area_a(c_0)$. Roughly speaking: Compared to $area_b((c_0,1))$, in $area_a(c_0)$ are less edge- and cut-conditions to check. This yields $\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_{b\to a}, c_0]$, hence Eqn. (17.2) is fulfilled.

Second case: $c < c_0$

In contrast to the case $c = c_0$, the edges and cuts in $area_a(c_0)$ correspond bijectively to the edges and cuts in $area_b((c_0,1))$. Particularly, if ref_a is an extended valuation for c_0, we have $(ref_{a\to b})_{\to a} = ref_a$, and vice versa, if ref_a is an extended valuation for $(c_0,1)$, we have $(ref_{b\to a})_{\to b} = ref_b$. Thus, instead of proving Eqn. (17.1) and Eqn. (17.2), it is sufficient to show that for each extended partial valuation ref_{0a} for c_0 we have

$$\mathcal{M} \models_{endo} \mathfrak{G}_a[ref_{0a}, c_0] \iff \mathcal{M} \models_{endo} \mathfrak{G}_b[ref_{0a\to b}, (c_0,1)] \tag{17.3}$$

So let ref_{0a} be an extended partial valuation for c_0. Let $d_0 \in area_a(c_0)$ be the cut with $c \leq d_0$. The following is easy to see: The valuation ref_{0a} satisfies the edge-conditions for all edges $e \in area_a(c_0)$ and the cut-conditions for all cuts $d \in area(a)$ with $d \neq d_0$ if and only if $ref_{0a\to b}$ satisfies the edge-conditions for all (corresponding) edges $(e,1) \in area_b((c_0,1))$ and the cut-conditions for all (corresponding) cuts $(d,1) \in area(a)$ with $(d,1) \neq (d_0,1)$. It remains to show that ref_{0a} satisfies the cut-condition for d_0 if and only if $ref_{0a\to b}$ satisfies the cut-condition for $(d_0,1)$. Moreover, we can assume that

$$\begin{array}{l} ref_{0a} \text{ satisfies the edge-conditions for all edges } e \in area_a(c_0) \\ \text{ and the cut-conditions for all cuts } d \in area(a) \text{ with } d \neq d_0 \end{array} \tag{17.4}$$

(otherwise $\mathcal{M} \not\models_{endo} \mathfrak{G}_a[ref_{0a}, c_0]$ and $\mathcal{M} \not\models_{endo} \mathfrak{G}_b[ref_{0a\to b}, (c_0,1)]$, hence, Eqn.17.3 is satisfied in this case).

We set $D := \{d \in Cut_a \cup \{\top_a\} \mid d \leq d_0 \text{ and } d \not< c\}$. In Fig. 17.1 we have sketched one possible situation.

Let $Q(d)$ be the following property for contexts $d \in D$:

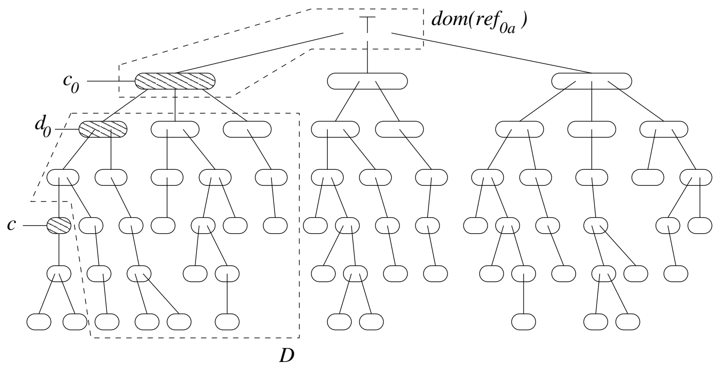


Fig. 17.1. The situation when a subgraph is iterated from c_0 into c

1. If $ref_a : V'_a \to U$ with $ref_a \supseteq ref_{0a}$ is an extended partial valuation for d, then
   a) there is a context d' with $d < d' \leq c_0$ and an edge $e \in area_a(d')$ such that $ref_a(e) \notin I(\kappa_a(e))$, i.e., the edge-condition is not satisfied for e, or
   b) we have
   $$\mathcal{M} \models_{endo} \mathfrak{G}_a[d, ref_a] \Longrightarrow \mathcal{M} \models_{endo} \mathfrak{G}_b[(d,1), ref_{a\to b}]$$
2. If $ref_b : V'_b \to U$ with $ref_b \supseteq ref_{0a\to b}$ is an extended partial valuation for $(d,1)$, then
   $$\mathcal{M} \models_{endo} \mathfrak{G}_b[(d,1), ref_b] \Longrightarrow \mathcal{M} \models_{endo} \mathfrak{G}_a[d, ref_{b\to a}]$$

Assume that we have already shown that $Q(d)$ is satisfied for each cut $d \in area(c)$. Particularly, we have $Q(d_0)$. Note that, due to Eqn. 17.4, the condition $Q(d_0)$1.a) cannot be fulfilled. If ref_{0a} satisfies the cut-condition for d_0, then ref_{0a} cannot be extended to an extended partial valuation ref_a for d_0 with $\mathcal{M} \models_{endo} \mathfrak{G}_a[d_0, ref_a]$. The contraposition of $Q(d_0)$.2 yields that $ref_{0a\to b}$ cannot be extended to an extended partial valuation ref_b for $(d_0,1)$ with $\mathcal{M} \models_{endo} \mathfrak{G}_b[(d_0,1), ref_b]$, i.e., $ref_{0a\to b}$ satisfies the cut-condition for $(d_0,1)$. Vice versa, using the contraposition of $Q(d_0)$.1, we obtain that, if $ref_{0a\to b}$ satisfies the cut-condition for $(d_0,1)$, then ref_{0a} satisfies the cut-condition for d_0. Thus, we would be done. Hence, it is now sufficient to show that $Q(d)$ holds for each $d \in D$. This will be done in the remaining part of the proof.

D is a forest such that for each $d \in Cut_c$ with $d \neq c$, then all cuts $e \in area(d)$ are elements of Cut_c as well. Thus we can, similar to the proof of Lemma 13.7, carry out the proof by induction over D. But now, we have to show separately that $Q(c)$ holds. So, in order to prove $Q(d)$, we distinguish between $d = c$ and $d \neq c$.

- $d = c$

  We start with the proof of 1.

  Let $ref_a : V'_a \to U$ be an extended partial valuation for c with $ref_a \supseteq ref_{0a}$. We suppose that condition a) is not satisfied (otherwise we are done), thus we have $\mathcal{M} \models_{endo} \mathfrak{G}_a[c, ref_a]$.

  First we show:

  $$\text{If } w_1 \in area(c_0),\ w_2 \in area(c) \text{ are two vertices with } w_1 \Theta w_2\,, \\ \text{then } ref_a(w_1) = ref_a(w_2)\,. \tag{17.5}$$

  Let $w_1 \in area(c_0)$ and $w_2 \in area(c)$ be two vertices with $w_1 \Theta w_2$. Then there are vertices $v_1, \ldots, v_n$ with $w_1 = v_1$, $v_n = w_2$, $c = ctx(v_1) \geq ctx(v_2) \geq \ldots \geq ctx(v_n) = c_0$, and for each $1 \leq i \leq n-1$, there is an identity edge e_i from v_i to v_{i+1}. Let $1 \leq i \leq n-1$. We have $c_0 \geq ctx(v_i) \geq ctx(e_i) = ctx(v_{i+1}) \geq c$. For $ctx(e) > c$, then, as condition a) is not satisfied, we conclude $ref_a(v_i) = ref_a(v_{i+1})$. For $ctx(e) = c$, then, as ref_{0a} satisfies all edge-condition in c_0, we conclude again $ref_a(v_i) = ref_a(v_{i+1})$. So we have $ref_a(w_1) = ref_a(v_1) = ref_a(v_2) = \ldots = ref_a(v_n) = ref_a(w_2)$, which proves Eqn. (17.5).

  We have to show $\mathcal{M} \models_{endo} \mathfrak{G}_b[(c,1), ref_{a\to b}]$, i.e., we have to check the edge- and vertex-conditions in $area_b((c,1))$. In $area_b((c,1))$ we have:

  1. Edges $(e,1)$ and cuts $(d,1)$ which correspond to the edges and cuts in the context c of $\mathfrak{G}_a$. Due to $\mathcal{M} \models_{endo} \mathfrak{G}_a[c, ref_a]$, the edge- and cut-conditions for these edges and cuts are fulfilled.
  2. Edges $(e,2)$ and cuts $(d,2)$ which correspond to the edges and cuts in the context c_0 of $\mathfrak{G}_0$. Due to Eqn.17.4. The edge- and cut-conditions for these edges and cuts are fulfilled.
  3. Further identity edges $e_{v,w} = ((w,1),(v,2))$ with $v \in V_0$, $w\Theta v$ and $w \in area(c_0)$. Due to Eqn. (17.5), the edge-conditions for these edges are fulfilled.

  We conclude $\mathcal{M} \models_{endo} \mathfrak{G}_b[(d,1), ref_{a\to b}]$, hence we have shown that the property $Q(c)$.1 holds.

  Similar to the case $c = c_0$, it is easy to see that the property $Q(c)$.2 holds as well (again, we have to check less edge- and cut-conditions).

- $d \in D$, $d \neq c$.

  We start with the proof of $Q(d)$.1.

  Let $ref_a \supseteq ref_{0a}$ be an extended partial valuation for d in $\mathfrak{G}_a$ with $\mathcal{M} \models_{endo} \mathfrak{G}_a[d, ref_a]$. Then $ref_{a\to b}$ is an extended partial valuation for

$(d,1)$ in $\mathfrak{G}_a$. The edges $e \in d$ correspond bijectively to the edges $(e,1)$ in $(d,1)$, thus it is easy to see that the edge-conditions in $area_b((d,1))$ are satisfied by $ref_{a\to b}$. It remains to show that the cut-conditions in $area_b((d,1))$ hold as well. Each cut in $area_b((d,1))$ has the form $(d',1)$ with $d' \in area_a(d)$. So let $(d',1) \in area_b((d,1))$ be a cut.

We have to show $\mathcal{M} \not\models_{endo} \mathfrak{G}[ref_{a\to b},(d',1)]$. Assume that we have $\mathcal{M} \models_{endo} \mathfrak{G}[ref_{a\to b},(d',1)]$. Then there is an extended partial valuation $\overline{ref_{a\to b}}$ for $(d',1)$ with $\overline{ref_{a\to b}} \supseteq ref_{a\to b}$ and $\mathcal{M} \models \mathfrak{G}_b[\overline{ref_b},(d',1)]$. Obviously, $\overline{ref_a} := \overline{ref_{a\to b\to a}}$ is an extended partial valuation for d' with $\overline{ref_a} \supseteq ref_a$, and the induction hypothesis $Q(d')$, 2. yields $\mathcal{M} \models_{endo} \mathfrak{G}_a[d',\overline{ref_a}]$, hence $\mathcal{M} \models_{endo} \mathfrak{G}_a[d',ref_a]$, which contradicts $\mathcal{M} \models_{endo} \mathfrak{G}_a[d,ref_a]$. Thus 1. is shown.

The proof of of $Q(d)$.2. is done analogously.

As we shown that $Q(d)$ holds for each $d \in area(c)$, we are done. □

Like in the Alpha part, we obtain from the preceding lemmata (which show that each rule of the calculus is sound) and Thm. 13.10 (which shows that the transformation-rules are sound) the soundness of the Beta-calculus.

Theorem 17.4 (Soundness of the Beta-Calculus). *For each set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of EGIs over $\mathcal{A}$ we have*

$$\mathfrak{H} \vdash \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \models \mathfrak{G}$$

Proof: Analogous to the proof of Thm. 9.5.

18

First Order Logic

In the next chapters, the correspondence between Beta graphs and the symbolic form of first order predicate logic is elaborated. This will be done by providing translations between Beta graphs and formulas of $\mathcal{FO}$. There are many calculi for $\mathcal{FO}$ from which it is well-known that they are sound and complete. Proving that a logic system with the expressiveness of first order logic is complete is somewhat extensive. For this reason, in contrast to Alpha, the completeness of Beta will not be shown directly. Instead of this, using the translations between graphs and formulas, the completeness of $\mathcal{FO}$ can be transferred to the system of existential graphs.

Showing the correspondence between Beta and $\mathcal{FO}$ and using the completeness of $\mathcal{FO}$ are two good reasons to employ translations between Beta and $\mathcal{FO}$. Nonetheless, there are many different styles of $\mathcal{FO}$. In this chapter, the the style of $\mathcal{FO}$ we will use in the rest of this treatise is presented.

18.1 Syntax

We start with the definition of the well-formed formulas of $\mathcal{FO}$.

Definition 18.1 (Formulas). *Let $Var := \{x_1, x_2, x_3, \ldots\}$ be a countably infinite set of signs.*[1] *The elements of Var are called* VARIABLES.

The formulas of $\mathcal{FO}$ over $\mathcal{A}$ and the set $FV(f)$ of free variables of a formula f, and the set $Subform(f)$ of subformulas of f are inductively defined as follows:[2]

[1] We will use the letters x, y, u, v for variables, too. Furthermore we will use the Greek letter 'α' to denote variables, i.e. 'α' is a metavariable.

[2] Note that we do not have object names or function names, therefore we do not need to define *terms*. But we will come back in Chpt. 23, where EGIs are augmented with object and function names, to this issue.

1. *If $R \in \mathcal{R}$ is a relation name with arity n and $\alpha_1, \ldots, \alpha_n$ are variables, then $f := \underline{R}\underline{(}\alpha_1\underline{,}\ldots\underline{,}\alpha_n\underline{)}$ is a formula.*[3]
   - $FV(f) := FV(\alpha_1) \cup \ldots \cup FV(\alpha_n)$, $Subform(f) := \{f\}$.
2. *If f' is a formula, then $f := \underline{\neg} f'$ is a formula.*
   - $FV(f) := FV(f')$, $Subform(f) := Subform(f') \cup \{f\}$.
3. *If f_1 and f_2 are formulas, then $f := \underline{(}f_1 \underline{\wedge} f_2\underline{)}$ is a formula.*
   - $FV(f) = FV(f_1) \cup FV(f_2)$, $Subform(f) := Subform(f_1) \cup Subform(f_2) \cup \{f\}$.
4. *If f' is a formula and α is a variable, then $f := \underline{\exists}\alpha\underline{.}f'$ is a formula.*
   - $FV(f) := FV(f') \backslash \{\alpha\}$, $Subform(f) := Subform(f') \cup \{f\}$.

If f is a formula with $FV(f) = \emptyset$, then f is said to be CLOSED. *A closed formula is also called a* SENTENCE.

For some dyadic relation names R, esp. for '$R = \doteq$', we will use the infix-notation instead of the prefix-notation, i.e., we will write $\alpha_1 R \alpha_2$ instead of $R(\alpha_1, \alpha_2)$ (in particular, we will write $\alpha_1 \doteq \alpha_2$ or $\alpha_1 = \alpha_2$ instead of $\doteq (\alpha_1, \alpha_2)$).

Keep in mind that we use an identity-relation in the formulas of $\mathcal{FO}$ as well as in the metalanguage. We distinguish these two levels of identity by using the symbol '$\doteq$' for the identity on the syntactical level (in formulas as well as in graphs) and by using symbol '$=$' to denote the identity on the meta-level. In some cases we try to ease the reading by using different spaces around '$=$' and by using the symbol $\doteq$. For example, in '$f \;=\; x_3 \doteq x_5$', the first '$=$' is a metalevel sign, and the second '$\doteq$' is a sign in $\mathcal{FO}$. So the string '$f \;=\; x_3 \doteq x_5$' means that f is the formula '$x_3 \doteq x_5$'.

The remaining junctors (i.e., $\vee$, $\rightarrow$ and $\leftrightarrow$) and the remaining quantifier $\forall$ are defined as usual: We set

- $f_1 \vee f_2 := \neg(\neg f_1 \wedge \neg f_2)$,
- $f_1 \rightarrow f_2 := \neg(f_1 \wedge \neg f_2)$,
- $f_1 \leftrightarrow f_2 := \neg(f_1 \wedge \neg f_2) \wedge \neg(f_2 \wedge \neg f_1)$, and
- $\forall \alpha . f := \neg \exists \alpha . \neg f$.

[3] In this definition, we have underlined the signs which have to be understood literally. For example, when we write $f := \underline{R}\underline{(}\alpha_1\underline{,}\ldots\underline{,}\alpha_n\underline{)}$, this means that the formula f is defined to be the following sequence of signs: We start with the relation name which is denoted by R. Please note that 'R' *is* not a relation name, but it is a metavariable which *stands for* a relation name. After R, we proceed with the sign '('. After that, we write down the variable which is denoted by α_1 (thus, α_1 is a metavariable, too). We proceed with the sign ','. After that, we write down the variable which is denoted by α_2, proceed with the sign ',', and so on until we write down the variable is denoted by α_n. Finally, we write down the sign ')'.

This shall be read as an abbreviation: e.g. when we write $f_1 \to f_2$, we can replace this by $\neg(f_1 \wedge \neg f_2)$ to get a formula in our language.

Brackets are omitted or added to improve the readability of formulas. To avoid an overload of brackets, we agree that formulas which are composed with the dyadic junctor $\to$ are bracket from the right, e.g. $f_1 \to f_2 \to f_3$ has to read as $(f_1 \to (f_2 \to f_3))$. Furthermore we agree that quantifiers bind more strongly than binary junctors.

It will turn out later (especially in Def. 18.4) that it is important to distinguish between subformulas and SUBFORMULA OCCURRENCES of a formula f. For example, the formula $f := \top(x) \wedge \top(x)$ has only two subformulas, namely $\top(x)$ and f itself. But the subformula $\top(x)$ occurs twice in f, hence we have three subformula occurrences in f (two times $\top(x)$ and f itself).

Defining subformula occurrences is a rather technical problem, thus, we do not provide a definition for them, but we want to point out that it can be done (for a further discussion we refer to [vD96]). Of course, this is the same for variables: A variable t can appear several times in a formula f, and all these appearances are called OCCURRENCES OF t IN f.

Next we define substitutions.

Definition 18.2 (Substitutions). *Let $\alpha \in Var$ be a variable and let $x_i \in Var$ be a variable. Furthermore, let $\alpha_1, \alpha_2, \ldots$ denote variables and $f, f', f_1, f_2, \ldots$ denote formulas. We define the substitutions $f[\alpha/x_i]$ inductively as follows:*

1. *For a variable $x_j \in Var$, let $x_j[\alpha/x_i] := x_j$ for $j \neq i$, and $x_i[\alpha/x_i] := \alpha$.*
2. *If $f := R(\alpha_1, \ldots, \alpha_n)$, then $f[\alpha/x_i] := R(\alpha_1[\alpha/x_i], \ldots, \alpha_n[\alpha/x_i])$.*
3. *If $f := \neg f'$, then $f[\alpha/x_i] := \neg f'[\alpha/x_i]$.*
4. *If $f := f_1 \wedge f_2$, then $f[\alpha/x_i] := f_1[\alpha/x_i] \wedge f_2[\alpha/x_i]$.*
5. *If $f := \exists x_j.f'$, $j \neq i$, then $f[\alpha/x_i] := \exists x_j.f'[\alpha/x_i]$.*
6. *If $f := \exists x_i.f'$, then $f[\alpha/x_i] := \exists x_i.f'$.*

We say that $f[\alpha/x_i]$ is obtained from f by SUBSTITUTING α FOR x_i IN f.

A main difference between $\mathcal{FO}$ and $\mathcal{EGI}$ are the syntactical elements which are used to range over objects. In $\mathcal{EGI}$, only one sign, namely the LoI, is used for this purpose. In $\mathcal{FO}$ we have a whole set Var of variables instead. All variables are tantamount. For this reason the next well-known definition is needed.

Definition 18.3 (α-Conversion of Formulas). *Let f be a formula and let $\exists\alpha.h$ be a subformula of f. Let β be a (so called* fresh*) variable (i.e. we have $\beta \notin Var(f)$). Let f' be the formula that we get when we replace the subformula*

$\exists\alpha.h$ by $\exists\beta.h[\beta/\alpha]$. Then we say that we get f' from f by RENAMING A BOUND VARIABLE *(this is in literature often called α-*CONVERSION *of a formula).*

Example: Consider the formula

$$R(x) \wedge \exists x.S(x) \wedge \exists x.(R(x) \wedge \exists x.T(x))$$

If we replaced the first bound occurrence of x – i.e., we consider the subformula $\exists x.S(x)$ – by the variable y, we get the formula

$$R(x) \wedge \exists y.S(y) \wedge \exists x.(R(x) \wedge \exists x.T(x))$$

In this formula, consider the subformula $\exists x.(R(x) \wedge \exists x.T(x))$. We replace x by u and get

$$R(x) \wedge \exists y.S(y) \wedge \exists u.(R(u) \wedge \exists x.T(x))$$

Finally, we replace the remaining bound variables x in $\exists x.T(x)$ by v and get

$$R(x) \wedge \exists y.S(y) \wedge \exists u.(R(u) \wedge \exists v.T(v))$$

In formulas of $\mathcal{FO}$, the set of all occurences of the negation sign '$\neg$' can be treated like the set of all cuts in an existential graph. This motivates the next definition:

Definition 18.4 (Structure of Formulas). *Let $f \in \mathcal{FO}$ be a formula. We set*

$$Neg_f := \{g \in \mathcal{FO} \mid \neg g \text{ is a subformula occurrence of } f\}$$

Furthermore we set $\top_f := f$. The set $Neg_f \cup \{\top_f\}$ is ordered by

$$g \leq h :\Longleftrightarrow g \text{ is a subformula occurrence of } h$$

The set Neg_f of a formula f is the counterpart of the set Cut of a graph $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, and $\mathrm{Neg}_f \cup \{\top_f\}$ is the counterpart of $Cut \cup \{\top\}$. To give an example, look at the following formula:

$$f := \exists x.(CAT(x) \wedge \neg slim(x) \wedge \neg(\exists y.LASAGNE(y) \wedge see(x,y) \wedge \neg eat(x,y)))$$

The ordered set $(\mathrm{Neg}_f \cup \{\top_f\}, \leq)$ can be sketched as follows:

$$\begin{tikzcd}
 & \exists x.CAT(x) \wedge \neg slim(x) \wedge \neg(\exists y.LASAGNE(y) \wedge see(x,y) \wedge \neg eat(x,y)) \arrow[dl, dash] \arrow[d, dash] \\
slim(x) & \exists y.LASAGNE(y) \wedge see(x,y) \wedge \neg eat(x,y) \arrow[d, dash] \\
 & eat(x,y)
\end{tikzcd}$$

Another example is

$$g := \neg(\neg\top(x) \wedge \neg\top(x))$$

This example shows that is is important to discriminate between subformulas and subformula occurrences. The ordered set $(\mathrm{Neg}_g \cup \{\top_g\}, \leq)$ can be sketched as follows:

$$\begin{tikzcd}[column sep=small]
 & \neg(\neg\top(x) \wedge \neg\top(x)) \arrow[d, dash] & \\
 & \neg\top(x) \wedge \neg\top(x) \arrow[dl, dash] \arrow[dr, dash] & \\
\top(x) & & \top(x)
\end{tikzcd}$$

18.2 Semantics

The semantics of $\mathcal{FO}$ relies on the well-known relational structures which have already been presented in Def. 13.1. We have to define how formulas are evaluated in relational structures.

Definition 18.5 (Evaluation of Formulas in Structures). *Let $f \in \mathcal{FO}$ be a formula and let $\mathcal{M} = (U, I)$ be a relational structure. A* VALUATION *val is a mapping $val : Var \to U$ which assigns to each variable an element of the universe U. Inductively over the structure of formulas, we define $\mathcal{M} \models_{val} f$ as follows:*

1. *If $R \in \mathcal{R}_n$ and if $\alpha_1, \ldots, \alpha_n$ are variables, then let $\mathcal{M} \models_{val} R(\alpha_1, \ldots, \alpha_n)$ iff $(val(\alpha_1), \ldots, val(\alpha_n)) \in I(R)$.*
2. *If f' is a formula and $f := \neg f'$, then we set $\mathcal{M} \models_{val} f$ iff $\mathcal{M} \not\models_{val} f'$.*
3. *If f_1 and f_2 are formulas and $f := f_1 \wedge f_2$, then we set $\mathcal{M} \models_{val} f$ iff $\mathcal{M} \models_{val} f_1$ and $\mathcal{M} \models_{val} f_2$.*
4. *If f' is a formula, x_i is a variable and $f := \exists x_i.f'$, then we set $\mathcal{M} \models_{val} f$ iff there is a valuation val' with $val'(x_j) = val(x_i)$ for each $i \neq j$ and $\mathcal{M} \models_{val'} f'$.*

If f is a formula and $\mathcal{M}$ is a relational structure such that $\mathcal{M} \models_{val} f$ holds for all valuations val, we write $\mathcal{M} \models f$. If F is a set of formulas and f is a formula such that each relational structure $\mathcal{M}$ with $\mathcal{M} \models g$ for all $g \in F$ satisfies $\mathcal{M} \models f$, we write $F \models f$. We abbreviate $\{g\} \models f$ by $g \models f$. Two formulas f, g with $f \models g$ and $g \models f$ are called semantically equivalent.

18.3 Calculus

In literature we find a huge amount of calculi for first order logic. For our purpose it makes no difference which calculus we use. We decided to choose a (so-called Hilbert-style) calculus. Carrying out proofs in this calculus is arduous (for this reason, usually so called sequent calculi or natural deduction calculi are preferred), but we will not use this calculus for proofs in this treatise. Vice versa: We will have to carry over the rules of the calculus to proofs for EGIs, and this turns out to be somewhat easy for this calculus.

The chosen calculus is is a calculus for the fragment of $\mathcal{FO}$ which is based on the junctors $\to$ and $\neg$ instead of $\wedge$ and $\neg$. This causes no troubles, as the following argumentation will show:

Remember that we can express the junctor $\to$ by means of the junctors $\wedge$ and $\neg$: We set $f_1 \to f_2 := \neg(f_1 \wedge \neg f_2)$ (and conversely, we can express the junctor $\wedge$ with the junctors $\to$ and $\neg$: We set $f_1 \wedge f_2 := \neg(f_1 \to \neg f_2)$). If we denote the set of formulas which use the symbols $\exists, \neg, \wedge$ by $\mathcal{FO}_{\exists,\neg,\wedge}$ and if we denote the set of formulas which use the symbols $\exists, \neg, \to$ by $\mathcal{FO}_{\exists,\neg,\to}$, we can translate each formula $f \in \mathcal{FO}_{\exists,\neg,\wedge}$ to a formula $f^* \in \mathcal{FO}_{\exists,\neg,\to}$, and vice versa. Of course we can define canonically the relation $\models$ between relational structures and formulas of $\mathcal{FO}_{\exists,\neg,\to}$ as well. It is easy to show that we have

$$\mathcal{M} \models f \quad \Longleftrightarrow \quad \mathcal{M} \models f^*$$

for all relational structures $\mathcal{M}$ and all formulas $f \in \mathcal{FO}_{\exists,\neg,\wedge}$. So we can immediately carry over results from $\mathcal{FO}_{\exists,\neg,\to}$ to $\mathcal{FO}_{\exists,\neg,\wedge}$ and vice versa. In particular we will argue that the calculus we present is sound and complete in $\mathcal{FO}_{\exists,\neg,\to}$, hence it is sound and complete in $\mathcal{FO}_{\exists,\neg,\to}$.

In Fig. 18.1 we list all axioms and rules for the $\mathcal{FO}$-calculus we use in this treatise. With this rules, we define the relation $\vdash$ as follows:

Definition 18.6 (Proofs). *Let F be a set of formulas and let f be a formula. A sequence $(f_1, \ldots, f_n)$ is called* PROOF FOR f FROM F *or* DERIVATION OF f FROM F*, if $f_n = f$ and for each $i = 1, \ldots, n$, one of the following conditions holds:*

- *$f_i \in F$, or*
- *there are f_j, f_k with $j, k < i$ and f_i is derived from f_j, f_k using* MP*, or*
- *there is f_j with $j < i$ and f_i is derived from f_j using* Ex4*, or*
- *f_i is one of the remaining axioms.*

If there is a derivation of f from F, we write $F \vdash f$. We write $g \vdash f$ instead of $\{g\} \vdash f$ for formulas g. Two formulas g with $f \vdash g$ and $g \vdash f$ are called PROVABLY EQUIVALENT.

The rules MP, P1, P2, P3 form a sound and complete calculus for propositional logic (see [TW94]). Formulas f which can be derived from $\emptyset$ only with these rules are called TAUTOLOGOUS.

The rules Ex1, Ex2, Ex3, Id1, Id2, Id3, Cong form the step from propositional logic to first order predicate logic. The rules Ex1-Ex3 are common rules which are needed when the existential quantifier is introduced (see for example [Sho67]). The rules Id1, Id2, Id3, Cong are well-known rules which capture the fact that the sign $\doteq$ is interpreted in any model by the (extensional) identity. It is well known that these axioms and rules are sound and complete, i.e., we have the following theorem:

Theorem 18.7 (Soundness and Completeness of $\mathcal{FO}$). *Each set $F \cup \{f\}$ of EGIs over of formulas over $\mathcal{A}$ satisfies*

$$F \vdash f \quad \Longleftrightarrow \quad F \models f$$

Let $\alpha, \alpha_1, \alpha_2, \alpha_3, \ldots$ be variables and let f, g, h be formulas. Then we have the following axioms and rules in $\mathcal{FO}$:

MP: $f, f \to g \quad \vdash \quad g$

P1: $\vdash\ f \to (g \to f)$

P2: $\vdash\ (\neg f \to \neg g) \to (g \to f)$

P3: $\vdash\ (f \to (g \to h)) \to ((f \to g) \to (f \to h))$

Ex1: $\vdash\ f \to \exists \alpha . f$

Ex2: $\vdash\ f \to \exists \alpha_1 . f[\alpha_1/\alpha_2] \qquad$ if $\alpha_1 \notin \mathrm{FV}(f)$

Ex3: $f \to g \vdash \exists \alpha . f \to g \qquad$ if $\alpha \notin \mathrm{FV}(g)$

Id1: $\vdash\ \alpha_0 = \alpha_0$

Id2: $\vdash\ \alpha_0 = \alpha_1 \to \alpha_1 = \alpha_0$

Id3: $\vdash\ \alpha_0 = \alpha_1 \to \alpha_1 = \alpha_2 \to \alpha_0 = \alpha_2$

Cong: $\vdash\ \alpha_0 = \alpha_n \to \alpha_1 = \alpha_{n+1} \to \ldots \to \alpha_{n-1} = \alpha_{2n-1}$
$\to R(\alpha_0, \ldots, \alpha_{n-1}) \to R(\alpha_n, \ldots, \alpha_{2n})$

Fig. 18.1. The $\mathcal{FO}$-calculus we use in this treatise.

19

Syntactical Translations

At the beginning of Chpt. 13, it has already been mentioned that in literature, often a semantics for existential graphs is established as a mapping -let us call it Φ– of existential graphs to $\mathcal{FO}$-formulas. In fact, as argued in this chapter, this use of the term 'semantics' is in my view not appropriate. Instead of this, Φ should considered as a translation instead.

In this chapter, I will provide a mathematical definition for Φ which maps EGIs to $\mathcal{FO}$-formulas, and, vice versa, a mathematical definition for a mapping Ψ which maps $\mathcal{FO}$-formulas to EGIs. In a broader semantical understanding, it will turn out that Φ and Ψ are mutually inverse to each other.

19.1 Definition of Φ and Ψ

In this section definitions for mappings Ψ and Φ and which can be understood as translations between the two logical systems of first order predicate logic and EGIs, i.e., $\mathcal{FO}$ and $\mathcal{EGI}$, are provided. According to the structures of formulas resp. graphs, these mappings are defined recursively.

Before we start with the definitions for Ψ and Φ, let us shortly discuss some problems caused by structural differences between $\mathcal{FO}$ and the system of EGIs. To go into details:

1. In EGIs, we have no syntactical devices which correspond to the free variables of $\mathcal{FO}$, but (as Ψ is defined recursively), we need translations from formulas with free variables to EGIs. For this reason, in this section, we consider existential graph where the vertices may be labelled with variables. But their introduction should be understood as a mere technical

trick.[1] It will turn out that existential graph without variables will be translated to formulas without free variables, and vice versa.

2. In $\mathcal{FO}$, we have an infinite set of variables which are used to range over objects. In EGIs, vertices are used for this purpose. Thus a formula $\Phi(\mathfrak{G})$ will be only given up to the names of the variables.
3. In $\mathcal{FO}$, we can syntactically express different orders of formulas in conjunctions. As conjunction is an assossiative and commutative operation, in $\mathcal{FO}$ the calculus allows to change the order of formulas in conjunctions. In EGIs, conjunction is expressed by the juxtaposition of graphs. Thus we have no possibility to express different orderings of graphs in conjunctions. For the mapping Φ this yields the following conclusion: A formula $\Phi(\mathfrak{G})$ of a existential graph $\mathfrak{G}$ is moreover only given up the order of the subformulas of conjunctions.

Particularly, it cannot be expected that $\Phi \circ \Psi$ is the identity mapping.

We start with the definition of EGIs where the vertices can be labelled with variables.

Definition 19.1 (Existential Graph Instances with Variables). *A structure* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ *is called* EXISTENTIAL GRAPH INSTANCE WITH VARIABLES, *iff* $(V, E, \nu, \top, Cut, area, \kappa)$ *is an EGI and if* $\rho : V \to Var \,\dot{\cup}\, \{*\}$ *is a mapping. The sign '$*$' is called* GENERIC MARKER. *Furthermore, we set* $V^* := \{v \in V \mid \rho(v) = *\}$ *and* $V^{Var} := \{v \in V \mid \rho(v) \in Var\}$. *The vertices* $v \in V^*$ *are called* GENERIC VERTICES, *and the vertices* $v \in V^{Var}$ *are called* VARIABLE VERTICES.

EGIs with variables are a mere helper construction for the definitions of Φ and Ψ (nonetheless, in Chpt. 24, we will come back to labelled vertices in order to add object names to EGIs). The relationship between EGIs without and EGIs with variables correspond to the relationship between formulas (with free variables) and sentences (formulas without free variables) in $\mathcal{FO}$. In their graphical notation, existential graph instance with variables are drawn like EGIs, and above the vertex-spots of variable vertices, we write the corresponding variables.

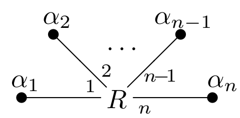
Now we are prepared to provide the definitions of Ψ and Φ. We start with the mapping Φ which maps an EGI to a formula of $\mathcal{FO}$.

Definition of Ψ:

We define Ψ inductively over the composition of formulas (see Def. 18.1). To each formula f, we assign an EGI with variables $\Psi(f)$. For each case, we first provide a diagram or an informal description before we state the explicit mathematical definition.

[1] But in Chpt. 24, so called *vertex-based EGIs, which syntatically are defined like EGIs with variables, are investigated.*

- $R(\alpha_1, \ldots, \alpha_n)$ for a n-ary relation name R and variables $\alpha_1, \ldots, \alpha_n$:

  $\Psi(R(\alpha_1, \ldots, \alpha_n)) :=$ 

  That is, we set $\Psi(R(\alpha_1, \ldots, \alpha_n)) := \mathfrak{G}$ with

  $$\mathfrak{G} := (\{1, \ldots, n\}, \{0\}, \{(0, (1, \ldots, n))\}, \top, \emptyset, \{(\top, \{0, 1, \ldots, n\})\},$$
  $$\{(0, R), (1, \top), \ldots, (n, \top)\}, \{(1, \alpha_1), \ldots, (n, \alpha_n)\})$$

- $f_1 \wedge f_2$ for two formulas f_1 and f_2: $\quad \Psi(f_1 \wedge f_2) := \Psi(f_1) \quad \Psi(f_2)$

  (i.e., $\Psi(f_1 \wedge f_2)$ is the juxtaposition of $\Psi(f_1)$ and $\Psi(f_2)$).

- $\neg f$ for a formula f: $\quad \Psi(\neg f) := \boxed{\Psi(f)}$

  For $\Psi(f) = (V, E, \nu, \top, Cut, area, \kappa, \rho)$ let $c_0 \notin V \cup E \cup Cut \cup \{\top\}$ be a new cut. Now we set $\Psi(\neg f) := (V, E, \nu, \top, Cut', area', \kappa, \rho)$ with

  $Cut' = Cut \cup \{c_0\}$ and
  $area' = area \backslash \{(\top, area(\top))\} \cup \{(c_0, area(\top)), (\top, c_0)\}$

- $\exists \alpha.f$ for a formula f and a variable α (this case is called EXISTENTIAL STEP):

  If $\alpha \notin \mathrm{FV}(f)$, we set $\Psi(\exists \alpha.f) := \Psi(f)$.

  For $\alpha \in \mathrm{FV}(f)$, all vertices v with $\rho(v) = \alpha$ are replaced by a single, generic vertex v_0 on the sheet of assertion, i.e., the following steps have to be carried out:

  1. A new vertex v_0 with $\rho(v_0) = \alpha$ is juxtaposed to $\Psi(f)$ (i.e., we draw $\overset{*}{\bullet} \quad \Psi(f) \quad$).
  2. On each hook (e, i) such that a vertex v with $\rho(v) = \alpha$ is attached to it, we replace v by v_0.
  3. All vertices $v \neq v_0$ with $\rho(v) = \alpha$ are erased.
  4. The label of v is changed from α to $*$.

  The mathematically precise procedure is as follows:

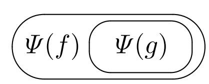
  Let $\Psi(f) := \mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$, and let V_α be the set of all α-vertices, i.e., $V_\alpha := \{w \in V \mid \rho(w) = \alpha\}$. Let v_0 be a fresh vertex. Let $sub : V \to V'$ be the helper function defined by $sub(v) = v_0$, if $\rho(v) = \alpha$, and $sub(v) = v$, if $\rho(v) \neq \alpha$. Now we can define $\Psi(\exists x.f) := (V', E', \nu', \top', Cut', area', \kappa', \rho')$ as follows:

$$\begin{aligned}
V' &:= V \backslash V_\alpha \,\dot\cup\, \{v_0\} \\
E' &:= E \\
\nu' &: \ \nu'(v_1, \ldots, v_n) := \nu(sub(v_1), \ldots, sub(v_n)) \\
\top' &:= \top \\
Cut' &:= Cut \\
area' &: \ area'(c) := area(c) \text{ for } c \neq \top \text{ , and } area'(\top) := area(\top) \,\dot\cup\, \{v_0\} \\
\kappa' &:= \kappa \\
\rho' &:= \rho \,\dot\cup\, \{(v_0, *)\}
\end{aligned}$$

This completes the definition of Ψ. For formulas without free variables, $\Psi(f) := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ is formally an EGI with variables, but we have $\rho(v) = *$ for each $v \in V$. For this reason, $\Psi(f)$ is identified with the EGI (without variables) $\mathfrak{G}$.

In Sect. 12.1 we had shown that the ordered set of contexts $(Cut \cup \{\top\}, \leq)$ is a tree (see Cor. 12.3) and can be considered to be the 'skeleton' of a existential graph with cuts. In Chpt. 18, Def. 18.4 we had defined a corresponding structure for $\mathcal{FO}$-formulas. Now the inductive definition of Ψ yields the following: If f is a formula and $\Psi(f) := \mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ then it is evident that $(\mathrm{Neg}_f, \leq)$ and $(Cut \cup \{\top\}, \leq)$ are isomorphic orders. We denote the canonically given isomorphism by $\Psi_{Neg} : \mathrm{Neg}_f \to Cut \cup \{\top\}$.

As implications are important (e.g. nearly all axioms and rules of the $\mathcal{FO}$-calculus are build up from implications), we want to remark the following: If f and g are sentences, then we have

$\Psi(f \to g) = \Psi(\neg(f \wedge \neg g)) =$ 

Remember that this device of two nested cuts is what Peirce called a *scroll*. Scrolls are the kind how implications are written down in existential graphs.

As we have finished the definition of Ψ, we proceed with the definition of Φ which maps EGIs to $\mathcal{FO}$-formulas. The definition of Φ is nearly straight forward, but we have to take care how empty cuts and isolated vertices are translated: There is no 'canonical' translation of an empty cut or an isolated vertex into a $\mathcal{FO}$-formula. In the Alpha part, we had only to cope with the problems of empty areas of a cut: They had been translated to $\neg(P_1 \wedge \neg P_1)$. For Beta, the approach is slightly different: We first transform a graph into a graph having no isolated vertices or empty cuts. This graph is called a *standardization* of the starting graph, and it can easily be translated to a $\mathcal{FO}$-formula.

Definition 19.2 (Standardization). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ as follows: Insert $\bullet\!\!-\!=\!-\!\!\bullet$ into each empty context c of $\mathfrak{G}$, and replace each isolated vertex $\bullet$ of $\mathfrak{G}$ by $\bullet\!\!-\!=\!-\!\!\bullet$ The resulting* *graph $\mathfrak{G}'$ is called the* STANDARDIZATION *of $\mathfrak{G}$. If $\mathfrak{G}$ contains neither empty cuts, nor isolated vertices, then $\mathfrak{G}$ is said to be in* STANDARD-FORM.

Lemma 19.3 (A Graph is Synt. Equivalent to its Standardization). *Let $\mathfrak{G}$ be an EGI and $\mathfrak{G}'$ its standardization. Then $\mathfrak{G} \vdash \mathfrak{G}'$ and $\mathfrak{G}' \vdash \mathfrak{G}$.*

Proof: Due to the rules 'erasing a vertex' and 'inserting a vertex', isolated vertices may be added to or removed from arbitrary contexts. Moreover, due to the iteration/deiteration-rule, an isolated vertex $\bullet$ can be replaced by $\bullet\!\!- = -\!\!\bullet$, and vice versa. This yield this lemma. $\Box$

Note that, as the calculus is sound, we have that $\mathfrak{G}$ and $\mathfrak{G}'$ are semantically equivalent as well.

Now the formal definition of Φ can be provided.

Definition of $\boldsymbol{\Phi}$.

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. We assume that $\mathfrak{G}$ is in standard-form, i.e., it neither contains empty cuts, nor isolated vertices.

To each vertex $v \in V$ we assign a fresh variable α_v (i.e., for $v_1 \neq v_2$, we have $\alpha_{v_1} \neq \alpha_{v_2}$). Let $\alpha_{empty} \notin \mathrm{FV}(\mathfrak{G}) \cup \{\alpha_v \mid v \in V \text{ and } \rho(v) = *\}$ be a further variable. Now, inductively over the tree $Cut \cup \{\top\}$, we assign to each context $c \in Cut \cup \{\top\}$ a formula $\Phi(\mathfrak{G}, c)$. So let c be a context such that $\Phi(\mathfrak{G}, d)$ is already defined for each cut $d < c$. First, we define a formula f which encodes all edges which are directly enclosed by c. Hence, if c does not directly enclose any edges or vertices, simply set $f := (\exists\alpha_{empty}.\top(\alpha_{empty}))$. Otherwise, let f be the conjunction of the following atomic formulas:[2]

$$\kappa(e)\underline{(}\alpha_{w_1}\underline{,}\dots\underline{,}\alpha_{w_j}\underline{)} \text{ with } e \in E \cap area(c) \text{ and } \nu(e) = (w_1, \dots, w_j).$$

Let $v_1, \dots, v_n$ be the vertices of $\mathfrak{G}$ which are directly enclosed by c, and let $area(c) \cap Cut = \{c_1, \dots, c_l\}$ (by induction, we already assigned formulas to these cuts). If $l = 0$, set $\Phi(\mathfrak{G}, c) := \underline{\exists}\alpha_{v_1}\underline{.}\dots\underline{\exists}\alpha_{v_n}\underline{.}f$, otherwise set

$$\Phi(\mathfrak{G}, c) := \underline{\exists}\alpha_{v_1}\underline{.}\dots\underline{\exists}\alpha_{v_n}\underline{.}\underline{(}f\underline{\triangle}\,\underline{\neg}\Phi(\mathfrak{G}, c_1)\underline{\triangle}\dots\underline{\triangle}\,\underline{\neg}\Phi(\mathfrak{G}, c_l)\underline{)} \quad .$$

Finally we set $\Phi(\mathfrak{G}) := \Phi(\mathfrak{G}, \top)$ and the definition of $\Phi(\mathfrak{G})$ is finished for graphs in standard-form. If $\mathfrak{G}$ is an EGI which is not in standard-form, let $\mathfrak{G}'$ be its standardization, and set $\Phi(\mathfrak{G}) := \Phi(\mathfrak{G}')$. This completes the definition of Φ.

Let $\mathfrak{G}$ be a existential graph and $f := \Phi(\mathfrak{G})$. Similar as for Ψ, it is evident that $(Cut \cup \{\top\}, \leq)$ and $(\mathrm{Neg}_f, \leq)$ are isomorphic quasiorders. We denote the canonically given isomorphism by $\Phi_{Cut} : Cut \cup \{\top\} \to \mathrm{Neg}_f$.

---

[2] Like in Def. 18.1, the signs which have to be understood literally are underlined. For example, the first formula is the sequence of signs which consists of the result of the evaluation of $\kappa(w)$, a left bracket, the result of the evaluation of $\Phi_t(w)$ and a right bracket.

I want to point out that Φ is, strictly speaking, not a function. We have assigned arbitrary variables to the generic nodes, and the order of quantifiers or formulas in conjunctions is arbitrary as well. So Φ determines a formula only up to the names of the variables, the order of quantifiers and the order of the subformulas of conjunctions. To put it more formally: The image $\Phi(\mathfrak{G})$ of a existential graph $\mathfrak{G}$ is only uniquely given up to the following equivalence relation:

Definition 19.4 (Equivalence Relation for Formulas). *Let $\cong$ be the smallest equivalence relation on $\mathcal{FO}$ such that the following conditions hold:*

1. *If f_1, f_2, f_3 are formulas then we have $f_1 \wedge f_2 \cong f_2 \wedge f_1$ and $(f_1 \wedge f_2) \wedge f_3 \cong f_1 \wedge (f_2 \wedge f_3)$,*
2. *if f_1, f_2 are formulas with $f_1 \cong f_2$ and if α, β are variables then $\exists\alpha.\exists\beta.f_1 \cong \exists\beta.\exists\alpha.f_2$,*
3. *if f and f' are equal up to renaming bound variables then $f \cong f'$, and*
4. *if f_1, f_2, g_1, g_2 are formulas with $f_1 \cong f_2$ and $g_1 \cong g_2$ then $\neg f_1 \cong \neg f_2$, $f_1 \wedge g_1 \cong f_2 \wedge g_2$ and $\exists\alpha.f_1 \cong \exists\alpha.f_2$.*

It is well known that the relation $\cong$ respects the meaning of a formula, i.e., for formulas f, g with $f \cong g$ we have $f \models g$ and $g \models f$ (and thus we have $f \vdash g$ and $g \vdash f$). So all possible images $\Phi(\mathfrak{G})$ of a existential graph $\mathfrak{G}$ are semantically and provably equivalent and can therefore be identified, and we consider Φ a mapping which assigns a formula to each existential graph.

To illustrate the mappings Φ and Ψ, I give a small sample. A well-known example (see [Sow97a], Fig. 12) for translating formulas to existential graphs is the formula which expresses that a binary relation F is a (total) function.

$$f \ := \ \forall x.\exists y.(xFy \wedge \forall z.(xFz \to (y = z)))$$

This formula is written down only by using the $\exists$-quantifier and the junctors $\wedge$ and $\neg$. It is

$$f \ := \ \neg\exists x.\neg\exists y.(xFy \wedge \neg\exists z.(xFz \wedge \neg(y = z)))$$

In Fig. 19.1, the EGI $\mathfrak{G}_f := \Psi(f)$ is depicted. Each variable occurrence of f generates a vertex in $\Psi(f)$. To make the translation more transparent, each vertex is labeled with its generating variable occurrences of f.

The existential graph in Fig. 19.1 can be translated back to a first order logic formula by the mapping Φ. One possible result (up to the chosen variables and to the order of the subformulas) is:

$$\Phi(\Psi(f)) \ = \ \neg\exists u.\neg\exists v.(uFv \wedge \neg\exists w.(\neg(v = w) \wedge uFw))$$

It is easy to see that $\Phi(\Psi(f) \cong f$, i.e., f is another possible result of $\Phi(\Psi(f))$. As $\Phi(\mathfrak{G})$ is be given only up to the relation $\cong$, it is clear that we cannot prove

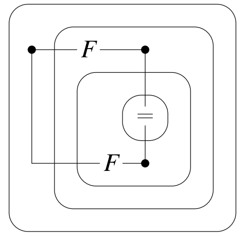
$\Psi(f) =$ 

Fig. 19.1. The EGI $\Psi(f)$

that $\Phi \circ \Psi$ is the identity mapping. Moreover, we usually will even not have $f \cong \Phi \circ \Psi(f)$. For example, consider the following formula:

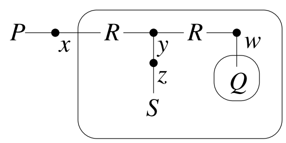
$$g \ := \ \exists y.P(y) \wedge \exists x.\exists x.R(x,y)$$

Then we have

$$\Phi(\Psi(g)) \ = \ \exists y.\exists x.P(y) \wedge R(x,y)$$

Roughly speaking, $\Phi \circ \Psi$ removes superfluous quantifiers, and moreover, it may move quantifiers outwards.

In contrast to $\Phi \circ \Psi$, $\Psi \circ \Phi$ is a well-defined mapping which maps graphs to graphs. To exemplify the mapping Φ, consider

$\mathfrak{G} :=$ 

For $\mathfrak{G}$, we have

$$\Phi(\mathfrak{G}) = \exists x.(P(x) \wedge \neg\exists y.\exists z.\exists w.(R(x,y) \wedge y = z \wedge S(z) \wedge R(y,w) \wedge \neg Q(w)))$$

and $\Psi(\Phi(\mathfrak{G})) = \mathfrak{G}$. This equality holds for $\mathfrak{G}_f$ as well: From $\Phi(\Psi(f) \cong f$ we conclude $\Psi(\Phi(\mathfrak{G}_f)) = \mathfrak{G}_f$. The examples suggest that $\Psi(\Phi(\mathfrak{G})) = \mathfrak{G}$ holds for each EGI $\mathfrak{G}$. In fact, we will prove this conjecture for each graph $\mathfrak{G}$ in standard-form (due to the definition of Φ, this is the best we can expect). This will be done in the next chapter, in Sect. 20.2.

19.2 Semantical Equivalence between Graphs and Formulas

In this section, we will show that the mappings Φ and Ψ preserve the meaning of graphs resp. formulas.

We start our investigation with the mapping Φ.

Theorem 19.5 (Main Semantical Theorem for Φ). *Let an EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be given and let $\mathcal{M}$ be a relational structure over $\mathcal{A}$. Then we have the following equivalence:*

$$\mathcal{M} \models \mathfrak{G} \quad \Longleftrightarrow \quad \mathcal{M} \models \Phi(\mathfrak{G})$$

Proof: Recall that Φ is defined inductively over $Cut \cup \{\top\}$ for graphs in standard-form, but due to Lem. 19.3, we can assume w.l.o.g. that $\mathfrak{G}$ is an EGI in standard form. Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EG. Like Φ, the relation $\models_{class}$ between models and graphs is inductively defined as well. In the definition of $\models_{class}$, we needed total valuations $ref : V \to U$. In the definition of Φ, we assigned to each vertex $v \in V$ a variable $\alpha_v \in \text{Var}$. Thus, we can canonically transform each valuation $ref : V \to U$ to a valuation $val_{ref} : \text{Var} \to U$ on the set of variables by setting $val_{ref}(\alpha_v) := ref(v)$ for each variable in $\{\alpha_v \mid v \in V\}$ (for all other variables α, the image $val_{ref}(\alpha)$ is arbitrary). Using this definition, it is easy to show inductively over $Cut \cup \{\top\}$ that, for every total valuation $ref : V \to U$ and every context $c \in Cut \cup \{\top\}$, we have the following equivalence:

$$\mathcal{M} \models_{class} \mathfrak{G}[c, ref] \Longleftrightarrow \mathcal{M} \models_{val_{ref}} \Phi(\mathfrak{G}, c)$$

If we take now an arbitrary valuation ref, we have

$$\mathcal{M} \models_{class} \mathfrak{G} \Leftrightarrow \mathcal{M} \models_{class} \mathfrak{G}[\top, ref] \Leftrightarrow \mathcal{M} \models_{val_{ref}} \Phi(\mathfrak{G}, \top) \Leftrightarrow \mathcal{M} \models \Phi(\mathfrak{G})$$

The last equivalence holds because $\Phi(\mathfrak{G}) = \Phi(\mathfrak{G}, \top)$ has no free variables. □

Next we will have to show the corresponding result for Ψ. The idea of the proof cannot directly be adopted for the following reason: In the definition of Ψ, we considered EGIs with variables. These graphs are a mere helping construct. So far, we have not defined how EGIs with variables are evaluated in models. In order to prove that Ψ respects the entailment-relation $\models$ as well, we first have to extend our semantics to EGIs with variables. This will be done analogously to $\mathcal{FO}$, that is, we treat variable-vertices like free variables in $\mathcal{FO}$.

For EGIs with variables, the assignment of objects to variable vertices will be adopted from ($\mathcal{FO}$-) valuations $val : \text{Var} \to U$. For this reason, partial and total valuations of EGIs with variables are defined to be valuations which assign objects only to the *generic* vertices. This yields the following definition:

Definition 19.6 (Valuations for EGIs with Variables). *Let $\mathfrak{G}$ be an EGI with variables, and let $\mathcal{M} = (U, I)$ be a relation structure. Each mapping $ref : V' \to U$ with $V' \subseteq V^*$ is called a* PARTIAL VALUATION OF $\mathfrak{G}$. *If $c \in Cut$, $V' \supseteq \{v \in V^* \mid v > c\}$ and $V' \cap \{v \in V^* \mid v \leq c\} = \emptyset$, then ref is called a partial valuation* for the context c. *If $V' = V^*$, then ref is called* (TOTAL) VALUATION OF $\mathfrak{G}$.

Let $val : Var \to U$ be a ($\mathcal{FO}$-)valuation, and let $ref : V' \to U$ be a partial valuation for $\mathfrak{G}$. Let $ref \mathbin{\dot\cup} val : V' \mathbin{\dot\cup} V^{Var} \to U$ be the mapping with

$$(ref \,\dot\cup\, val)(v) = \begin{cases} ref(v) & v \in V' \\ val(\alpha) & v \in V^{Var} \text{ with } \rho(v) = \alpha \end{cases}$$

Now $\mathcal{M} \models_{endo} \mathfrak{G}$ resp. $\mathcal{M} \models_{class} \mathfrak{G}$ are defined exactly like in Def. 13.4 resp. in Def. 13.3 for combined valuations $ref \,\dot\cup\, val$.

Definition 19.7 (Endoporeutic Evaluation of EGIs with Variables). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be an existential graph and let (U, I) be a relational structure over $\mathcal{A}$. Inductively over the tree $Cut \cup \{\top\}$, we define $(U, I) \models_{endo} \mathfrak{G}[c, ref \,\dot\cup\, val]$ for each context $c \in Cut \cup \{\top\}$ and every partial valuation $ref : V' \subseteq V \to U$ for c:*

$(U, I) \models_{endo} \mathfrak{G}[c, ref] :\Longleftrightarrow$

ref can be extended to a partial valuation $\overline{ref} : V' \cup (V \cap area(c)) \to U$ (i.e., $\overline{ref}(v) = ref(v)$ for all $v \in V'$), such that the following conditions hold:

- *$(\overline{ref} \,\dot\cup\, val)(e) \in I(\kappa(e))$ for each $e \in E \cap area(c)$ (edge condition))*
- *$(U, I) \not\models_{endo} \mathfrak{G}[d, \overline{ref} \,\dot\cup\, val)]$ for each $d \in Cut \cap area(c)$ (cut condition and iteration over $Cut \cup \{\top\}$))*

If $val : Var \to U$ is a valuation with $\mathcal{M} \models_{endo} \mathfrak{G}[\top, \emptyset \,\dot\cup\, val]$, we write $\mathcal{M} \models_{endo} \mathfrak{G}[val]$ for short.

Lemma 19.8 (Replacing a Generic Marker by a Variable). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be an EGI with variables, let $\alpha \in Var$ be a variable such that there is no vertex $w \in V$ with $\rho(v) = \alpha$. Let $v \in area(\top)$ be a generic vertex. Let $\mathfrak{G}_{[v \mapsto \alpha]}$ be obtained from $\mathfrak{G}$ be exchanging the label of v from '$*$' to α (i.e., we set $\mathfrak{G}_{[v \mapsto \alpha]} := (V, E, \nu, \top, Cut, area, \kappa, \rho_{[v \mapsto \alpha]})$, where $\rho_{[v \mapsto \alpha]}(v) = \alpha$, and $\rho_{[v \mapsto \alpha]}(w) = \rho(w)$ for all $w \neq v$). Let $\mathcal{M} := (U, I)$ be a model and let $val : Var \to U$ be a valuation. Then we have*

$$\mathcal{M} \models_{endo} \mathfrak{G}[val] \iff \text{there is an } u \in U \text{ with } \mathcal{M} \models_{endo} \mathfrak{G}_{[v \mapsto \alpha]}[val_{[\alpha \to u]}] \quad ,$$

where $val_{[\alpha \to u]} : Var \to U$ is the valuation with $val_{[\alpha \to u]}(\alpha) = u$ and $val_{[\alpha \to u]}(\beta) = val(\beta)$ for all variables $\beta \in Var$ with $\beta \neq \alpha$.

Proof: Follows immediately from the definition of $\models_{endo}$ for EGIs and EGIs with variables. □

Theorem 19.9 (Main Semantical Theorem for Ψ). *Let f be a sentence and let $\mathcal{M}$ be a relational structure over $\mathcal{A}$. Then we have the following equivalence:*

$$\mathcal{M} \models f \iff \mathcal{M} \models \Psi(f)$$

Proof: Ψ is defined inductively over the construction of formulas. The same holds for the relation $\models$ between models and formulas. In the definition of $\models$, we needed valuations $val : \mathrm{Var} \to U$. Let f be a formula, $\mathfrak{G} := \Psi(f)$, and let $val : \mathrm{Var} \to U$ a valuation for the variables. In the definition of Ψ, we assigned to each *occurrence* of a variable $\alpha \in \mathrm{Var}$ (including the occurrences of α after an quantifier $\exists$ a vertex v_α). Now the valuation val can canonically be transformed to a valuation $ref_{val} : V \to U$ by setting $ref_{val}(v_\alpha) := val(\alpha)$. Then ref_{val} is a total valuation for the EGI with variables $\mathfrak{G}$.

Let $\mathcal{M}$ be a relational structure. We will show by induction over the construction of formulas that we have

$$\mathcal{M} \models_{val} f \quad \Longleftrightarrow \quad \mathcal{M} \models_{endo} \mathfrak{G}[val] \tag{19.1}$$

for arbitrary formulas f, $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho) := \Psi(f)$, and valuations $ref : \mathrm{Var} \to U$.

- For $f = R(\alpha_1, \ldots, \alpha_n)$ with $R \in \mathcal{R}$ and variables $\alpha_1, \ldots, \alpha_n$, Eqn. (19.1) is obviously satisfied.
- Let $f = \neg g$. For $\mathfrak{G}_g := (V_g, E_g, \nu_g, \top_g, Cut_g, area_g, \kappa_g, \rho_g) = \Psi(g)$ and $\mathfrak{G}_f := (V_f, E_f, \nu_f, \top_f, Cut_f, area_f, \kappa_f, \rho_f) = \Psi(f)$, we have:

$$\begin{aligned} \mathcal{M} \models_{val} f \quad & \Longleftrightarrow \quad \mathcal{M} \not\models_{val} g \\ & \overset{\text{ind. hyp}}{\Longleftrightarrow} \quad \mathcal{M} \not\models_{endo} \mathfrak{G}_g[val] \\ & \overset{\text{Def. } \Psi \text{ and. } \models_{\text{endo}}}{\Longleftrightarrow} \quad \mathcal{M} \models_{endo} \mathfrak{G}_f[val] \end{aligned}$$

- The case $f = f_1 \wedge f_2$ is done analogously to the last case.
- Let $f = \exists \alpha . g$. Let $\mathfrak{G}_g := (V_g, E_g, \nu_g, \top_g, Cut_g, area_g, \kappa_g, \rho_g) = \Psi(g)$, $\mathfrak{G}_f := (V_f, E_f, \nu_f, \top_f, Cut_f, area_f, \kappa_f, \rho_f) = \Psi(f)$. If $\alpha \notin \mathrm{FV}(g)$, we have $\mathfrak{G}_f = \mathfrak{G}_g$, and we are done. So let $\alpha \in \mathrm{FV}(f)$. In existential step of Ψ, the following steps are carried out:
  1. A new vertex v_0 with $\rho(v_0) = \alpha$ is juxtaposed to $\Psi(f)$. The resulting graph is denoted by $\mathfrak{G}_1$.
  2. On each hook (e, i) such that a vertex v with $\rho(v) = \alpha$ is attached to it, we replace v by v_0. The resulting graph is denoted by $\mathfrak{G}_2$.
  3. All vertices $v \neq v_0$ with $\rho(v) = \alpha$ are erased. The resulting graph is denoted by $\mathfrak{G}_3$.
  4. The label of v is changed from α to $*$. We obtain $\mathfrak{G}_f$.

  In order to prove this case, we start with the definition how f is evaluated in $\mathcal{M}$. We have:

$$\mathcal{M} \models_{val} \exists \alpha . g \overset{\text{Def.} \models}{\Longleftrightarrow} \text{ there is an } u \in U \text{ with } \mathcal{M} \models_{val_{[\alpha \to u]}} g \tag{19.2}$$

  Our induction hypothesis yields that the right side of Eqn. (19.2) is equivalent to

$$\text{there is an } u \in U \text{ with } \mathcal{M} \models_{endo} \mathfrak{G}_g[val_{[\alpha \to u]}] \qquad (19.3)$$

It is trivial that the or erasure insertion of a vertex v_0, either labeled with a generic marker, or with an variable, does not change the meaning of an EGI with variables. Particularly, Eqn. (19.3) is equivalent to

$$\text{there is an } u \in U \text{ with } \mathcal{M} \models_{endo} \mathfrak{G}_1[val_{[\alpha \to u]}] \qquad (19.4)$$

In the evaluation of EGIs with variables, we have already assigned objects to the variable vertices when we start the evaluation. For this reason, replacing a vertex labeled with α by another vertex labeled with α as well on a hook does not change the meaning of the graph. Thus, Eqn. (19.4) is equivalent to

$$\text{there is an } u \in U \text{ with } \mathcal{M} \models_{endo} \mathfrak{G}_2[val_{[\alpha \to u]}] \qquad (19.5)$$

Analogously to the step from Eqn. (19.3) to Eqn. (19.4), we now get that Eqn. (19.5) is equivalent to:

$$\text{there is an } u \in U \text{ with } \mathcal{M} \models_{endo} \mathfrak{G}_3[val_{[\alpha \to u]}] \qquad (19.6)$$

We have $\mathfrak{G}_3 = (\mathfrak{G}_f)_{[v_0 \to \alpha]}$. Now Lem. 19.8 yields that Eqn (19.6) is equivalent to $\mathcal{M} \models_{endo} \mathfrak{G}_f[val]$, thus the existential step is finished.

Now let f be a sentence and let $\mathfrak{G} := \Psi(f)$. As f has no free variables, and as $\Psi(f)$ has no variable-vertices, we have for each valuation $val : \mathrm{Var} \to U$ (which is simply irrelevant)

$$\mathcal{M} \models f \iff \mathcal{M} \models_{val} f \iff \mathcal{M} \models_{endo} \mathfrak{G}[val] \iff \mathcal{M} \models_{endo} \mathfrak{G} \quad ,$$

and we are done. □

From Thms. 19.5 and 19.9, we get analogously to Alpha (see Cor. 10.6) the following corollary:

Corollary 19.10 (Φ and Ψ respect $\models$). *Let $\mathfrak{H}$ be a set of EGIs and let $\mathfrak{G}$ be an EGI, and let f be a formula and F be a set of formulas. Then we have:*

$$F \models f \iff \Psi(F) \models \Psi(f) \quad \text{, and} \quad \mathfrak{H} \models \mathfrak{G} \iff \Phi(\mathfrak{H}) \models \Phi(\mathfrak{G}) \quad .$$

Moreover we have that $\mathfrak{G}$ and $\Psi(\Phi(\mathfrak{G}))$, as well as f and $\Phi(\Psi(f))$, are semantically equivalent.

Proof: Analogously to the proof of Cor. 10.6. □

20

Syntactical Equivalence to $\mathcal{FO}$ and Completeness

In Chpt. 13 we provided two translations Φ and Ψ between the logical systems of EGIs and $\mathcal{FO}$. Both systems are equipped with a semantical entailment relation $\models$ and a derivability relation $\vdash$. In Chpt. 13 we have already shown that Φ and Ψ respect the entailment relation $\models$. Moreover, we know that $\vdash$ is sound and complete on the side of $\mathcal{FO}$ (i.e., for two formulas f_1, f_2, we have $f_1 \vdash f_2 \Leftrightarrow f_1 \models f_2$), and $\vdash$ is sound for EGIs (i.e., for two EGIs $\mathfrak{G}_1, \mathfrak{G}_2$, we have $\mathfrak{G}_1 \vdash \mathfrak{G}_2 \Rightarrow \mathfrak{G}_1 \models \mathfrak{G}_2$). From this, we can conclude that Φ respects the syntactical entailment relation $\vdash$ as well. But so far, we neither can conclude that $\vdash$ is complete for EGIs, nor that Ψ respects $\vdash$. In this chapter, this gap will be closed.

In Sec. 20.1, we will show that Ψ respects $\vdash$, i.e., $f_1 \vdash f_2 \Rightarrow \Psi(f_1) \vdash \Psi(f_2)$. Unfortunately, this result is not sufficient to show that $\vdash$ is complete for EGIs: For any graph $\mathfrak{G}$ we know that $\mathfrak{G}$ and $\Psi(\Phi(\mathfrak{G}))$ have the same meaning, but it cannot be proven that $\mathfrak{G}$ and $\Psi(\Phi(\mathfrak{G}))$ are provably equivalent. But as the presented examples in the last chapter already indicated, in Sec. 20.2 an even better result is proven: For any graph $\mathfrak{G}$ in standard-form, we have $\mathfrak{G} = \Psi(\Phi(\mathfrak{G}))$. These two results of this chapter will be sufficient to show finally that $\vdash$ is complete for EGIs.

20.1 Ψ respects $\vdash$

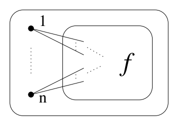
In this section we want to show that Ψ respects the derivability relation $\vdash$, i.e., we want to show that we have $f_1 \vdash f_2 \Longrightarrow \Psi(f_1) \vdash \Psi(f_2)$ for formulas f_1, f_2. The relations $\vdash$ for formulas resp. for EGIs are based on the appropriate calculi, so the idea of the proof is to show that Ψ respects every rule of the calculus for $\mathcal{FO}$. We want to point out that the calculus for $\mathcal{FO}$ is based on formulas with free variables, in contrast to the calculus on $\mathcal{EGI}$ which is based on EGIs without variables. For this reason we have to translate formulas *with* *free* variables to EGIs *without* variables. This can canonically be done by a slight modification of the mapping Ψ.

Definition 20.1 (Universal Closures of Formulas and $\Psi_\forall$). *Let f be a formula with $FV(f) = \alpha_1, \ldots, \alpha_n$. Then $f_\forall := \neg\exists\alpha_1 \ldots \exists\alpha_n \neg f$ is called* UNIVERSAL CLOSURE OF f. *Now we set*

$$\Psi_\forall := \begin{cases} \mathcal{FO} \to \mathcal{EGI} \\ \quad f \mapsto \Psi(f_\forall) \end{cases}$$

For each formula, $\Psi_\forall(f)$ is an EGIs without variables. The definition of the relation $\models$ in $\mathcal{FO}$ yields that a formula f is valid in a relational structure if and only if $f_\forall$ is valid in that structure. For this reason we have to focus on the universal closure of f (instead of the corresponding existential closure) and hence on the mapping $\Psi_\forall$.

Due to its definition, the mapping $\Psi_\forall$ shall be represented as follows:

$\Psi_\forall(f) :=$ 

In this notation, f shall stand for the subgraph which is generated by f. But this generated subgraph is *not* $\Psi(f)$, but the subgraph we get after the existential steps are applied to $\Psi(f)$. For this reason we write f instead $\Psi(f)$ in this diagrams, although f shall denote a subgraph of the written graph.

Now we are prepared to prove that $\Psi_\forall$ respects $\vdash$.

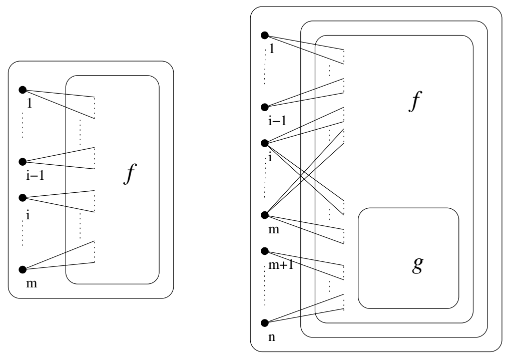
Lemma 20.2 ($\Psi_\forall$ Respects Syntactical Entailment). *Let $f_1, \ldots, f_n \vdash g$, $n \in \mathbb{N}_0$ be a rule of the calculus for $\mathcal{FO}$. Then we have $\Psi_\forall(f_1), \ldots, \Psi_\forall(f_n) \vdash \Psi_\forall(g)$ in the calculus for EGIs.*

Proof: We have to show the lemma for each rule.

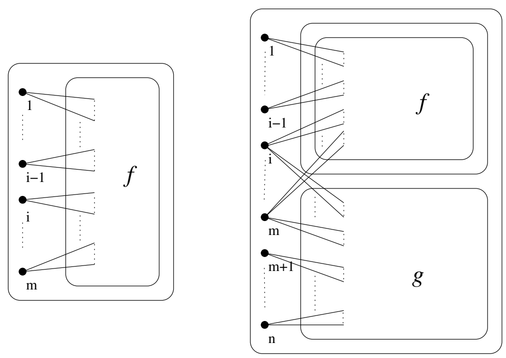
Modus Ponens: $f, f \to g \quad \vdash \quad g$

Let f, g be two formulas. Without loss of generality let $\text{FV}(f) = \{x_1, \ldots x_m\}$, and $\text{FV}(g) = \{x_i, \ldots x_n\}$.

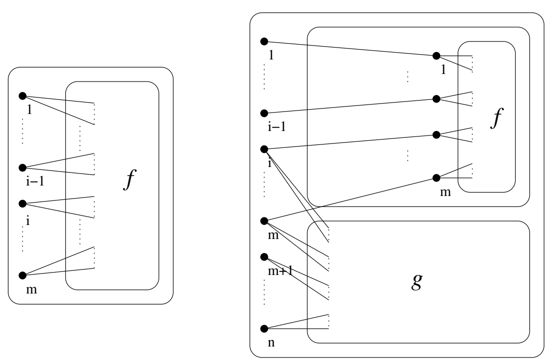
The graph we start with is the juxtaposition of the graphs $\Psi_\forall(f)$, $\Psi_\forall(f \to g)$:



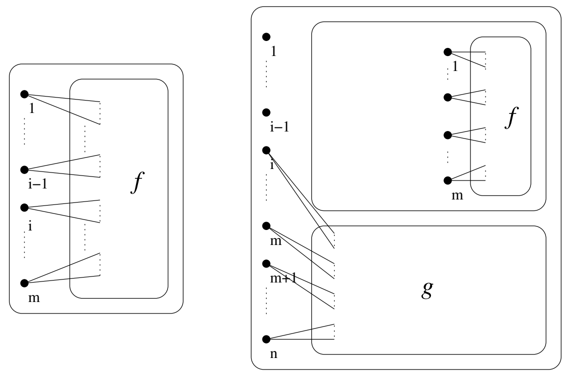
The double cut rule yields:



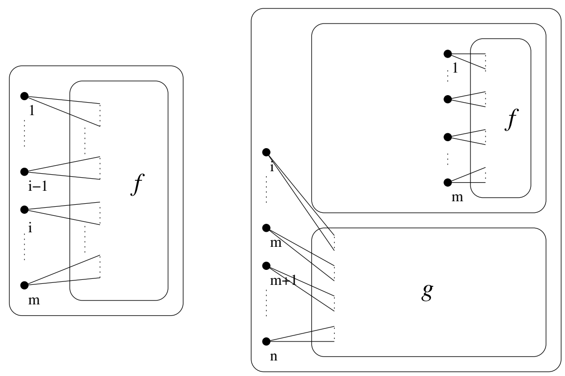
We split each vertex which has an index from 1 to $i-1$:



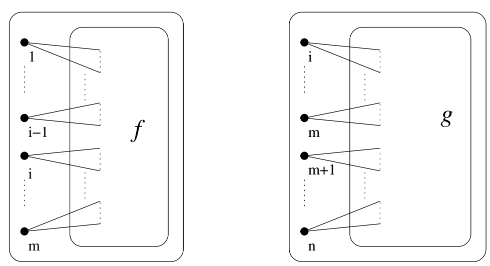
All new identity edges are erased:



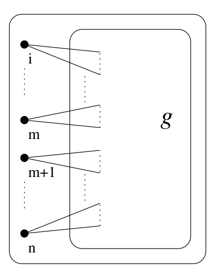
All vertices with an index between 1 and $i-1$ are now erased:



Deiteration yields:



Now erasure yields:

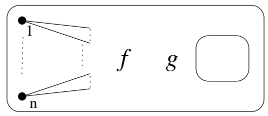


This is $\Psi_\forall(g)$, hence we have $\Psi_\forall(f)\,,\,\Psi_\forall(f \to g) \vdash \Psi_\forall(g)$.

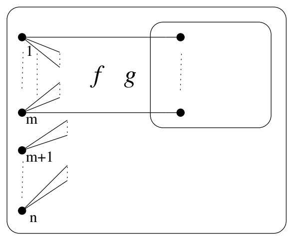
P1: $\vdash\ f \to (g \to f)$

Let f, g be two formulas. Without loss of generality let $\mathrm{FV}(f) := \{x_1, \ldots x_m\}$ and $\mathrm{FV}(g) := \{x_i, \ldots x_n\}$.

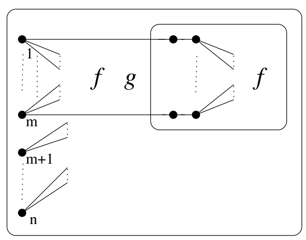
From the sheet of assertion, we can derive with double cut and insertion:



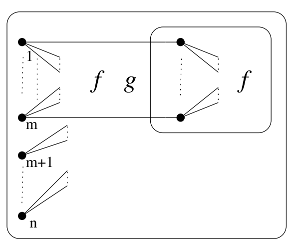
We iterate each vertex which has an index from 1 to m into the inner cut as follows:



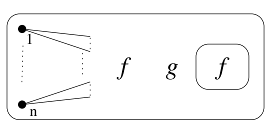
Iteration of the subgraph which is generated by f yields:



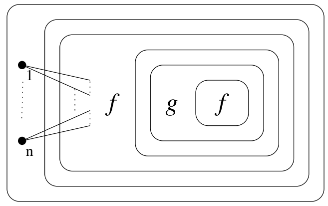
An m-fold application of the rule 'removing a vertex' yields:



We merge the vertices in the inner cut into the outer cut:



Double cut yields:

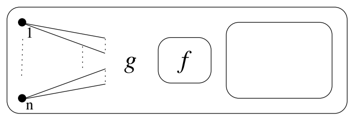


This is $\Psi_\forall((f \to (g \to f)))$.

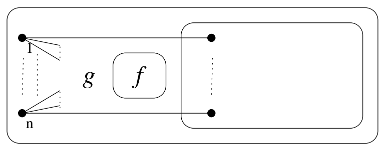
P2: $\vdash\ (\neg f \to \neg g) \to (g \to f)$

The scheme for the proof of P2 is the same like in the proof for P1. Again let f, g be two formulas with $\mathrm{FV}(f) := \{x_1, \dots x_m\}$ and $\mathrm{FV}(g) := \{x_i, \dots x_n\}$.

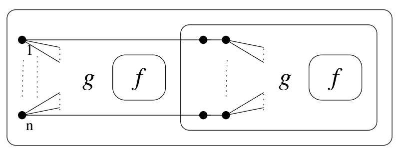
From the sheet of assertion, we can derive with double cut and insertion:



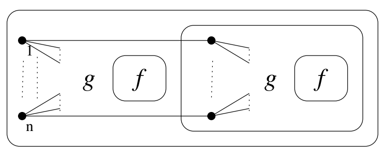
We iterate each vertex which has an index from 1 to n into the inner cut as follows:



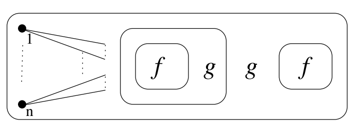
Iteration of the inserted subgraph yields:



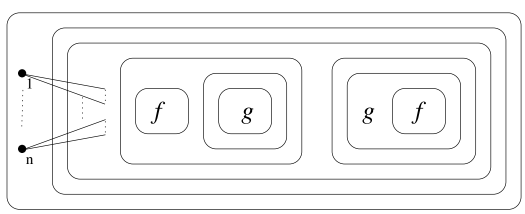
An n-fold application of the rule 'removing a vertex' yields:



We merge the vertices in the inner cut into the outer cut:



Double cut yields:

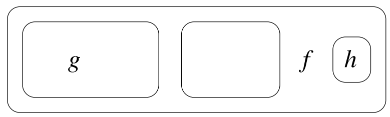


This is $\Psi_\forall((\neg f \to \neg g) \to (g \to f))$.

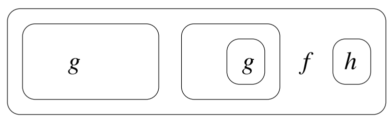
P3: $\vdash\ (f \to (g \to h)) \to ((f \to g) \to (f \to h))$

We have to show $\vdash \Psi_\forall((f \to (g \to h)) \to ((f \to g) \to (f \to h)))$. The scheme for the proof of P3 is the same like of the proofs for P1 and P2. From the empty sheet of assertion, we apply the rules double cut, insertion, iteration, removing vertices and double cut to get the desired graph. To simplify matters we assume that f, g and h have no free variables (in case of free variables, an additional application of the identity-erasure-rule is needed). The proof is now done as follows:

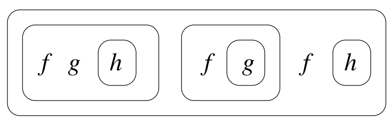
On the empty sheet of assertion we draw a double cut and insert f, g and h into the outer cut:



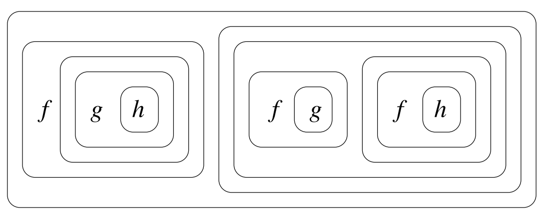
We iterate g into the inner cut:



Now a twofold iteration of f and a iteration of h yields:



Now a threefold application of the double cut rule yields:



This is $\Psi_\forall((f \to (g \to h)) \to ((f \to g) \to (f \to h)))$.

The rules MP, P1, P2 and P3 form a complete set of rules for propositional logic, hence they are in $\mathcal{FO}$ sufficient to derive all tautologous formulas f. Hence we now have $\Psi_\forall(f)$ for all tautologous formulas f. This will be used in the rest of the proof.

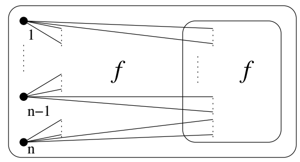
Ex1: $\vdash\ f \to \exists\alpha.f$

Let α be the variable x_n. We have to show $\vdash \Psi_\forall(f \to \exists x_n.f)$. To do this, we distinguish two cases.

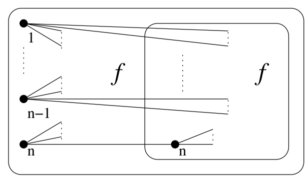
First, let $x_n \notin Free\mathrm{FV}(f)$. Then we have $\Psi_\forall(f \to \exists x_n.f) = \Psi_\forall(f \to f)$. As $f \to f$ is tautologous, we can derive $\Psi_\forall(f \to f)$, i.e., we can derive the graph $\Psi_\forall(f \to \exists x_n.f)$.

Now let $x_n \in \mathrm{FV}(f)$. Then $\Psi_\forall(f \to \exists x_n.f)$ is derived as follows:

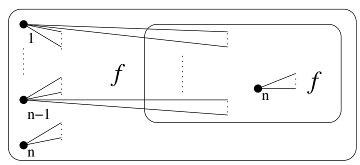
As $f \to f$ is tautologous, we can derive



The n-th vertex $\bullet_n$ is split:



Now the new identity edge is erased:



This is $\Psi_\forall(f \to \exists x_n.f)$.

Ex2: $\vdash\ f \to \exists \alpha_1.f[\alpha_1/\alpha_2]$, if $\alpha_1 \notin \mathrm{FV}(f)$

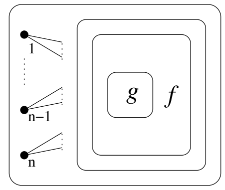
Let f be an formula. We want to show $\vdash \Psi_\forall(f \to \exists \alpha_1.f[\alpha_1/\alpha_2])$, if $\alpha_1 \notin \mathrm{FV}(f)$.

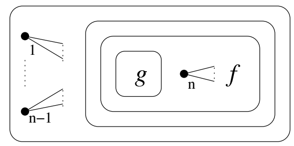
First we consider the case $\alpha_2 \notin \mathrm{FV}(f)$. Then we have $f \to \exists \alpha_1.f[\alpha_2/\alpha_1] = f \to \exists \alpha_1.f$, and furthermore we have $\Psi_\forall(f \to \exists \alpha_1.f) = \Psi_\forall(f \to f)$. As $f \to f$ is tautologous, we conclude $\vdash \Psi_\forall(f \to f)$, thus $\vdash \Psi_\forall(f \to \exists \alpha_1.f[\alpha_2/\alpha_1])$.

Now let $\alpha_2 \in \mathrm{FV}(f)$. It is easy to see that $\Psi(\exists \alpha_2.f) = \Psi(\exists \alpha_1.f[\alpha_2/\alpha_1])$ holds, thus we have $\Psi_\forall(f \to \exists \alpha_1.f[\alpha_2/\alpha_1]) = \Psi_\forall(f \to \exists \alpha_2.f)$. So this case can be reduced to the proof of Ex1.

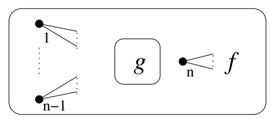
Ex3: $f \to g\ \vdash\ \exists \alpha.f \to g$, if $\alpha \notin \mathrm{FV}(g)$

Let α be the variable x_n, and without loss of generality let $\mathrm{FV}(f \to g) := \{x_1, \ldots x_n\}$. Suppose we have $\vdash \Psi_\forall(f \to g)$ with $x_n \notin \mathrm{FV}(g)$. We have to derive the graph $\Psi_\forall(\exists x_n.f \to g)$. The graphs we have to consider are:

$\Psi_\forall(f \to g) \;=$  and

$\Psi_\forall(\exists x_n.f \to g) \;=$  .

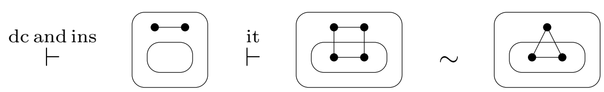
Now $x_n \notin \mathrm{FV}(g)$ yields that we have no identity edge between the n-th vertex $\bullet_n$ and any vertex of $\boxed{g}$. Hence, using double cut, both graphs are equivalent to:[1]



This yields that both graphs are equivalent, too, and the proof for Ex4 is finished.

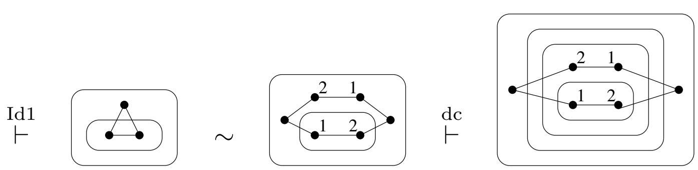
Id1 (reflexivity of identity): $\vdash\; \alpha_0 = \alpha_0$

$\Psi_\forall(\alpha_0 = \alpha_0)$ can be derived as follows:



Id2 (symmetry of identity): $\vdash\; \alpha_0 = \alpha_1 \to \alpha_1 = \alpha_0$

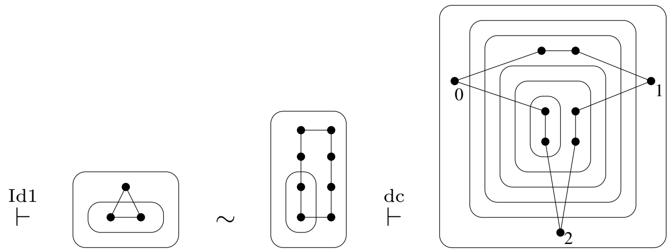
$\Psi_\forall(\alpha_0 = \alpha_1 \to \alpha_1 = \alpha_0)$ can be derived as follows:



Id3 (transitivity of identity): $\vdash\; \alpha_0 = \alpha_1 \to \alpha_1 = \alpha_2 \to \alpha_0 = \alpha_2$

$\Psi_\forall(\alpha_0 = \alpha_1 \to \alpha_1 = \alpha_2 \to \alpha_0 = \alpha_2)$ can be derived as follows (where we have (we have labeled the vertices, so that it is easier to realize which vertex belongs to which variable):

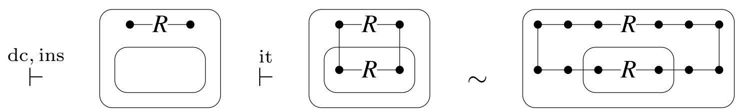
[1] In fact $\forall x_n.(f \to g)$ and $\exists x_n.f \to g$ are equivalent, too.



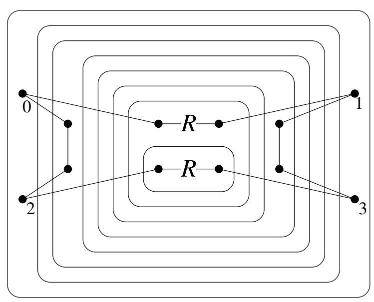
Cong2: $\vdash \alpha_0 = \alpha_n \to \ldots \to \alpha_{n-1} = \alpha_{2n-1} \to R(\alpha_0, \ldots, \alpha_{n-1}) \to R(\alpha_n, \ldots, \alpha_{2n})$

We perform this proof only for relation names with the arity 2, i.e., we consider $\Psi_\forall(\alpha_0 = \alpha_2 \to \alpha_1 = \alpha_3 \to R(\alpha_0, \alpha_1) \to R(\alpha_2, \alpha_3))$.

This proof can be performed as well for higher (or lower) arities, but then the graphs we need have a poor readability. So let $R \in \mathcal{R}_2$. We have:



A threefold application of the double-cut rule yields



which is the desired EGI (again we have labeled the vertices, so that it is easier to realize which vertex belongs to which variable). $\Box$

We have shown that each axiom and rule of the $\mathcal{FO}$-calculus is respected by the mapping $\Psi : \mathcal{FO} \to \mathcal{EGI}$. Thus the last lemma yields immediately that $\Psi : \mathcal{FO} \to \mathcal{EGI}$ respects the syntactical entailment relation $\vdash$, i.e., we have the following theorem.

Theorem 20.3 (Main Syntactical Theorem for Ψ). *Let F be a set of $\mathcal{FO}$-formulas and f be a $\mathcal{FO}$-formula (all possibly with free variables). Then we have*

$$F \vdash f \quad \Longrightarrow \quad \{\Psi_\forall(g) \mid g \in F\} \vdash \Psi_\forall(f)$$

In particular we have $f_1 \vdash f_2 \Longrightarrow \Psi(f_1) \vdash \Psi(f_2)$ for formulas f_1, f_2 without free variables.

Proof: A canonical proof based on using Lem. 20.2, carried out by induction over the length of the proofs, yields immediately:

$$\text{If } f_1, f_2 \text{ are two } \mathcal{FO}\text{-formulas with } f_1 \vdash f_2, \text{ then } \Psi_\forall(f_1) \vdash \Psi_\forall(f_2) \ . \tag{20.1}$$

Let F be a set of formulas and f be a formula with $F \vdash f$, i.e. we have $g_1, \ldots, g_n \in F$ with $g_1, \ldots, g_n \vdash f$. We set $F_\forall := \{g_\forall \mid g \in F\}$, $\Psi_\forall(f)_=\Psi(f_\forall)$, and $f_i := (g_i)_\forall$ for $1 \leq i \leq n$. We get $f_1, \ldots f_n \vdash f_\forall$. thus $f_1 \wedge \ldots \wedge f_n \vdash f_\forall$. Now Eqn. 20.1 yields $\Psi(f_1 \wedge \ldots \wedge f_n) \vdash \Psi(f_\forall)$. From the definition of Ψ and $\Psi_\forall$ we obtain $\Psi(f_1 \wedge \ldots \wedge f_n) = \Psi(f_1) \ldots \Psi(f_n) = \Psi_\forall(g_1) \ldots \Psi_\forall(g_n)$, which yields the main proposition of theorem. □

20.2 Identity of $\mathfrak{G}$ and $\Psi(\Phi(\mathfrak{G}))$ and Completeness of $\vdash$

At the end of Sect. 19.1, after the Definitions of Φ and Ψ, we have already investigated some examples for the mapping Φ and Ψ. In this section, we will prove the conjecture we had claimed after these examples, that is, we will show that each for $\mathfrak{G} = \Psi(\Phi(\mathfrak{G}))$ holds for each EGI $\mathfrak{G}$ in standard-form.

Theorem 20.4 ($\mathfrak{G} = \Psi(\Phi(\mathfrak{G}))$ for Graphs in Standard-Form). *For each EGI $\mathfrak{G}$ in standard-form, we have $\mathfrak{G} = \Psi(\Phi(\mathfrak{G}))$.*

Proof: Let $\mathfrak{G}$ be an arbitrary EGI, let $f := \Phi(\mathfrak{G})$. We already know that the mapping Φ_{Cut} is a bijection (more precisely: an isomorphism) between the contexts of $\mathfrak{G}$, i.e. $Cut \cup \{\top\}$, and f and the subformula occurrences of f which start with an '$\neg$', i.e. $\text{Neg}_f \cup \{\top_f\}$ (see the remark after the definition of Φ, p. 209. It should be noted that the notation Φ_{Cut} is a little bit sloppy: Each graph $\mathfrak{G}$ induces a separate mapping Φ_{Cut} on the contexts *of that graph*). Analogously, due to the definition of Φ, we have

- a bijection Φ_V between the vertices of $\mathfrak{G}$ and the subformula occurrences which starts with an '$\exists\alpha$' (with $\alpha \in \text{Var}$) of f, and
- a bijection Φ_E between the edges of $\mathfrak{G}$ and the atomar subformula occurrences of f,

and these bijections satisfy that

- for $v \in V$, $\Phi_V(v)$ is a subformula of $\Phi_{Cut}(ctx(v))$ if and only if $v \leq c$, and
- for $e \in E$, $\Phi_E(e)$ is a subformula of $\Phi_{Cut}(ctx(e))$ if and only if $e \leq c$.

This can easily be shown with an inductive proof over the inductive construction of Φ.

Now we consider formulas g which do not have any two different subformula occurrences of the form $\exists\alpha.h$ with $\alpha \notin \mathrm{FV}(h)$ (this restriction is necessary because in the definition of Ψ, in the existential step, we treated this case separately). First of all, similar to Ψ, we have a mapping Ψ_{Neg} which is an isomorphism) between $\mathrm{Neg}_h \cup \{\top_h\}$ and the contexts of $\Psi(g)$ (see the remark after the definition of Ψ, p. 208). Analogously, due to the definition of Ψ, we have

- a bijection $\Psi_\exists$ between the subformula occurrences which starts with an '$\exists\alpha$' (with $\alpha \in \mathrm{Var}$) of g and the generic vertices of $\Psi(g)$,
- a bijection Ψ_R between the atomar subformula occurrences of g and the edges of $\Psi(g)$,

and these bijections satisfy that

- a subformula occurrence h_1 which starts with an '$\exists\alpha$' is a subformula of a subformula occurrence $h_2 \in \mathrm{Neg}_f$ if and only if $\Psi_exists(h_1) \leq \Psi_{Neg}(h_2)$, and
- an atomar subformula occurrence h_1 is a subformula of a subformula occurrence $h_2 \in \mathrm{Neg}_f$ if and only if $\Psi_R(h_1) \leq \Psi_{Neg}(h_2)$.

Note that the range of mapping $\Psi_\exists$ is not the set of all vertices in $\Psi(f)$, but the set of all generic vertices.

Now let $\mathfrak{G}$ be an arbitrary EGI. We set $f := \Phi(\mathfrak{G})$ and $\mathfrak{G}' := \Psi(f) := (V', E', \nu', \top', Cut', area', \kappa', \rho')$. As $\mathfrak{G}'$ does not contain any variable-vertices, $\Psi_\exists \circ \Phi_V$ is a bijection from V to V'. As $\Psi_R \circ \Phi_E$ and $\Psi_{Neg} \circ \Phi_{Cut}$ are bijections between E and E resp. $Cut \cup \{\top\}$ and $Cut' \cup \{\top'\}$ as well, and due to our discussion above, we see that $\Psi_\exists \circ \Phi_V \,\dot{\cup}\, \Psi_R \circ \Phi_E \,\dot{\cup}\, \Psi_{Neg} \circ \Phi_{Cut}$ is an isomorphism between $\mathfrak{G}$ and $\mathfrak{G}'$. $\Box$

With this lemma, we can finally prove that the calculus for EGIs is complete.

Theorem 20.5 (Completeness of the Beta-Calculus). *Each set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of EGIs over $\mathcal{A}$ satisfies*

$$\mathfrak{H} \models \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \vdash \mathfrak{G}$$

Proof: Let $\mathfrak{H} \models \mathfrak{G}$. From Cor. 19.10 we conclude $\Phi[\mathfrak{H}] \models \Phi[\mathfrak{G}]$. Theorem 18.7 yields $\Phi[\mathfrak{H}] \vdash \Phi[\mathfrak{G}]$. By definition of $\vdash$ we have $\mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H}$ with $\Phi(\mathfrak{G}_1), \ldots, \Phi(\mathfrak{G}_n) \vdash \Phi(\mathfrak{G})$. By Thm. 20.3 we get $\Psi(\Phi(\mathfrak{G}_1)), \ldots, \Psi(\Phi(\mathfrak{G}_n)) \vdash \Psi(\Phi(\mathfrak{G}))$. For $1 \leq i \leq n$, Thm. 20.4 and Lem. 19.3 yield $\mathfrak{G}_i \vdash \Psi(\Phi(\mathfrak{G}_i))$ and $\Psi(\Phi(\mathfrak{G})) \vdash \mathfrak{G}$. Hence we get $\mathfrak{G}_1, \ldots, \mathfrak{G}_n \vdash \mathfrak{G}$, thus $\mathfrak{H} \vdash \mathfrak{G}$. $\Box$

21

Working with Diagrams of Peirce's Graphs

In the preceeding chapters, we have developed a mathematical theory of Peirce's graphs. Our starting point have been the original manuscripts of Peirce in which he describes his graph system. We had to cope with two problems: First of all, Peirce did not provide a comprehensive, self-contained treatise on his graphs. Instead, his description of existential graphs is scattered throughout different manuscript. What is more important is that Peirce's original manuscripts do by no means satisfy the needs of contemporary mathematics.

Recall that Peirce distinguished between existential graphs and existential graph replicas. Existential graphs can be understood as abstract structures, and their diagrammatic representations are existential graph replicas. In order to develop a mathematical theory of existential graphs, we introduced abstract mathematical structures called *existential graph candidates*, and we defined formal existential graphs as classes of existential graph instances.

Before we start this chapter, let us clarify some notations to ease the discussion. Actually, we have five different, but closely related items to deal with:

1. EGIs as abstract mathematical structures. The are usually denoted by the letter $\mathfrak{G}$.
2. Diagrams of EGIs.
3. Formal existential graphs as classes of EGIs. In the following, I will abbreviate formal existential graphs by EGs. EGs are usually denoted by the letter $\mathfrak{E}$.
4. Diagrams of EGs.
5. Finally, the term 'Peirce graphs' will be used to refer to Peirce's original system, which has been formalized by EGs and their diagrams.

In the preceeding chapters, we elaborated EGs as mathematical logic system, i.e., we developed a mathematical syntax, semantics and the calculus for this system. In order to grasp as best as possible Peirce's understanding of his system, we investigated extensively the informal given syntax and semantics of Peirce's graphs in Chpt. 11, before the mathematical syntax and semantics for EGIs and EGs have been introduced, and we investigated the informal given transformation rules of Peirce in Chpt. 14, before we introduced mathematically a calculus for EGIs and and EGs. To summarize: The mathematical elaboration of EGs has carried out as close as possible to Peirce's understanding. But Peirce's graphs and EGs are not the same: Formal existential graphs are a possible mathematical formalizations of Peirce's original system, which is only informally given.

Defining EGIs and EGs was necessary to elaborate a precise mathematical theory. With this theory, we can now get back to Peirce's original system: The goal of this chapter is to develop a better understanding of Peirce's original system and to provide a purely graphical logic system.

This goal has already partly been fulfilled. First of all, we have already argued that EGs reflect Peirce's understanding of his graphs as best as possible. We provided and discussed how EGIs graphically can be depicted. From this we have obtained a graphical representation of EGs as well, and the graphical representations of EGs correspond to the Peirce's non-degenerated graph replicas. In this understanding, we have fixed the syntax of Peirce's graphs as well of his graph replicas. Particularly, this definition of the syntax of Peirce's graph replicas should not be understood as a mathematical syntax, as the representations of EGIs, thus of EGs, are not mathematically defined. Nonetheless, this syntax is precise enough for an unambiguous understanding of Peirce's graph replicas.

Similarly, we have elaborated a mathematical semantics for EGIs and EGs, which yields a mathematical semantics for Peirce's graph replicas as well. In Chpt. 16, we have moreover provided methods which ease the reading of diagrams of EGIs and EGs, based on mathematical investigations of EGIs. The calculus of Peirce's graphs has been captured as a formal calculus on these structures. Nonetheless, this calculus should be understood as a diagrammatic calculus, i.e. the rules should be understood as diagrammatic manipulations of the diagrams of EGI. Now it remains to investigate how the calculus for EGIs can be transferred to a purely graphical calculus for diagrams of EGs.

Except the double cut rule, the remaining rules — erasure, insertion, iteration, deiteration — rely on the notation of a subgraph. Thus we have first to investigate how subgraphs can graphically be represented. We start this investigation with the subgraphs of EGIs, before the results of the investigation will be transferred to formal EGs.

The basic observation is that a subgraph behaves similarly to the enclosure of a cut, i.e., it can be understood as a subset of vertices, edges, and cuts, such that

- If a cut c belongs to the subgraph, then all elements which are enclosed by c belong to the subgraph as well, and
- The subgraph itself is placed in the area of a context.

This will give us the possibility to represent a subgraph similarly to the representation of a cut, that is, by a closed, doublepoint-free and smooth curve such that in the diagram of the graph, all elements of the diagram which denote an element of the subgraph are enclosed by the subgraph-line.[1]

Starting on page 128, it has been described how EGIs are graphically depicted. Recall the basic idea of the representation: We defined a quasiorder $\leq$ on the elements of the graph such with the following properties:

- we had $x \lneq c$ for an arbitrary element x and a context c iff x is enclosed by c, and
- $(Cut \cup \{\top\}, \leq)$ is a tree.

Due to the second condition, we can represent the cuts as closed, doublepoint-free and smooth curves which do not intersect or overlap, such that the cut-line of a cut c is enclosed by the cutline of a context d iff c is enclosed by d.

We adopt this idea to represent a subgraph of an EGI similarly to the representation of cuts by cut-lines. In order to do that, the mapping *area* of an EGI will be slightly extended such that it captures the subgraph as well. So let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa', \rho')$ be a subgraph of the EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ in the context $\top'$. Let $s \notin Cut \cup \{\top\}$ denote the subgraph. We define a mapping $\overline{area} : Cut \cup \{\top\} \cup \{s\} \to \mathfrak{P}(V \cup E \cup Cut \cup \{s\})$ as follows:

$$\overline{area}(c) := \begin{cases} \{s\} \cup area(\top') \cap (V' \cup E' \cup Cut') & \text{for } c = s \\ area(\top') \backslash (V' \cup E' \cup Cut') & \text{for } c = \top \\ area(c) & \text{else} \end{cases}$$

This mapping fulfills the same conditions we had for the mapping *area* for EGIs, i.e., we have:

a) $c_1 \neq c_2 \Rightarrow \overline{area}(c_1) \cap \overline{area}(c_2) = \emptyset$,

b) $V \cup E \cup Cut \cup \{s\} = \bigcup_{d \in Cut \cup \{\top\} \cup \{s\}} \overline{area}(d)$,

c) $c \notin \overline{area}^n(c)$ for each $c \in Cut \cup \{\top\} \cup \{s\}$ and $n \in \mathbb{N}$

[1] Recall that we use the term 'enclose' in two different meanings: On the mathematical level of EGIs, we had defined an enclosing-relation which had been derived from the mapping *area* in subgraphs, on the diagrammatic level of the representations, we said that an item of the diagram is enclosed by a cut-line if and only if it is placed in the inner region of this cut-line. The latter notation was introduced right after Def. 7.8, p. 69.

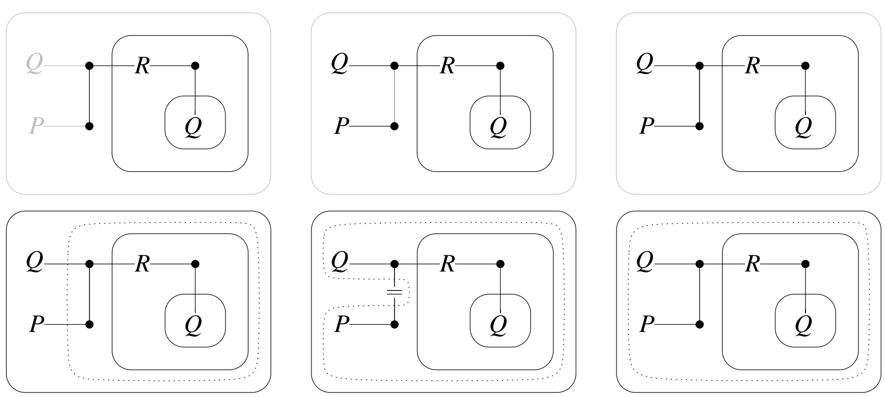
Analogously to Def. 7.3 for formal Alpha graphs resp. Def. 12.2 for EGIs, we can deduce from the mapping $\overline{area}$ a quasiorder on $V \dot\cup E \dot\cup \top \dot\cup \{s\}$ which is, reduced to $\top \dot\cup \{s\}$, a tree. In order to avoid confusion with the already defined order $\leq$ on $V \dot\cup E \dot\cup \top$, we will denote this new order with $\sqsubseteq$.

Now, analogously to the representation of EGIs based on the order $\leq$, we can now represent the EGI $\mathfrak{G}$ with the subgraph $\mathfrak{G}'$, based on the order $\sqsubseteq$. The only difference to a usual representation for $\mathfrak{G}$ is that we now have a further element s, indicating the subgraph, which is drawn similar to cuts as a smooth, doublepoint-free line. This line is called the SUBGRAPH-LINE of the subgraph $\mathfrak{G}'$. To distinguish it from the cut-lines, we agree to draw the subgraph-line a dotted manner. As we have $area(d) = \overline{area}(d)$ for each cut $c \neq \top'$, and as we have $area(\top') = \overline{area}(\top) \cup \overline{area}(s)$, the representation of $\mathfrak{G}$ based on $\sqsubseteq$ is obtained from an representation of $\mathfrak{G}$ based on $\leq$, where the subgraph-line of s is added to the area-space of $\top'$. Finally, due to he third and fourth condition of subgraphs (see Def. 12.10), we have:

$$x \in V' \cup E' \cup Cut' \quad \Longleftrightarrow \quad x \in \sqsubseteq[s] \text{ and } x \neq s \tag{21.1}$$

Thus the elements of the diagram which are enclosed by the subgraph-line denote exactly the elements of $\mathfrak{G}$ which belong to the subgraph.

To exemplify subgraph-lines, we consider again the examples for subgraphs given on page 135. To ease the comparison, in the first row, the representations of subgraphs from page 135 are repeated; in the second row, the subgraphs are indicated by subgraph-lines. It should be noted that in the second example, when the subgraph is indicated by a subgraph-line, it is necessary to represent the identity-link due to the usual convention for edges to make clear that the vertices incident to the edge belong to the subgraph, whilst the identity-edge does not.



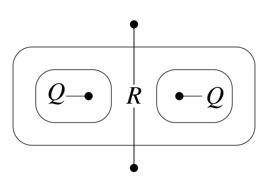
Starting on page 132, it has shortly been investigated which diagrams occur as diagrams of EGIs. It turned out that, as EGIs are based on relational graphs with dominating nodes, cut-lines have to satisfy the following condition: If a vertex-spot is enclosed by a cut-line, and if this spot is is connected to a relation-sign with an edge-line, then the relation-sign is enclosed by this cut-line, as well. In short: If a vertex spot is placed within a cut-line, then all incident relation signs are as well. For subgraphs, we have a similar restriction: If an edge belongs to a subgraph, then all incident vertices belong to the subgraph as well (otherwise a subgraph would not be an EGI). Thus we see that a properly drawn subgraph-line satisfies, roughly speaking, the following condition: If a relation sign as placed within a subgraph line, then all incident vertices are as well. Moreover, a properly drawn subgraph-line has to fulfill the usual conditions for entities in a diagram. So we can refine our (informal) definition of subgraph-lines as follows:

Informal Definition 21.1 (Subgraph-Lines for Diagrams of EGIs) *A* subgraph-line *(in the diagram of an EGI) is a closed, doublepoint-free and smooth curve, which does not touch or cross cut-lines, vertex-spots, or relation signs, which crosses no edge-line more than once, and which satisfies: If a relation-sign is enclosed by the subgraph-line, and if a vertex-spot is connected to the relation-sign with an edge-line, then the vertex-spot (and the edge-line) are enclosed by the subgraph-line as well. To distinguish subgraph-lines from cut-lines, they are drawn in a dotted manner.*

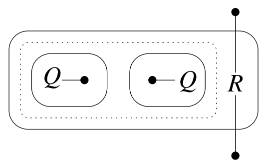
We have argued that each subgraph of an EGI can be indicated by a subgraph line. One the other hand, it is easy to see that if a subgraph-line is added to the diagram of an EGI, than the elements which are enclosed by the subgraph-line (more precisely: The elements of the EGI such that their representations are enclosed by the subgraph line) form a subgraph. To summarize our discussion:

> If a diagram of an EGI $\mathfrak{G}$ is given, and we add to the diagram a subgraph-line, then this subgraph-line indicates a (uniquely determined) subgraph of $\mathfrak{G}$. On the other hand, if $\mathfrak{G}'$ is a subgraph of $\mathfrak{G}$, then there exists a diagram of $\mathfrak{G}$ where $\mathfrak{G}'$ can be indicated by a subgraph-line in this diagram.

It is important to grasp the impact of the phrase '[...] then there exists a diagram of $\mathfrak{G}$[...]'. A subgraph $\mathfrak{G}'$ of $\mathfrak{G}$ can be indicated by a subgraph line in an *appropriately chosen* diagram. That is, if a diagram of $\mathfrak{G}$ is already given, it may happen the subgraph can not be indicated by a subgraph-line which is added to the diagram. Thus sometimes the graph $\mathfrak{G}$ has to represented in a different way, i.e., the diagram has to be redrawn, in order to indicate the subgraph by a subgraph-line. In order to see this, consider the following EGI:



The substructure containing both inner cuts and all what they contain is a subgraph. But in this representation, this subgraph cannot be indicated by a subgraph-line. We have to *redraw* the whole graph in a different way in order indicate the subgraph by a subgraph-line. This is done in the representation below.



EGs are classes of EGIs. For EGs, we have no notation of a subgraph. This is due to the fact that different EGIs representing the same EGs have different subgraphs. Nonetheless, for the graphical representation of EGs, we can adopt the notation of subgraph-lines. As we do not have vertex spots in diagrams of EGs, we set:

Informal Definition 21.2 (Subgraph-Lines for Diagrams of EGs) *A* subgraph-line *in the diagram of a formal existential graph is a closed, doublepoint-free and smooth curve, which does not touch or cross cut-lines, nor relation signs, and which does not touch heavily drawn lines (but it is allowed that heavily drawn lines are crossed). To distinguish subgraph-lines from cut-lines, they are drawn in a dotted manner. The part of the graph inside the subgraph-line is called a* subgraph *of the existential graph.*

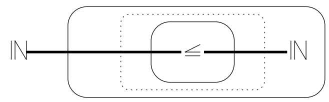
Of course, there each subgraph-line is directly enclosed either by the sheet of assertion of by a uniquely given cut-line. Analogously to the formal definition for EGIs, we will say that this is the context of the subgraph.

The definition of subgraphs in diagrams is not a mathematical definition, as already the diagrams are not mathematically defined. On the other hand, this informal definition is tightly connected to the (still informal) notation of subgraph-lines in diagrams of EGIs, and we have thoroughly investigated that these subgraph-lines correspond to the (formally defined) subgraphs in EGIs, so this informal definition is reasonable. Nonetheless, this deserves a deeper discussion. To be more precisely: As all rules of the calculus for EGIs (except double cut) depend on the notation of a subgraph, we have to investigate how the rules for EGIs carry over to diagrammatic rules in the diagrammatic representations of EGs.

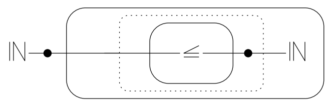
Let us, before we start this investigation, first collect some simple facts about subgraphs in diagrams of EGs.

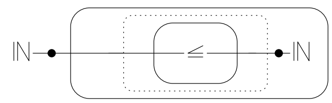
Recall how diagrams of EGIs are converted to diagrams of EGs. First of all, it is easy to see that if we have a diagram of an EGI $\mathfrak{G}$ with a subgraph line s, then this line is a subgraph line in the corresponding diagram of the corresponding EG $[\mathfrak{G}]_\sim$.

On the other hand, if a diagram of an EG $\mathfrak{E}$ with subgraph-line in is given, then we can find an EGI $\mathfrak{G}' \in \mathfrak{E}$ with a (not necessarily unique) given subgraph $\mathfrak{G}_0$, such that there is a diagram of $\mathfrak{G}'$ with a subgraph-line denoting $\mathfrak{G}_0$ which can be converted –including the subgraph-line– into the given diagram of $\mathfrak{E}$. This shall be discussed with a small example. Consider the following diagram of an EG (stating that there is a minimal natural number) with a subgraph-line.

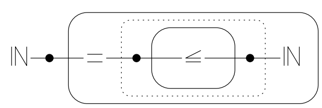


Below, you find a two diagrams of an EGI representing the given EG.





Please note that in the left diagram, the dotted line is *not* a subgraph-line for the EGI. Recall that if a relation-sign is enclosed by the subgraph-line, and if a vertex-spot is connected to the relation-sign with an edge-line, then the vertex-spot has to be enclosed by the subgraph-line as well. This does not hold for none of the two vertices in the left diagram. The right diagram is better drawn for our purpose, thus this condition is respected for the right vertex. But there is no way to redraw the right diagram such that the condition can be fulfilled for the left vertex as well. But the transformation rules for ligatures allows to insert additional vertices. By doing so, we can obtain a graph $\mathfrak{G}'$ as described above. Below, an example is given.



To summarize: If the diagram of an EGI with a subgraph-line is given, then in the corresponding diagram of the corresponding EG, this subgraph-line is still a subgraph line. Vice versa, if the diagram of an EG with a subgraph-line is given, there exists an EGI representing that EG such that this EGI has an diagram –including a subgraph-line– which corresponds to the diagram of the EG. The existence of an EGI which represents the EG is sufficient for the intended investigation how the calculus for EGIs can be transferred to a purely graphical calculus for the diagrams of EGs: Recall that an EG $\mathfrak{E}_a$ entails an EG $\mathfrak{E}_b$ if there exist EGIs $\mathfrak{G}_a$, $\mathfrak{G}_b$, with $\mathfrak{E}_a = [\mathfrak{G}_a]_\sim$, $\mathfrak{E}_b = [\mathfrak{G}_b]_\sim$ and $\mathfrak{G}_a \vdash \mathfrak{G}_b$.

Now we can finally investigate how this calculus for EGIs can be transferred to a purely graphical calculus for diagrams of EGs. In fact, due to the discussion so far, we could simply adopt the rules for EGIs as rules for the diagrams of EGs. But recall that in the rules for EGIs, we had to distinguish between subgraphs and *closed* subgraphs. The iteration and deiteration rule work fine with subgraphs, but as erasing or inserting a non-closed subgraph from resp. into an EGI would lead to a non well-formed EGI, thus it is only allowed to erase or insert *closed]* subgraphs. For diagrams of EGs, this distinction is not needed. For this reason, the erasure and insertion rule are, given as graphical transformation rules for the diagrams of EGs, simplified. The graphical transformation rules can be described as follows:

- **Erasure:** Let a diagram of an EG be given with a subgraph line s in a positive context. Then s and all what is scribed inside s can be erased. This operation includes the right to cut heavily drawn lines, where they cross s. Moreover, this operation includes the right to erase parts of a heavily drawn line in a positive context.
- **Insertion**: Let a diagram of an EG be given with a cut-line cl of a negative cut c. Then the diagram D_0 of an arbitrary graph may be scribed inside the area-space of cl. This operation includes the right to connect points on heavily drawn lines of the inserted graph, which are directly placed on the sheet of assertion of $\mathfrak{G}_o$, with points on heavily drawn lines placed in the area-space of cl.
- **iteration:**
  - Let a diagram of an EG be given with a subgraph $\mathfrak{G}_0$ (indicated by a subgraph line) in a context c (the sheet of assertion; or a cut, indicated by a cut-line) and let d be a context which is identical to c or enclosed by c, and which does not belong to $\mathfrak{G}_0$. Then a copy of $\mathfrak{G}_0$ may be scribed on the area-space of d. In this transformation, the following is allowed: If we have a heavily drawn point p of $\mathfrak{G}_0$ from which a heavily drawn line inwardly (particularly, it crosses no cut-line more than once) to a point in d, then this point in d may be connected with the copy of p.
  - It is allowed to add new branches to a ligature, or to extend any line of identity inwardly through cuts.
- **deiteration:** If $\mathfrak{G}_0$ is a subgraph of $\mathfrak{G}$ which could have been inserted by rule of iteration, then it may be erased.
- **double cuts:** Double cuts, i.e. two cut-lines c_1, c_2 such that there is nothing between them, except lines of identity (heavily drawn lines with no branches and which do not cross and cuts) which start on c_1 and end on c_2, may be inserted into or erased from any diagram of an EG.
- **erasing a vertex:** An isolated vertex spot may be erased from arbitrary contexts.
- **inserting a vertex** An isolated vertex spot may be inserted in arbitrary contexts.

All rules –except erasure and insertion– are informally descriptions of the formally defined rules for EGIs, adopted for diagrams of EGs. For this reason, each application of a formal rule corresponds to an application of the corresponding graphical rule, and vice versa. Thus it remains to discuss how the graphical erasure rule and insertion rule are related to the formal erasure rule and insertion rule for EGIs. To be more precisely: We have to show that each application of a formal rule for EGIs is reflected by the diagrammatic rules, and vice versa.

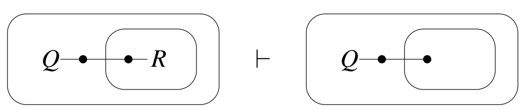
As the rules are dual to each other, it is sufficient to discuss the erasure-rule. Let us recall this rule for EGIs: In positive contexts, any directly enclosed edge, isolated vertex, and closed subgraph may be erased.

Let us first assume that we have an EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ with a closed subgraph $\mathfrak{G}_0$, which is erased from a positive cut c. There exists a diagram of $\mathfrak{G}$, where $\mathfrak{G}_0$ is indicated by a subgraph line s. This subgraph line does not cross any other graphical items of diagram. Obviously, s is a subgraph line in the corresponding diagram of the corresponding EG $[\mathfrak{G}]_\sim$, and erasing $\mathfrak{G}_0$ corresponds to erasing the part of the diagram of $[\mathfrak{G}]_\sim$ what is written inside s. Thus we see that this application of the formal erasure-rule is reflected by the diagrammatic erasure rule for diagrams of EGs.

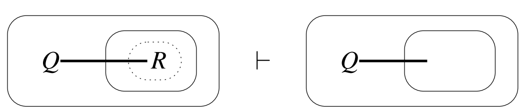
As an isolated vertex is a closed subgraph, this argument holds for the erasure of an isolated vertex from $\mathfrak{G}$ as well.

Let us finally assume that an edge e in c is erased from $\mathfrak{G}$. This case is a little bit tricky, so let us first ease the discussion of this case, we assume that e is an 1-ary edge, i.e. we have $\nu(e) = v$ for a $v \in V$. We have to distinguish the cases $ctx(e) = ctx(v)$ and $ctx(e) < ctx(v)$. The latter case is *not* reflected by the diagrammatic erasure rule for diagrams of EGs: We need furthermore an application of the graphical deiteration rule. This shall be exemplified with a simple example. We consider two different EGIs which represent the same EG. In both EGS, the edge labeled with R is erased.

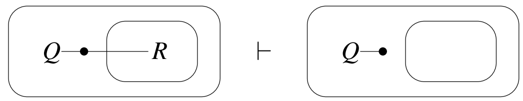
Consider first the following EGIs, where the second EGI is obtained from the first EGI by erasing the edge e labeled with R.



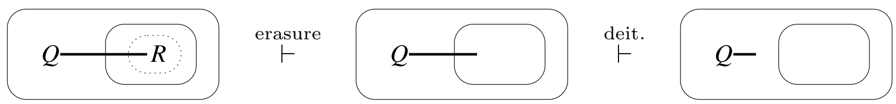
This application of the rule on EGIs has the following graphical counterpart for the corresponding EGs:



This works fine as the vertex v connected to e is placed in the same cut, i.e., we had $ctx(e) = ctx(v)$. Let us now consider an example where this condition is violated.



Note that this application of the erasure rule on EGIs is *not directly* reflected by the graphical erasure rule. Instead, we need an additional application of the graphical deiteration rule, which allows to retract heavily draw lines outwards through cuts. Thus, on the level of diagrams of EGs, erasing the edge is carried out in two steps as follows:

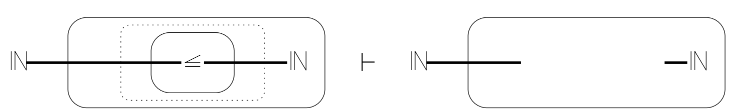


The idea behind this example can easily be lifted to the erasure of an n-ary edge $e = (v_1, \ldots, v_n)$ from an EGI $\mathfrak{G}$: On the level on the diagrams of EGs, this erasure is reflected by an application of the graphical erasure rule, followed by an application of the graphical deiteration rule for each i with $ctx(v_i) > ctx(e)$.[2]

So far we have seen that an application of the erasure rule for a formal EGI $\mathfrak{G}$ is reflected by an application of the graphical erasure rule, possibly followed by some application of the graphical deiteration rule, for the diagram of the corresponding EG $[\mathfrak{G}]_\sim$.

Let us now discuss the inverse direction, i.e., we have two diagrams of EGs $\mathfrak{E}_a, \mathfrak{E}_b$ where the diagram of $\mathfrak{E}_b$ is obtained from the diagram of $\mathfrak{E}_a$ by an application of the graphical erasure rule. That is, in the diagram of $\mathfrak{E}_a$, we have a subgraph-line s indicating a subgraph, and the diagram of $\mathfrak{G}_b$ is obtained from the diagram of $\mathfrak{G}_b$ by erasing s and what is scribed inside s.

In the erasure rule for EGIs, only *closed* subgraphs can be erased. If the subgraph line s in the diagram of $\mathfrak{E}_a$ does not cross any heavily drawn line, then it corresponds to a subgraph line of a closed subgraph $\mathfrak{G}_0$ of an EGI $\mathfrak{G}_a$ with $[\mathfrak{G}_a]_\sim = \mathfrak{E}_a$, thus in this case, the application of the graphical erasure rule corresponds to an application of the formal erasure rule for EGIs. So we have to discuss the case where s might cross heavily drawn lines. The basic idea for this case has already been discussed on the end of Sec. 14.1, page 151: Each time s crosses a heavily drawn line, the line is broken in this place with the erasure rule. This shall be exemplified with the EG from above. Consider the following application of the graphical erasure rule

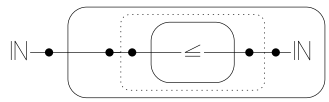


(21.2)

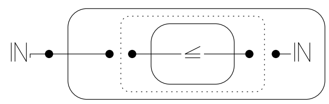
We have to find EGIs $\mathfrak{G}_a$, $\mathfrak{G}_b$ such that the left diagram is a diagram of $[\mathfrak{G}_a]_\sim$, the right diagram is a diagram of $[\mathfrak{G}_b]_\sim$, and we have $\mathfrak{G}_a \vdash \mathfrak{G}_b$.

[2] This holds even if we have a vertex v which is attached more than once to e.

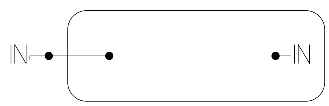
The heavily drawn lines in the diagrams of EGs correspond to vertex-spots and edge-lines in EGIs. Due to the transformation rules for ligatures, we can assume that we have an EGI $\mathfrak{G}_a$ such that if s crosses an edge-line in its graphical representation (which can be transformed into a graphical representation for $[\mathfrak{G}_a]_\sim$), then this edge-line is the edge-line of an identity edge between two vertices such that both vertices are placed in the context of the subgraph and such that in the graphical representation of the EGI, one vertex-spot is placed inside and the other vertex is placed outside the subgraph-line. For our example, we can choose $\mathfrak{G}_a$ as follows:



(In the diagram, I omitted to label the identity edges with the relation name '$\doteq$'). Now we can erase the edges of which the edge-lines in the diagram cross the subgraph-line with the erasure rule for EGIs. We obtain:



Now the subgraph-line indicates a *closed* subgraph, which can be erased with the formal erasure rule. We get:



This is the desired EGI $\mathfrak{G}_b$, as the diagram of $[\mathfrak{G}_b]_\sim$ is the right diagram of Eqn. (21.2).

The idea of this example applies to arbitrary diagrams of EGs. That is: If we have two diagrams of two EGs $\mathfrak{E}_a, \mathfrak{E}_b$ where the diagram of $\mathfrak{E}_b$ is obtained from the diagram of $\mathfrak{E}_a$ by an application of the graphical erasure rule, then we can find EGIs $\mathfrak{G}_a, \mathfrak{G}_b$ with $[\mathfrak{G}_a]_\sim = \mathfrak{E}_a$, $\mathfrak{E}_b = [\mathfrak{G}_b]_\sim$, and we can derive $\mathfrak{G}_b$ from $\mathfrak{G}_a$ by first erasing some identity edges and then a closed subgraph with the formal erasure rule. Moreover, it shows that the graphical erasure of a part of a heavily drawn line in a positive context corresponds to the formal erasure of an identity edge in an accordingly chosen EGI.

So far we have elaborated how the formal erasure rule for EGIs and the graphical erasure rule are related to each other. The insertion rule is simply the inverse direction of the erasure rule, restricted to negative contexts (in contrast to the restriction to a positive context in the erasure rule). For this reason, the discussion can be applied for the insertion-rule as well. That is: An application of the formal insertion-rule for EGIs corresponds to an an application of the graphical insertion rule, in the case of inserting an edge possibly preceeded by some applications of the graphical iteration rule. Vice versa, an application of the graphical insertion rule corresponds to an application of the formal insertion rule (for closed subgraphs), possibly followed by inserting some identity edges.

As already mentioned, the graphical rules iteration, deiteration and double cut for the diagrams of existential graphs correspond directly to the formally defined rules for EGIs. As we now have worked out all rules, including the important notation of a subgraph, in a graphical manner, we finally have developed a purely diagrammatic logic system.

Extending the System

22

Overview

In the last chapters, Peirce's system of existential graphs has been intensively investigated, and they have been developed as a diagrammatic version of first order logic.

In the common symbolic formalizations of first order logic, the formulas formalize statements about objects, relations, and functions. But as mentioned, Peirce did not incorporate objects names or function names into EGs, i.e., he treated all names in EGs as names for predicates (see Sect. 11.2). The next two chapters show how the syntax, semantics, and calculus for EGIs have to be extended in order to have names for objects and functions as well. As functions are special relations, the approach for function names is on the side of the syntax straight forward: Besides relation names, edges can be labeled with function names as well. names, two different accounts are possible: We can assign object names to edges or to vertices. In Chpt. 24, we investigate how the systems of EGIs has to be extended when we assign all types of names to edges. In Chpt. 23, edges are labeled with relation- function names, but the object names are assigned to vertices. The graphs of of this chapter will be therefore called vertex-based EGIs.

Even if we incorporate object- and function names, EGIs and vertex-based EGIs are still formalizations of judgments. That is, in a given relational structure $\mathcal{M}$, a graph $\mathfrak{G}$ evaluates to true ($\mathtt{tt}$), which was denoted as usual $\mathcal{M} \models \mathfrak{G}$, or to false ($\mathtt{ff}$). In Chpts. 25, the system of EGIs is extended by a syntactical device which corresponds to free variables of $\mathcal{FO}$. The resulting graphs are not evaluated to formulas, but to relations instead, and are therefore called *relation graph instances* (RGIs). We will consider 0-ary relations as well, and as there are exactly two 0-ary-relations, which can naturally identified with the truth values $\mathtt{tt}$ and $\mathtt{ff}$, RGIs are an extension of EGIs.

It is well-known that Peirce's extensively investigated a *logic of relations* (which he called 'relatives'). Much of the third volume of the collected papers is dedicated to this topic (see for example "Description of a Notation for the Logic of Relatives, Resulting From an Amplification of the Conceptions of Boole's Calculus of Logic" (3.45–3.149, 1870) "On the Algebra of Logic" (3.154–3.251, 1880), "Brief Description of the Algebra of Relatives" (3.306–3.322, 1882), and "the Logic of Relatives" (3.456–3.552, 1897)). As Burch writes, in Peirce's thinking 'reasoning is primarily, most elementary, reasoning about *relations*' ([Bur91a], p. 2, emphasis by Burch). Burch elaborated in his book 'A Peircean Reduction Thesis' ([Bur91a]) a prototypic version of Peirce's algebra of relations, termed *Peircean Algebraic Logic (PAL)*. The development of PAL is driven to a large extent by the form of EGs. Thus, in contrast to EGIs with object- and function names, RGIs are still in the line of reconstructing Peirce's historical diagrammatic logic.

Finally, in Chpt. 26, a version of Peirce's famous reduction thesis for relation graphs is provided. Roughly speaking, for relations the reduction thesis says that ternary relations suffice to construct arbitrary relations, but that not all relations can be constructed from unary and binary relations alone. A strong version of this claim has recently be proven by Hereth-Correia and Pöschel in [HCP06]. This result will be transfered in Chpt. 26 to relation graphs.

In the formal development of existential graphs, we first defined existential graph *instances* (EGIs), and then defined formal existential graphs as classes of EGIs. Two EGIs are in the same class if they can be transformed into each other with the four transformation rules for ligatures. The introduction so far indicates that we follow the same path in the next chapters. Indeed, all investigations in the following chapters are carried out on different form of graph instances (EGIs, vertex-based EGIs and RGIs over an alphabet with object-, function- and relation-names). For all of these classes, we have a sound and complete calculus which particularly contains the transformation rules for ligatures. Thus for all of these classes, we can canonically define the corresponding formal graphs (formal existential graphs, formal vertex-based existential graphs, and formal relation graphs over an extended alphabet). For this reason, we omit to explicitely define the different forms of formal graphs.

Particularly, the conventions for the diagrammatic representations of formal existential graphs are adopted for the other classes of formal graphs. For formal vertex-based existential graphs, an additional discussion is needed. In vertex-based EGIs, vertices are additionally labeled. For representing these vertices, two possible conventions are provided. But it turns out that only one of these conventions is suited for diagrammatically representing formal vertex-based existential graphs.

23

Adding Constants and Functions

In this (and the next) chapter, it shall be shown how constants and function names can be added to the system of existential graphs.

An n-ary function F is an n-ary relation which satisfies a specific property, namely: For each n objects $o_1, \ldots, o_{n-1}$ exists exactly one object o_n with $F(o_1, o_2, \ldots, o_{n-1}, o_n)$. So, functions can be understood as special relations. But Please note that we adopt the arity of relations for functions. That is, in our terminology, an n-ary function assigns a value to $n-1$ arguments. This understanding of the arity of a function is not the common one, but it will ease the forthcoming notation.

Analogously, even an object o can be understood as a special relation, namely the relation $\{(o)\}$. That is: objects correspond to unary relations which contain exactly one element (or as functions with zero arguments).

For these reasons, it is self-suggesting to employ constants and function names as special relation names. So far, we had considered EGIs and EGs which are based on an alphabet as defined in Def. 12.6. In this definition, only names for relations are introduced. We first extend the definition of an alphabet:

Definition 23.1 (Alphabet with Constants, Functions and Relations). *An* ALPHABET *is a structure* $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *of* CONSTANT NAMES*,* FUNCTION NAMES *and* RELATION NAMES*, resp., together with an arity-function* $ar : \mathcal{F} \cup \mathcal{R} \to \mathbb{N}$ *which assigns to each function name and relation name its arity. To ease the notation, we set* $ar(C) = 1$ *for each* $C \in \mathcal{C}$*. We assume that the sets* $\mathcal{C}, \mathcal{F}, \mathcal{R}$ *are pairwise disjoint. The elements of* $\mathcal{C} \dot{\cup} \mathcal{F} \dot{\cup} \mathcal{R}$ *are the* NAMES *of the alphabet. Let* $\doteq \in \mathcal{R}_2$ *be a special name which is called* IDENTITY.

In EGIs, thus in EGs, the edges had so far been labeled with relation names (see Def. 12.7). Now, we allow constants and function names as labels for edges as well. That is, in the definition of existential graph instances, we have to modify the condition for the labeling function κ. This is done as follows:

Definition 23.2 (Existential Graph Instance over $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$). *An* EXISTENTIAL GRAPH INSTANCE (EGI) OVER AN ALPHABET $\mathcal{A} = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *is a structure* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *where*

- $(V, E, \nu, \top, Cut, area)$ *is a relational graph with cuts and dom. nodes, and*
- $\kappa : E \to \mathcal{C} \dot\cup \mathcal{F} \dot\cup \mathcal{R}$ *is a mapping such that* $|e| = ar(\kappa(e))$ *for each* $e \in E$.

Similar to Def. 12.7, the system of all (extended) EGIs over $\mathcal{A}$ *will be denoted by* $\mathcal{EGI}^{\mathcal{A}}$ *(it will be clear from the used alphabet whether we consider EGIs as defined in Def. 12.7 or in this definition).*

Of course, we have to modify the semantics for EGIs as well: n-ary function names have to interpreted as n-ary functions over the universe of discourse, and constants have to be interpreted as objects in the universe of discourse. That is, the models for the herein defined EGIs are as follows:

Definition 23.3 (Relational Structures over $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$). *A* RELATIONAL STRUCTURE *or* RELATIONAL MODEL OVER AN ALPHABET $\mathcal{A} = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *is a pair* $\mathcal{M} := (U, I)$ *consisting of a nonempty* UNIVERSE U *and a function* $I := I_{\mathcal{C}} \cup I_{\mathcal{F}} \cup I_{\mathcal{R}}$ *with*

1. $I_{\mathcal{C}} : \mathcal{C} \to U$,
2. $I_{\mathcal{F}} : \mathcal{F} \to \bigcup_{k \in \mathbb{N}} \mathfrak{P}(U^k)$ *is a mapping such that for each* $F \in \mathcal{F}$ *with* $ar(F) = k$, $I(F) \in U^k$ *is (total) function* $I(F) : U^{k-1} \to U$, *and*
3. $I_{\mathcal{R}} : \mathcal{R} \to \bigcup_{k \in \mathbb{N}} \mathfrak{P}(U^k)$ *is a mapping such that for each* $R \in \mathcal{F}$ *with* $ar(R) = k$, $I(R) \in U^k$ *is a relation. The name '$\doteq$' is mapped to the identity relation on* U.

With the understanding that objects u (by implicitly identifying an object $u \in U$ with the unary relation $\{(u)\}$) and functions are special relations, the models we have just defined are a restriction of the models of Def. 13.1, where each name of arity n is mapped to an arbitrary n-ary relation.

Of course, when considering constants and function names, we have new entailments between graphs. For example, if C is a constant, the empty sheet of assertion (semantically) entails the graph $\bullet\!\!-\!\!\!-\, C$. Thus it must be possible to derive this graph from the empty sheet of assertion.

The new entailments must be reflected by the calculus, thus the calculus has to be extended in order to capture the specific properties of constants and functions. There are basically two approaches: Firstly, we can add axioms, secondly, we can add new rules to the calculus. Besides the empty sheet of assertion, Peirce's calculus for existential graphs has no axioms. To preserve this property, we will adopt he second approach.

23.1 General Logical Background

Before the new rules for constants and function names are introduced, this sections aims to describe the methodology how this shall be done.

As already mentioned, constants and function names can be understood as relation names which are mapped to relations with specific properties. If we have an alphabet $\mathcal{A}' = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ with constants and function names, we can then consider the alphabet $\mathcal{A} := (\mathcal{C} \dot{\cup} \mathcal{F} \dot{\cup} \mathcal{R}, ar)$, where each name is now understood as relation name. In this understanding, each EGI over $\mathcal{A}'$ is an EGI over $\mathcal{A}$ as well. Moreover, if $\mathcal{M}' := (U, I)$ with $I' := I'_{\mathcal{C}} \cup I'_{\mathcal{F}} \cup I'_{\mathcal{R}}$ is relational structure over the alphabet $\mathcal{A}'$, then $\mathcal{M} := (U, I)$ with

1. $I(F) := I'_{\mathcal{F}}(F)$ and $I(R) := I'_{\mathcal{R}}(R)$ for each $F \in \mathcal{F}$ and each $R \in \mathcal{R}$, and
2. $I(C) := \{(I'_{\mathcal{C}}(C)\}$

is the corresponding model (model with respect to Def. 13.1) over the alphabet $\mathcal{A}$. We implicitly identify $\mathcal{M}$ and $\mathcal{M}'$. Due to this convention, each model over $\mathcal{A}'$ is an model over $\mathcal{A}$ as well. But the models for $\mathcal{A}'$ form a subclass of the models for $\mathcal{A}$. That is, if we denote the models for $\mathcal{A}'$ with $\mathfrak{M}_2$ and the models for $\mathcal{A}$ with $\mathfrak{M}_1$, we have $\mathfrak{M}_2 \subsetneq \mathfrak{M}_1$.

We therefore have to deal with two classes of models. This yields two entailment relations as well. In the following, if $\mathfrak{H}$ is a set of EGIs and if $\mathfrak{G}$ is an EGI such that $\mathcal{M} \models \mathfrak{G}$ for each relational structure $\mathcal{M} \in \mathfrak{M}_i$ with $\mathcal{M} \models \mathfrak{G}'$ for each $\mathfrak{G}' \in \mathfrak{H}$, we write $\mathfrak{H} \models_i \mathfrak{G}$.

In the Beta part of this treatise, EGs have been evaluated in $\mathfrak{M}_1$, and we had obtained a sound and complete calculus. In the following, this calculus shall be denoted by $\vdash_1$. (This use of the symbol '$\vdash$' is a little bit sloppy: Usually, the symbol denotes the syntactical entailment relation between formulas of a given logic, which is *obtained* from a set of rules. We will use '$\vdash$' in this sense as well, but the set of rules shall also be denoted with '$\vdash$'. It will be clear from the context which use of '$\vdash$' is intended.) The soundness and completeness of $\vdash_1$ can be now restated as follows: If $\mathfrak{H} \cup \{\mathfrak{G}\}$ is a set of EGIs over $\mathcal{A}$, we have

$$\mathfrak{H} \vdash_1 \mathfrak{G} \quad \Longleftrightarrow \quad \mathfrak{H} \models_1 \mathfrak{G} \tag{23.1}$$

We seek a calculus $\vdash_2$ which extends $\vdash_1$ (that is, $\vdash_2$ has new rules, which will informally be denoted by $\vdash_2 \supseteq \vdash_1$) and which is sound and complete with respect to $\mathfrak{M}_2$.

The calculus $\vdash_1$, and hence $\vdash_2$ as well, encompasses the 5 basic-rules of Peirce. Thus for both calculi, the deduction theorem (see Thm. 8.7) holds, i.e., for $i = 1, 2$, we have

$$\mathfrak{G}_a \vdash_i \mathfrak{G}_b \quad \Longleftrightarrow \quad \vdash_i \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}} \tag{23.2}$$

We will extend $\vdash_1$ to $\vdash_2$ as follows: First of all, the new rules in $\vdash_2$ have to be sound. Then for a set of graphs $\mathfrak{H}$ and an EGI $\mathfrak{G}$ we have

$$\mathfrak{H} \vdash_2 \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \models_2 \mathfrak{G} \tag{23.3}$$

On the other hand, let us assume that for each $\mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2$, there exists a graph $\mathfrak{G}_{\mathcal{M}}$ with

$$\vdash_2 \mathfrak{G}_{\mathcal{M}} \quad \text{and} \quad \mathcal{M} \not\models \mathfrak{G}_{\mathcal{M}} \tag{23.4}$$

If the last two assumptions (23.3) and (23.4) hold, we obtain that $\vdash_2$ is an adequate calculus, as the following theorem shows.

Theorem 23.4 (Completeness of $\vdash_2$). *A set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of EGIs over an alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ satisfies*

$$\mathfrak{H} \models_2 \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \vdash_2 \mathfrak{G}$$

Proof: Let $\mathfrak{H}_2 := \{\mathfrak{G}_{\mathcal{M}} \mid \mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2\}$. From (23.3) we conclude: $\models_2 \mathfrak{G}_{\mathcal{M}}$ for all $\mathfrak{G}_{\mathcal{M}} \in \mathfrak{H}_2$. Now (23.4) yields:

$$\mathfrak{M}_2 = \{\mathcal{M} \in \mathfrak{M}_1 \mid \mathcal{M} \models \mathfrak{G} \text{ for all } \mathfrak{G} \in \mathfrak{H}_2\} \tag{23.5}$$

Now let $\mathfrak{H} \cup \{\mathfrak{G}\}$ be an arbitrary set of graphs. We get:

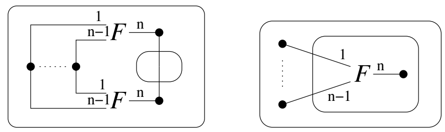
$$\begin{aligned}
\mathfrak{H} \models_2 \mathfrak{G} \overset{\text{Def}}{\Longleftrightarrow}\ & \text{f.a. } \mathcal{M} \in \mathfrak{M}_2 : \text{ if } \mathcal{M} \models \mathfrak{G}' \text{ for all } \mathfrak{G}' \in \mathfrak{H}, \text{ then } \mathcal{M} \models \mathfrak{G} \\
\overset{(23.5)}{\Longleftrightarrow}\ & \text{f.a. } \mathcal{M} \in \mathfrak{M}_1 : \text{ if } \mathcal{M} \models \mathfrak{G}' \text{ for all } \mathfrak{G} \in \mathfrak{H}_2 \cup \mathfrak{H}, \text{ then } \mathcal{M} \models \mathfrak{G} \\
\Longleftrightarrow\ & \mathfrak{H} \cup \mathfrak{H}_2 \models_1 \mathfrak{G} \\
\overset{(23.1)}{\Longleftrightarrow}\ & \mathfrak{H} \cup \mathfrak{H}_2 \vdash_1 \mathfrak{G} \\
\Longleftrightarrow\ & \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
& \mathfrak{G}_1 \quad \mathfrak{G}_2 \quad \ldots \quad \mathfrak{G}_n \quad \mathfrak{G}'_1 \quad \mathfrak{G}'_2 \quad \ldots \quad \mathfrak{G}'_m \quad \vdash_1 \mathfrak{G} \\
\overset{(23.2)}{\Longleftrightarrow}\ & \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
& \vdash_1 \boxed{\mathfrak{G}_1\ \mathfrak{G}_2 \ldots\ \mathfrak{G}_n\ \mathfrak{G}'_1\ \mathfrak{G}'_2 \ldots\ \mathfrak{G}'_m\ \boxed{\mathfrak{G}_b}} \\
\overset{\vdash_2 \supseteq \vdash_1,\ (23.4)}{\Longrightarrow}\ & \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
& \vdash_2 \mathfrak{G}'_1 \ldots\ \mathfrak{G}'_m\ \boxed{\mathfrak{G}_1\ \mathfrak{G}_2 \ldots\ \mathfrak{G}_n\ \mathfrak{G}'_1\ \mathfrak{G}'_2 \ldots\ \mathfrak{G}'_m\ \boxed{\mathfrak{G}_b}} \\
\overset{\text{deit.}}{\Longleftrightarrow}\ & \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
& \vdash_2 \mathfrak{G}'_1 \ldots\ \mathfrak{G}'_m\ \boxed{\mathfrak{G}_1\ \mathfrak{G}_2 \ldots\ \mathfrak{G}_n\ \boxed{\mathfrak{G}_b}} \\
\overset{\text{era.}}{\Longrightarrow}\ & \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \vdash_2 \boxed{\mathfrak{G}_1 \ldots\ \mathfrak{G}_n\ \boxed{\mathfrak{G}_b}} \\
\overset{(23.2)}{\Longleftrightarrow}\ & \text{there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \vdash_2 \mathfrak{G} \\
\overset{\text{Def.}}{\Longrightarrow}\ & \mathfrak{H} \vdash_2 \mathfrak{G}
\end{aligned}$$

□

23.2 Extending the Calculus

In this section, the calculus is extended in order to capture the specific properties of constants and functions. We start the scrutiny with functions.

The following EGI holds in a model (U, I) exactly if F is interpreted as an n-ary (total) function $I(F) : U^{n-1} \to U$:

$\mathfrak{G}_F :=$ 

More precisely: The left subgraph is satisfied if F is interpreted as partial function (that is, to objects $o_1, \ldots, o_{n-1}$ exist at most one o_n with $I(F)(o_1, \ldots, o_n)$), the right subgraph is satisfied if for objects $o_1, \ldots, o_{n-1}$ exist at least one o_n with $I(F)(o_1, \ldots, o_n)$. In other words: The left subgraph guarantees the uniqueness, the right subgraph the existence of function values.

According to the last subsection, we have to find rules which are sound and which enable us to derive each graph $\mathfrak{G}_F$ with $F \in \mathcal{F}$.

Definition 23.5 (New Rules for Function Names). *The calculus for existential graph instances over the alphabet $\mathcal{C} \dot{\cup} \mathcal{F} \dot{\cup} \mathcal{R}$ consists of all rules of Defs. 12.14 and 15.2. Moreover, if $F \in \mathcal{F}$ is an n-ary function name, then the following additional transformations may be performed:*

- **Functional Property Rule (uniqueness of values)**

  Let e and f be two n-ary edges with $\nu(e) = (v_1, \ldots, v_{n-1}, v_e)$, $\nu(f) = (v_1, \ldots, v_{n-1}, v_f)$,[1] $ctx(e) = ctx(v_e)$, $ctx(f) = ctx(v_f)$,[2] and $\kappa(e) = \kappa(f) = F$. Let c be a context with $c \leq ctx(e)$ and $c \leq ctx(f)$. Then arbitrary identity-links id with $\nu(id) = (v_e, v_f)$ may be inserted into c or erased from c.

- **Total Function Rule (existence of values)**

  Let $v_1, \ldots, v_{n-1}$ be vertices, let c be a context with $c \leq ctx(v_1), \ldots, ctx(v_{n-1})$. Then we can add a vertex v_n and an edge e to c with $\nu(e) = (v_1, \ldots, v_n)$ and $\kappa(e) = F$. Vice versa, if v_n and e are a vertex and an edge in c with $\nu(e) = (v_1, \ldots, v_n)$ and $\kappa(e) = F$ such that v_n is not incident with any other edge, e and v_n may be erased.

---

[1] In Def. 12.1, we defined $v_e := v$ for edges e with $\nu(e) = \{(v)\}$. The notation v_e and v_f can be understood to be a generalization of Def. 12.1.

[2] The conditions $ctx(e) = ctx(v_e)$, $ctx(f) = ctx(v_f)$ had been added for sake of convenience, and to keep this calculus closely related to the forthcoming calculus in the next chapter. But the proof of the soundness of this rule does not need this conditions, i.e., they could be dismissed. This applies to the forthcoming 'constant identity rule' as well.

We have to show that these rules are sound are complete. We start with the soundness of the rules.

Lemma 23.6 (The Total Function Rule is Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two EGIs over $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ with the total function rule, then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by adding a vertex v_n and an edge e to c according to the total function rule. We want to apply Lemma 13.8 to c, so let ref be a valuation for the context c.

Let us first assume that we $\mathcal{M} \models \mathfrak{G}[c, ref]$. That is, there is an extension $\overline{ref}$ of ref to $V \cap area(c)$ with $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}]$. Let $o := I(F)(ref(v_1, \ldots, ref(v_n))$. Then $\overline{ref}\,' := \overline{ref} \cup \{(v_n, o)\}$ is a extended partial valuation for c in $\mathfrak{G}'$ which obviously satisfies $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}\,']$, as the additional edge condition for e in the context c of $\mathfrak{G}'$ holds due to the definition of $\overline{ref}\,'$. Particularly, we obtain $\mathcal{M} \models \mathfrak{G}'[c, ref]$.

Now let us assume $\mathcal{M} \models \mathfrak{G}'[c, ref]$. That is, there is an extension $\overline{ref}\,'$ of ref to $V \cap area(c)$ with $\mathcal{M} \models \mathfrak{G}'[c, \overline{ref}\,']$. Let $\overline{ref} := \overline{ref}\,' \backslash \{(v_n, \overline{ref}\,'(v_n))\}$ we obviously have $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}]$, thus we conclude $\mathcal{M} \models \mathfrak{G}[c, ref]$.

Now Lemma 13.8 yields the lemma. □

Lemma 23.7 (The Functional Property Rule is Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two EGIs over $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ with the functional property rule, then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}'$ by inserting an identity-link id with $\nu(id) = (v_e, v_f)$ into c. We set $c_e := ctx(e)$ and $c_f := ctx(f)$. The EGIs $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for the context c. First note that the contexts c_e and c_f must be comparable. W.l.o.g. we assume $c_e \geq c_f \geq c$.

We first consider the case $c_e = c_f = c$. We want to apply Lemma 13.8 to c, so let ref_c be a partial valuation for c. In $\mathfrak{G}'$ in the context c, we have added the edge id, thus for c, there is one more edge condition to check. So it is sufficient to show that

$$(U, I) \models \mathfrak{G}[c, ref_c] \quad \Longrightarrow \quad (U, I) \models \mathfrak{G}'[c, ref_c] \tag{23.6}$$

holds. So let $(U, I) \models \mathfrak{G}[c, ref_c]$. That is, there is an extension $\overline{ref_c}$ of ref_c to $V \cap area(c)$ with $\mathfrak{G} \models \mathfrak{G}[c, \overline{ref_c}]$, i.e., $\overline{ref_c}$ satisfies all edge- and cut-conditions in c. Particularly, it satisfies the edge-conditions for e and f, that is:

$$\begin{array}{ll} (\overline{ref_c}(v_1), \ldots \overline{ref}(v_{n-1}), \overline{ref_c}(v_e)) \in I(\kappa(e)) & \text{and} \\ (\overline{ref_c}(v_1), \ldots \overline{ref}(v_{n-1}), \overline{ref_c}(v_f)) \in I(\kappa(f)) & \end{array}$$

i.e.,

$$\overline{ref_c}(v_e) = I(F)\,(\overline{ref_c}(v_1), \ldots \overline{ref_c}(v_{n-1})) = \overline{ref_c}(v_f)$$

From this we conclude that the additional edge condition for id in $\mathfrak{G}'$ is satisfied by $\overline{ref_c}$. We obtain $\mathfrak{G}' \models \mathfrak{G}[c, \overline{ref_c}]$, hence $\mathfrak{G}' \models \mathfrak{G}[c, ref_c]$, thus Eqn. (23.6) holds. Now Lemma 13.8 yields $\mathcal{M} \models \mathfrak{G} \iff \mathcal{M} \models \mathfrak{G}'$.

Next we consider the case $c_e = c_f > c$. We want to apply Lemma 13.8 to c_e, so let ref_{c_e} be a partial valuation for c_e. In order to apply Lemma 13.8, it is sufficient to show that

$$\mathfrak{G} \models \mathfrak{G}[c_e, \overline{ref_{c_e}}] \quad\iff\quad \mathfrak{G}' \models \mathfrak{G}[c_e, \overline{ref_{c_e}}] \tag{23.7}$$

holds for each extension $\overline{ref_{c_e}}$ of ref_{c_e} to $area(c_e) \cap V$. So let $\overline{ref_{c_e}}$ be such an extension, If $\overline{ref_{c_e}}$ does not satisfy the edge-conditions for e and f, we have $\mathfrak{G} \not\models \mathfrak{G}[c, \overline{ref_{c_e}}]$ and $\mathfrak{G}' \not\models \mathfrak{G}[c, \overline{ref_{c_e}}]$, thus Eqn. (23.7) holds. So let $\overline{ref_{c_e}}$ satisfy the edge-conditions for e and f. Analogously to the case $c_e = c_f = c$ we obtain $\overline{ref_{c_e}}(v_e) = \overline{ref_{c_e}}(v_f)$. Moreover, for each extension ref_c of $\overline{ref_{c_e}}$ to a partial valuation of c, we obtain

$$\mathfrak{G} \models \mathfrak{G}[c, ref_c] \quad\iff\quad \mathfrak{G}' \models \mathfrak{G}[c, ref_c]$$

This can be seen analogously to the case $c_e = c_f = c$, as $\mathfrak{G}$ and $\mathfrak{G}'$ differ only by adding the edge edge id in c, but for each extension of ref_c to $area(c) \cap V$, the edge-condition for id is due to $\overline{ref_{c_e}}(v_e) = \overline{ref_{c_e}}(v_f)$ fulfilled. Now it can easily be shown by induction that for each context d with $c_e > d \geq c$ and each extension ref_d of $\overline{ref_{c_e}}$ to $area(d) \cap V$, we have

$$\mathfrak{G} \models \mathfrak{G}[d, ref_d] \quad\iff\quad \mathfrak{G}' \models \mathfrak{G}[d, ref_d] \quad .$$

From this we obtain $\mathfrak{G} \models \mathfrak{G}[c_e, \overline{ref_{c_e}}] \iff \mathfrak{G}' \models \mathfrak{G}[c_e, \overline{ref_{c_e}}]$, i.e., Eqn. (23.7) holds again.

Next we consider the case $c_e > c_f > c$. The basic idea of the proof is analogous to the last cases, but we have two nested inductions. Again we want to apply Lemma 13.8 to c_e, so let ref_e be a partial valuation for c_e. Again we show that Eqn. (23.7) holds for each extension $\overline{ref_e}$ of ref_e to $area(c_e) \cap V$. Similarly to the last case, we assume that $\overline{ref_e}$ satisfies the edge-condition for e. It is sufficient to show that

$$\mathfrak{G} \models \mathfrak{G}[c_f, ref_f] \quad\iff\quad \mathfrak{G}' \models \mathfrak{G}[c_f, ref_f] \tag{23.8}$$

holds for each each extension ref_f of $\overline{ref_e}$ to $area(c_f) \cap V$: Then similarly to the last case, an inductive argument yields that for each context d with $c_e > d \geq c_f$ and each extension ref_d of $\overline{ref_{c_e}}$ to $area(d) \cap V$, we have

$$\mathfrak{G} \models \mathfrak{G}[d, ref_d] \quad\iff\quad \mathfrak{G}' \models \mathfrak{G}[d, ref_d] \quad .$$

From this we obtain $\mathfrak{G} \models \mathfrak{G}[c_e, \overline{ref_e}] \iff \mathfrak{G}' \models \mathfrak{G}[c_e, \overline{ref_e}]$ that is, Eqn. (23.7) holds.

It remains to show that Eqn. (23.8) holds. Let us consider an extension ref_f of $\overline{ref_e}$ to $area(c_f) \cap V$. To prove Eqn. (23.8), it is sufficient to show that

$$\mathfrak{G} \models \mathfrak{G}[c_f, \overline{ref_f}] \quad \Longleftrightarrow \quad \mathfrak{G}' \models \mathfrak{G}[c_f, \overline{ref_f}] \tag{23.9}$$

holds for each extension $\overline{ref_f}$ of ref_f to $area(c_f) \cap V$. Now can perform the same inductive argument like in the last case. If $\overline{ref_f}$ does not satisfy the edge-condition for f, we are done. If $\overline{ref_f}$ satisfies the edge-condition, we have $\overline{ref_f}(v_e) = \overline{ref_f}(v_f)$. Now for each extension ref_c of $\overline{ref_f}$ to $area(c) \cap V$, we again obtain

$$\mathfrak{G} \models \mathfrak{G}[c, ref_c] \quad \Longleftrightarrow \quad \mathfrak{G}' \models \mathfrak{G}[c, ref_c]$$

Now from the usual inductive argument we obtain that for each context d with $c_f > d \geq c$ and each extension ref_d of $\overline{ref_f}$ to $area(d) \cap V$, we have

$$\mathfrak{G} \models \mathfrak{G}[d, ref_d] \quad \Longleftrightarrow \quad \mathfrak{G}' \models \mathfrak{G}[d, ref_d] \quad .$$

From this we conclude that Eqn. (23.9), thus Eqn. (23.8), holds. This finishes the proof for the case $c_e > c_f > c$.

Finally, the cases $c_e > c_f = c$ and $c_f > c_e = c$ can be handled analogously. □

Next, the new rules for constants are introduced. As already been mentioned in the introduction of this chapter, it is well-known that functions f with zero arguments correspond to objects in the universe of discourse. For this reason, a distinction between constants and function names is, strictly speaking, not necessary. So the rules for object names correspond to rules for 1-ary functions (i.e. functions f with $dom(f) = \emptyset$). Their soundness and completeness can be proven analogously to the last section: It is sufficient to provide the rules.

Definition 23.8 (New Rules for Constant Names). *The calculus for existential graph instances over the alphabet $\mathcal{C} \dot\cup \mathcal{F} \dot\cup \mathcal{R}$ consists of all rules of Defs. 12.14, 15.2 and 23.5. Moreover, if $C \in \mathcal{C}$ is a constant name, the following additional transformations may be performed:*

- **Constant Identity Rule**

  Let e, f be two unary edges with $\nu(e) = (v_e)$, $\nu(f) = (v_f)$, $ctx(v_e) = ctx(e)$, $ctx(v_f) = ctx(f)$, and $\kappa(e) = \kappa(f) = C$. Let c be a context with $c \leq ctx(e)$ and $c \leq ctx(f)$. Then arbitrary identity-links id with $\nu(id) = (v_e, v_f)$ may be inserted into c or erased from c.

- **Existence of Constants Rule**

  In each context c, we may add a fresh vertex v and an fresh unary edge e with $\nu(e) = (v)$ and $\kappa(e) = C$. Vice versa, if v and e are a vertex and an edge in c with $\nu(e) = (v)$ and $\kappa(e) = F$ such that v is not incident with any other edge, e and v may be erased from c.

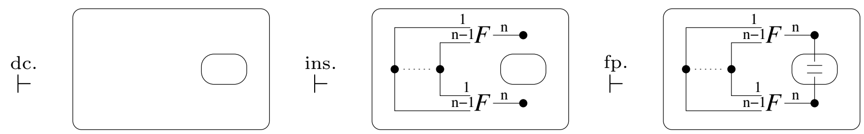
  That is: Devices ●— C may be inserted into or erased from c.

As objects are handled like 1-ary functions, we immediately obtain the soundness of the rules from Lem. 23.6 and Lem. 23.7. It remains to prove the completeness of the extended calculus.

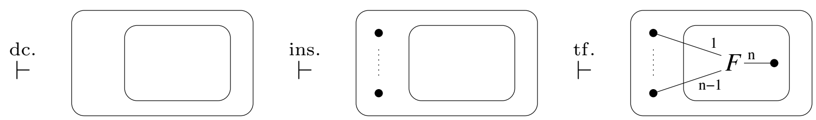
Theorem 23.9 (Extended Calculus is Complete). *Each set* $\mathfrak{H} \cup \{\mathfrak{G}\}$ *of EGIs over* $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *satisfies*

$$\mathfrak{H} \models \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \vdash \mathfrak{G}$$

Due to the remark before Def. 23.8 and Thm. 23.4, it is sufficient to show that for each $F \in \mathcal{F}$, the graph $\mathfrak{G}_F$ can be derived with the new rules. The functional property rule (abbreviated by fp) enables us to derive the left subgraph of $\mathfrak{G}_F$ as follows:



The right subgraph of $\mathfrak{G}_F$ can be derived with the total function rule (tf):



So we can derive $\mathfrak{G}_F$ as well, thus we are done. □

23.3 Examples for EGIs with Constants and Functions

In this section, a few examples for graphs with constant and function names are provided. This will be EGIs as well as formal existential graphs (recall that we said in Chpt. 22 that formal existential graphs over an extended alphabet are canonically defined).

In the symbolic notation of logic, if we have constant names and function names, the notation of *terms* is usually introduced. For example, if C, D are constant names, x is a variable and F, G are function names with two arguments, then $F(G(C,x),D)$ is a term. Before the examples are provided, it shall briefly discussed how terms can be represented within the system of EGIs.

In the symbolic notation of logic, terms can be used as arguments for other terms as well. In the syntax of EGIs, constant names and function names are exactly treated like relation names, thus in this system, we can not use terms as arguments of other terms. Particularly, the mapping Ψ (see page 206) cannot be applied to formulas with functions, and there is no obvious direct counterpart of terms in EGIs. So there are two possibilities to translate a formula f with terms to an EGIs:

1. f is replaced by a semantically equivalent formula f', where the object names and function names are syntactically handled like relation names. Then f' can be translated via Ψ to an EGI.
2. The definition of Ψ is extended in order to encompass terms.

The first possibility can be formalized by a mapping Π, which transforms a formula with terms into a equivalent formula without terms. It is defined as follows:

- To each subterm occurrence t of a term which is not a variable, we assign a fresh variable α_t.
- If t is an atomar term which is a constant name C, let $\Pi(t)$ be the formula $C(\alpha_C)$.
- Let $t := F(t_1, \ldots, t_n)$, where F be a function name with n arguments and let $t_1, \ldots, t_n$ are terms. Then

$$\Pi(t) := \Pi(t_1) \wedge \ldots \Pi(t_n) \wedge F(\alpha_{t_1}, \ldots, \alpha_{t_n}, \alpha_t)$$

- For a subformula $f := R(t_1, \ldots, t_n)$, let

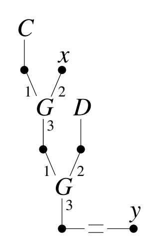
$$\Pi(f) := \exists\vec{\alpha}(\Pi(t_1) \wedge \ldots \Pi(t_n) \wedge R(\alpha_{t_1}, \ldots, \alpha_{t_n}))$$

  Here $\vec{\alpha}$ is the set of all fresh variables α_t, where t is a subformula occurrence of one of the terms t_i.
- We set $\Pi(f_1 \wedge f_2) := \Pi(f_1) \wedge \Pi(f_2)$, We set $\Pi(\neg f) := \neg\Pi(f)$ and $\Pi(\exists\alpha : f) := \exists\alpha : \Pi(f)$.

For example, for $f \ := \ y = F(G(C, x), D)$, we have

$$\begin{aligned}\Pi(f) := {} & \exists x_C \exists X_D \exists x_F \exists x_G : \\ & (C(x_C) \wedge D(x_D) \wedge G(x_C, x, x_G) \wedge F(x_G, x_D, x_F) \wedge y = x_F)\end{aligned}$$

This formula can be translated with Ψ to an EGI with variables. We have:

$\Psi(\Pi(f)) \quad :=$ 

Instead of first converting f, we can alternatively extend the definition of Ψ, as it was provided on page 206). The extended version of Ψ will be denoted by Ψ_{ext}. The original definition of Ψ starts with the translation of atomar subformulas $R(\alpha_1, \ldots, \alpha_n)$. This start is replaced by the following definition of Ψ_{ext}. In addition to Ψ, we introduce a mapping Ψ_V which assigns to each term t a vertex $\Psi_V(t)$ of $\Psi(t)$ (these vertices so-to-speak correspond to the new variables α_t we used in the definition of Π. Now Ψ_{ext} and Ψ_V are defined for EGIs over extended alphabets as follows:

Definition of $\boldsymbol{\Psi_{ext}}$ and $\boldsymbol{\Psi_V}$

- If t is an atomar term which is a variable α, let $\Psi_{ext}(t)$ be EGI with variables which contains only one vertex v, labeled with α. Let $\Psi_V(t) := v$.
- If t is an atomar term which is a constant name C, let $\Psi_{ext}(t)$ be EGI with variables which contains only one generic vertex v and an edge e, where e which is incident with v and labeled with C. Let $\Psi_V(t) := v$.
- Let $t := F(t_1, \ldots, t_n)$, where F be a function name with n arguments and let $t_1, \ldots, t_n$ are terms. To $t_1, \ldots, t_n$, we have already assigned graphs $\Psi_{ext}(t_1), \ldots, \Psi_{ext}(t_n)$, which contain dedicated vertices $\Psi_V(t_1), \ldots, \Psi_V(t_n)$, resp. Then let $\Psi_{ext}(t)$ be the graph which is obtained as follows:

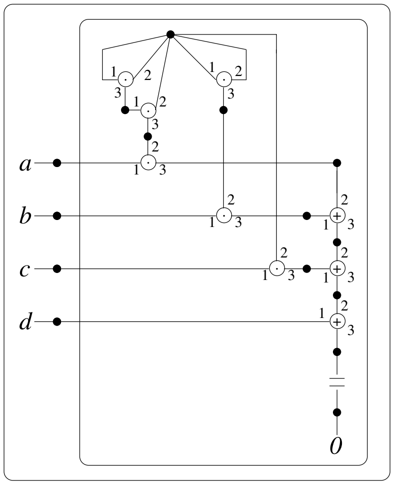
  We take the juxtaposition of the graphs $\Psi_{ext}(t_1), \ldots, \Psi_{ext}(t_n)$. To the sheet of assertion, we add a fresh vertex v and an $(n+1)$-ary edge e such that e is labeled with F and we have $e = (\Psi_V(t_1), \ldots, \Psi_V(t_n), v)$. Let $\Psi_V(t) := v$.
- Now let $f := R(t_1, \ldots, t_n)$, where R is an n-ary relation name and $t_1, \ldots, t_n$ are terms. Similarly to the last case, let $\Psi_{ext}(f)$ be the graph which is obtained as follows:

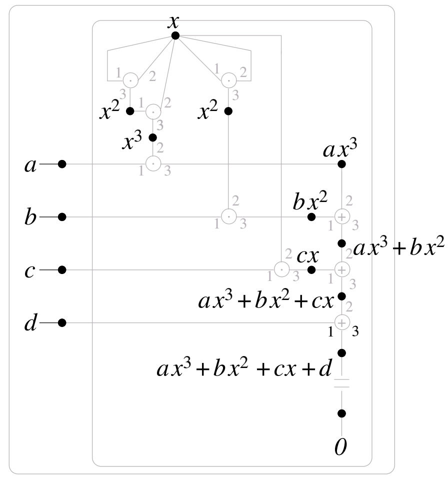
  We take the juxtaposition of the graphs $\Psi_{ext}(t_1), \ldots, \Psi_{ext}(t_n)$. To the sheet of assertion, we add a fresh n-ary edge e such that e is labeled with R and we have $e = (\Psi_V(t_1), \ldots, \Psi_V(t_n))$.
- The remaining cases, $(f_1 \wedge f_2,\ \neg f,\ \exists\alpha.f)$ are handled exactly like in the original definition of Ψ.

It can easily shown by induction over the construction of terms that both approaches yield the same graphs, i.e. for each formula f with terms, we have $\Psi(\Pi(f)) = \Psi_{ext}(f)$. Moreover, this definition strictly extends the definition of Ψ, i.e. if f does not contain any constant names or function names, we have $\Psi_{ext}(f) = \Psi(f)$.

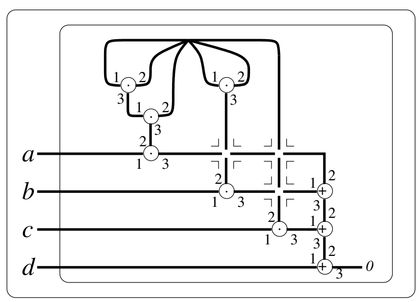
To provide an example, let us express that each polynomial of degree 3 has a root. The formula in $\mathcal{FO}$ is as follows: $\forall a, b, c, d \exists x : ax^3 + b^2 + cx + d = 0$. Below, the corresponding EGI is depicted. On the right, the mapping Ψ_V is visualized by labeling for each the subterm t of $ax^3 + b^2 + cx + d$ the vertex $\Psi_V(t)$ with t.

To provide an example, let us express that each polynomial of degree 3 has a root. The formula in $\mathcal{FO}$ is as follows: $\forall a, b, c, d \exists x : ax^3 + b^2 + cx + d = 0$. Below, the corresponding EGI and EG are depicted.





Below, the corresponding formal EG is depicted.

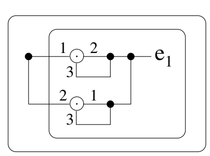


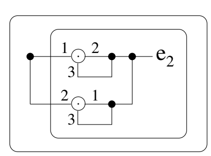
The next example is a formal proof with EGIs for a trivial fact in group theory, namely the uniqueness of neutral elements. Assume that e_1 and e_2 are neutral elements. In $\mathcal{FO}$, this can be expressed as follows:

$$\forall x.x \cdot e_1 = e_1 = e_1 \cdot x \qquad \text{and} \qquad \forall x.x \cdot e_2 = e_2 = e_2 \cdot x$$

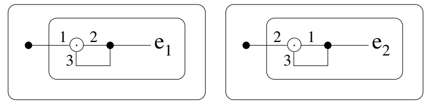
From this we can conclude $e_1 = e_2$. In the following, a formal proof with EGIs for this fact is provided. We assume that e_1, e_2 are employed as constant names and $\cdot$ as function name.

We start with the assumption that e_1, e_2 are neutral elements, i.e.



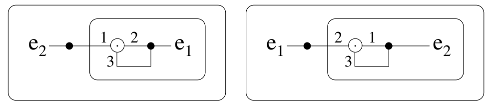


Erasure yields:

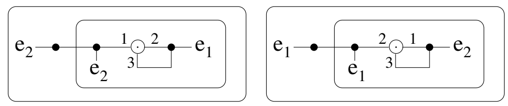


In the 'classical' proof, the universal quantified variable x is replaced by e_2 resp. e_1 in the formulas above. This idea is adopted for EGIs.

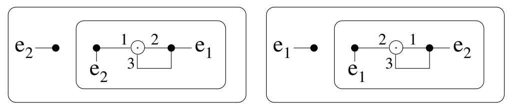
First, we insert e_1 and e_2 (i.e., edges which are labeled with e_1 and e_2) as follows:



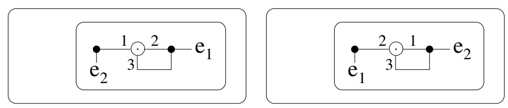
The edges are iterated:



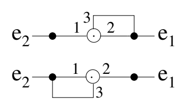
Now we can remove the identity edges with the constant identity rule.



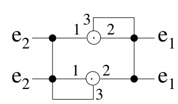
The next graph is derived with the existence of constants rule.



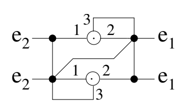
Next, we remove the double cuts and redraw the graph.



We can insert identity edges with the constant identity rule.



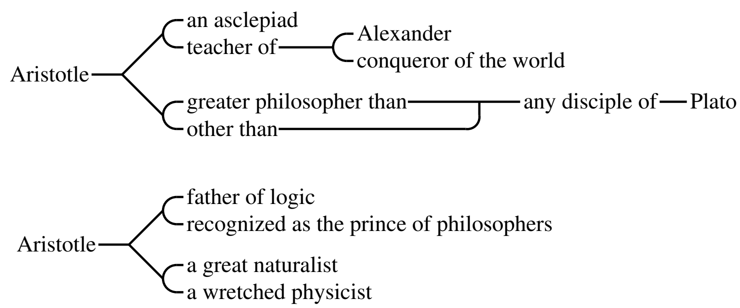
The functional property rule now allows to add another identity edge.



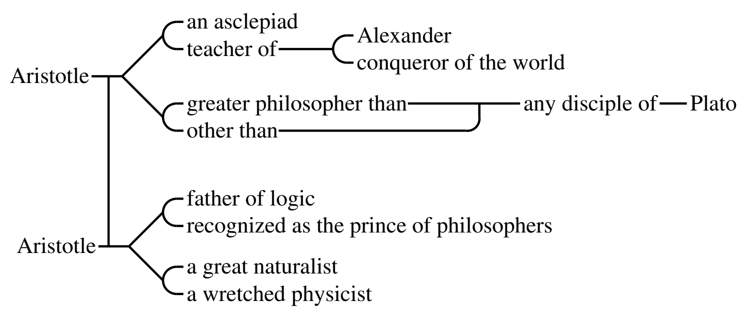
With the erasure rule, we can finally obtain:

As this graph expresses that e_1 and e_2 are identical, we are done.

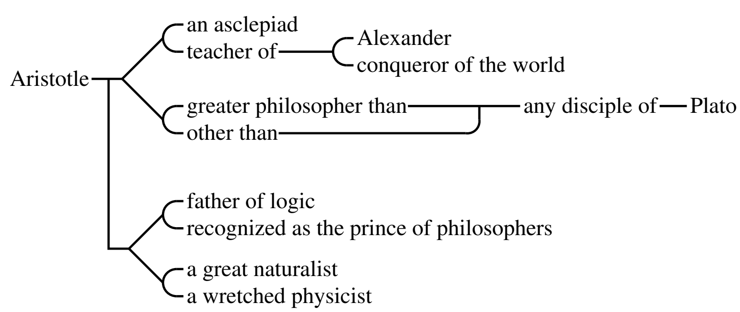
As a final, simple example, a proof for formal EGs with constants is provided. We come back to Peirce's example about Aristotle which was given in Fig. 11.1 on page 102. Let us assume now that 'Aristotle' is a name for a constant. If the graph



is given, we can now add a LoI as follows:



Now we can erase one instance of 'Aristotle', thus we obtain



which is the graph of Fig. 11.1, drawn slightly different.

24

Vertices with Constants

In the last chapter, we have introduced constants and function names to the alphabet. Both kinds of names have been employed as special relation names. For function names, as functions are relations with special properties, this approach is self-suggesting. For constants, the situation is different.

In symbolic notations of logic, when defining the well-formed formulas, constants are syntactically treated like variables. The counterparts of variables in EGIs[1] are vertices, which can roughly be understood as existentially quantified variables. So another approach to implement constants in EGIs is to label the vertices with them. This approach shall be discussed and carried out in this chapter. In order to distinguish the EGIs from the last chapter where constant names are assigned to edges to the EGIs in this chapter where constant names are assigned to vertices, we call the EGIs in this chapter *vertex-based EGIs*.

In the last chapter, we extended the alphabet we consider. Now we extend slightly the syntax of graphs, thus we need a different approach. In the next section 24.1, the syntax and semantics for EGIs is modified to vertex-based EGIs. The correspondence between EGIs and vertex-based EGIs is elaborated in Sec. 24.2, and based on this section, an ADEQUATE (i.e., a sound and complete) calculus for vertex-based EGIs is provided in Sec. 24.3. Finally, in Sec. 24.4, it is investigated how ligatures are handled in vertex-based EGIs.

24.1 Syntax and Semantics

We start with the syntax for vertex-based EGIs. They are based on an alphabet with constants and function names, as it is defined in Def. 23.1. Moreover, we have to introduce a new sign '$*$', called the *generic marker*.

[1] In this chapter, when the terms EGIs and EGs are used, we refer to EGIs and EGs over extended alphabets, as they have been defined in the last chapter.

Definition 24.1 (Vertex-Based Existential Graph Instances). *A structure* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ *is called a* VERTEX-BASED EXISTENTIAL GRAPH INSTANCE (VERTEX-BASED EGI) OVER AN ALPHABET $\mathcal{A} = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$*, iff*

- $(V, E, \nu, \top, Cut, area)$ *is a relational graph with cuts and dom. nodes,*
- $\kappa : E \to \mathcal{F} \dot\cup \mathcal{R}$ *is a mapping such that* $|e| = ar(\kappa(e))$ *for each* $e \in E$*, and*
- $\rho : V \to \{*\} \dot\cup \mathcal{C}$ *is a mapping.*

We set $V^* := \{v \in V \mid \rho(v) = *\}$ *and* $V^{\mathcal{C}} := \{v \in V \mid \rho(v) \in \mathcal{C}\}$*. The vertices* $v \in V^*$ *are called* GENERIC VERTICES*, and the vertices* $v \in V^{\mathcal{C}}$ *are called* CONSTANT OR OBJECT VERTICES.

Similar to Defs. 12.7 and 23.2, the system of all vertex-based EGIs over $\mathcal{A}$ *will be denoted by* $\mathcal{VEGI}^{\mathcal{A}}$.

Although syntactically (slightly) different, there is a close relationship between the system $\mathcal{EGI}^{\mathcal{A}}$, as is has been been elaborated in the last chapter, and $\mathcal{VEGI}^{\mathcal{A}}$. This relationship will be investigated and used in the next sections. But first, the syntactical difference between EGIs and vertex-based EGIs and possible diagrammatic representation of the latter shall be discussed.

Beginning on page 128, the conventions for diagrammatically depicting EGIs have been provided. In these representations, each vertex v has been represented as a bold spot, the vertex-spot of v. Now in vertex-based EGIs, vertices are additionally labeled. Thus we have mainly two possibilities to represent vertex-based EGIs:

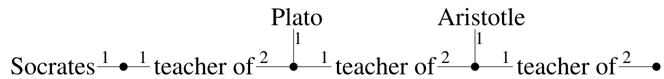
1. Each vertex v is still drawn as bold spot, but now additionally labeled with $\rho(v)$ (this representation was used for EGIs with variables as well). For sake of convenience, we can omit the labeling of generic spots.
2. For a constant vertex v, we use its label $\rho(v)$ *instead* of the vertex-spot.

In the following, we will use both methods to represent vertex-based EGIs. These two approaches shall be now discussed with an small example.

Let us consider the sentence 'Socrates is the teacher of Plato, Plato is the teacher of Aristotle, and Aristotle is the teacher of somebody'. Let us assume that we have an alphabet $\mathcal{A}$ with 'Socrates', 'Plato' and 'Aristotle' as constants and 'teacher_of' as 2-ary relation name. We first formalize the sentence as EGI $\mathfrak{G}_1 := (V_1, E_1, \nu_1, \top_1, \emptyset, \emptyset, \kappa_1)$. We set:

$$
\begin{aligned}
V_1 &= \{v_1, v_2, v_3, v_4\}\\
E_1 &= \{e_1, e_2, e_3, e_4, e_5, e_6\}\\
\nu_1 &= \{(e_1, (v_1)), (e_2, (v_1, v_2)), (e_3, (v_2)), (e_4, (v_2, v_3)), (e_5, (v_3)), (e_4, (v_3, v_4))\}\\
\kappa_1 &= \{(e_1, \text{Socrates}), (e_2, \text{teacher\_of}), (e_3, \text{Plato}),\\
&\qquad (e_4, \text{teacher\_of}), (e_5, (\text{Aristotle})), (e_6, (\text{teacher\_of}))\}
\end{aligned}
$$

The graphical representation of $\mathfrak{G}_1$ is as follows:



Next, the vertex-based EGI $\mathfrak{G}_2 := (V_2, E_2, \nu_2, \top_2, \emptyset, \emptyset, \kappa_2, \rho_2)$ formalizes the sentence as well:

$$\begin{aligned}
V_2 &= \{v_1, v_2, v_3, v_4\} \\
E_2 &= \{e_1, e_2, e_3\} \\
\nu_2 &= \{(e_1, (v_1, v_2)), (e_2, (v_2, v_3)), (e_3, (v_3, v_4))\} \\
\kappa_2 &= \{(e_1, \text{teacher_of}), (e_2, \text{teacher_of}), (e_3, \text{teacher_of})\} \\
\rho_2 &= \{(v_1, \text{Socrates}), (v_2, \text{Plato}), (v_3, \text{Aristotle}), (v_4, *)\}
\end{aligned}$$

Next, the two graphical representations of $\mathfrak{G}_2$ are provided. Due to the first convention, $\mathfrak{G}_2$ can be represented as follows:

The second convention yields:

Analogously to the definition of formal EGs as classes of EGIs, we can introduce formal vertex-based EGs as classes of vertex-based EGIs. When doing so, the second convention is appropriate. The corresponding formal vertex-based EG will then depicted as follows:

In Sec. 12.3, further notions like *subgraph, (partial) isomorphism, juxtaposition* for EGIs have been formally introduced (see Defs. 12.10, 12.11, 12.12, 12.13). Strictly speaking, as we have extended the syntax of EGIs, these notions have to be defined for vertex-based EGIs as well. But recall that all definitions had first been carried out for relational graphs with cuts, then they had been canonically lifted to EGIs by additionally demanding that the structures (like subgraphs or juxtapositions) or mappings (like isomorphisms or partial isomorphisms) respect the labeling κ. Analogously, we can for example define subgraphs of vertex-based EGIs as substructures which are subgraphs for the underlying relational graphs with cuts and which now respect κ and ρ. For this reason, we will use the terms 'subgraph', 'isomorphism' etc for vertex-based EGIs as well.

For ligatures, it is not that obvious how they should be defined for vertex-based EGIs, as it is not clear whether ligatures must contain only generic vertices, or not. Thus ligatures will be re-defined in this chapter. On the other hand, the notation of hooks remains.

Next, we turn to the semantics for vertex-based EGIs. The semantics for EGIs was defined in Chpt. 13 and was based on the notion of (partial) valuations for EGIs, as they had been defined in Def. 13.2. For vertex-based EGIs, any valuation on the vertices has on the constant vertices to coincide with the interpretation of the constant names. Thus we have to (slightly) extend the notions of valuations as follows:

Definition 24.2 (Partial and Total Valuations). *Let a vertex-based EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given and let* (U, I) *be a relational structure over* $\mathcal{A}$*. Each mapping* $ref : V' \to U$ *with* $V^{\mathcal{C}} \subseteq V' \subseteq V$ *and* $ref(v) = I(\rho(v))$ *for all* $v \in V^{\mathcal{C}}$ *is called a* PARTIAL VALUATION OF $\mathfrak{G}$*. If* $V' = V$*, then* ref *is called* (TOTAL) VALUATION OF $\mathfrak{G}$*.*

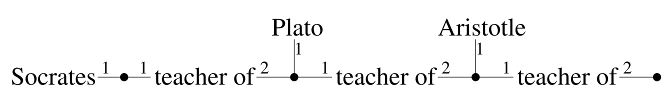
Let $c \in Cut \cup \{\top\}$*. If* $V' \supseteq \{v \in V^* \mid v > c\}$ *and* $V' \cap \{v \in V^* \mid v \leq c\} = \emptyset$*, then* ref *is called* PARTIAL VALUATION FOR c*. If* $V' \supseteq \{v \in V^* \mid v \geq c\}$ *and* $V' \cap \{v \in V^* \mid v < c\} = \emptyset$*, then ref is called* EXTENDED PARTIAL VALUATION FOR c*.*

All remaining definition of Chpt. 13 can now be adopted for vertex-based EGIs, which yields the formal semantics for vertex-based EGIs. Note that we can adopt the lemmata and theorems of Chpt. 13 for vertex-based EGIs as well. Particularly, we will use Thms. 13.7 and 13.8 for vertex-based EGIs.

24.2 Correspondence between vertex-based EGIs and EGIs

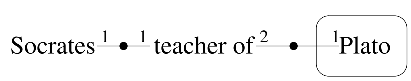
For extended EGIs over an alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$, i.e. for the system $\mathcal{EGI}^{\mathcal{A}}$, we have developed in the last chapter a sound and complete calculus. The first step to adopt this calculus for vertex-based EGIs over $\mathcal{A}$, i.e. for the system $\mathcal{VEGI}^{\mathcal{A}}$ is to find a reasonable 'translation' $\Xi : \mathcal{EGI}^{\mathcal{A}} \to \mathcal{VEGI}^{\mathcal{A}}$.

Let us consider two simple EGIs. The first one is an EGI we have already discussed:

$\mathfrak{G}_1 :=$ 

.

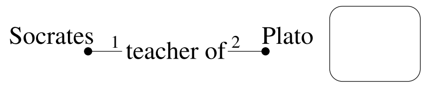
In order to consider an example where edge labeled with an element $C \in \mathcal{C}$ is enclosed by cut, we consider as second example the following EGI with the meaning 'Socrates is the teacher of someone who is not Plato'.

$\mathfrak{G}_2 :=$ 

In EGIs, an edge e labeled with a constant $C \in \mathcal{C}$ is incident with exactly one vertex v_e. In order to translate extended EGIs to vertex-based EGIs, such edges have to be converted to labeled vertices. One might think that we simply find a corresponding vertex-based EGI by erasing e and labeling v_e with C. For the first graph, this yields the graph

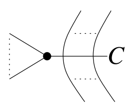
$\mathfrak{G}_1^R :=$ 

which is indeed a reasonable translation of $\mathfrak{G}_1$. But for $\mathfrak{G}_2$, this translation yields

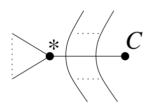
$\mathfrak{G}_2^R :=$ 

thus a graph with a different meaning (this graph is, in contrast to $\mathfrak{G}_2$, not satisfiable). Thus this idea for a translation does not work.

We see if we want to convert the edge e to a labeled vertex, this vertex has to be placed in the same cut as e, thus we cannot use the vertex v_e for this purpose. Instead, we will place a fresh vertex in the cut of e, labeled with C, and which is linked with an identity edge to v_e. Informally depicted, we will replace an device like

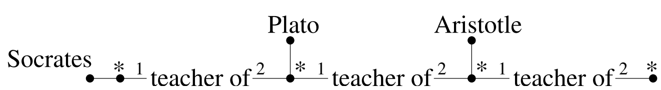


by

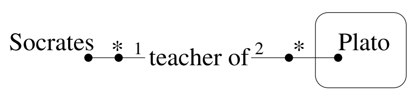


(the edge-lines left from v_e shall indicate that v_e may be incident with other edges as well, and the segments of cut-lines shall indicate that for an edge $e = (v)$ with $\kappa(e) = C$, it might happen that $ctx(e) < ctx(v)$ holds).

This idea will be captured by the mapping $\Xi : \mathcal{EGI}^{\mathcal{A}} \to \mathcal{VEGI}^{\mathcal{A}}$. For our examples, we will have

$\Xi(\mathfrak{G}_1) =$ 

and

$\Xi(\mathfrak{G}_2) =$ 

After these examples, the formal definition of Ξ can be provided.

Definition 24.3 (Translation from EGIs to Vertex-Based EGIs). *Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be an EGI over the alphabet* $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$. *Let* $E^{\mathcal{C}} := \{e \in e \mid \kappa(e) \in \mathcal{C}\}$. *For each* $e \in E^{\mathcal{C}}$, *let* v_e *be that vertex which is incident with* e, *and let* v'_e *be fresh vertex and* id_e *an fresh edge. Now let* $\Xi(\mathfrak{G}) := (V', E', \nu', \top', Cut', area', \kappa')$ *be the following vertex-based EGI:*

- $V' := V \,\dot\cup\, \{v'_e \mid e \in E^{\mathcal{C}}\}$,

- $E' := E\backslash E^{\mathcal{C}} \dot\cup \{id_e \mid e \in E^{\mathcal{C}}\}$,
- $\nu := \nu\big|_{E\backslash E^{\mathcal{C}}} \dot\cup \{(id_e, (v_e, v'_e)) \mid e \in E^{\mathcal{C}}\}$
- $\top' := \top$
- $Cut' := Cut$
- $area'(c) := area(c)\backslash(E^{\mathcal{C}} \cap area(c)) \dot\cup \{v'_e, id_e \mid e \in E^{\mathcal{C}} \cap area(c)\}$ *for each* $c \in Cut' \cup \{\top'\}$
- $\kappa' := \kappa \dot\cup \{(id_e, \doteq) \mid e \in E^{\mathcal{C}}\}$, *and*
- $\rho' := \{(v, *) \mid v \in V\} \dot\cup \{(v'_e, \kappa(e)) \mid e \in E^{\mathcal{C}}\}$.

For any element k of $\mathfrak{G}$ which is not an edge labeled with a constant name, k is an element of $\Xi(\mathfrak{G})$ as well. We will write $\Xi(k)$ to refer there is a corresponding element in $\Xi(\mathfrak{G})$. Analogously, for any subgraph $\mathfrak{G}_0$ of $\mathfrak{G}$, there exists a corresponding subgraph in $\Xi(\mathfrak{G})$ which will be denoted by $\Xi(\mathfrak{G}_0)$ (for any edge $e \in \mathfrak{G}_0$ labeled with a constant name, v'_e and id_e are the corresponding elements in $\Xi(\mathfrak{G}_0)$).

Obviously, if Ξ is applied to an EGI $\mathfrak{G}$, each constant vertex in $\Xi(\mathfrak{G})$ is induced by an edge in $\mathfrak{G}$, thus it is incident with exactly edge, which is an identity edge. But in vertex-based EGIs, a constant vertex is allowed to be incident with an arbitrary number of edges (it can even be isolated). For this reason, Ξ is not surjective. The range of Ξ is captured by the following definition.

Definition 24.4 (Vertex-Based EGIs with Separated Constant Vertices). *. Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an vertex-based EGI such that for each vertex $v \in V^{\mathcal{C}}$, we have that v is incident with exactly one edge e, and e has the form $e = (w, v)$ and satisfies $ctx(e) = ctx(v)$, $\kappa(e) = \doteq$ and $\rho(w) = *$. Then $\mathfrak{G}$ is said to have* Separated Constant Vertices *The system of all vertex-based EGIs with separated constant vertices will be denoted by $\mathcal{VEGI}^{sp,\mathcal{A}}$.*

The following lemma clarifies the relationship between $\mathcal{EGI}^{\mathcal{A}}$ and $\mathcal{VEGI}^{sp,\mathcal{A}}$ and will be the basis for the forthcoming calculus for vertex-based EGIs. It is easily be shown, thus the proof is omitted.

Lemma 24.5 (Ξ is Meaning-Preserving). *The mapping $\Xi : \mathcal{EGI}^{\mathcal{A}} \to \mathcal{VEGI}^{sp,\mathcal{A}}$ is a bijection, and it is meaning-preserving, i.e. for each model $\mathcal{M}$ and each EGI $\mathfrak{G}$ we have*

$$\mathcal{M} \models \mathfrak{G} \quad \Longleftrightarrow \quad \mathcal{M} \models \Xi(\mathfrak{G}) \quad .$$

Particularly, for $\mathfrak{H} \cup \{\mathfrak{G}\} \subseteq \mathcal{EGI}^{\mathcal{A}}$ we have

$$\mathfrak{H} \models \mathfrak{G} \quad \Longleftrightarrow \quad \Xi[\mathfrak{H}] \models \Xi(\mathfrak{G}) \quad .$$

24.3 Calculus for vertex-based EGIs

In this section, we will develop a sound and complete calculus for vertex-based EGIs.

In the last chapter, we extended the calculus of EGIs to EGIs over extended alphabets, which now include constants and function names. For developing a sound and complete calculus for these extended EGIs, we benefitted of interpreting constants and function names as special relations. Thus is was more easily to understand the addition of constants and function names as a restriction of the class of models, not as syntactical extension of the class of EGIs. And due to this understanding, it was easy to develop an adequate calculus for extended EGIs by simply adding some rules.

Now the situation is different: We have extended the syntax of EGIs by adding the additional function ρ. Moreover, we have seen in the last section that there is no a-priori correspondence between extended EGIs and vertex-based EGIs, but only between extended EGIs and vertex-based EGIs *with separated constant vertices*. But it is this correspondence which will give rise to an adequate calculus for (arbitrary) vertex-based EGIs.

Of course, we want to take benefit of the adequate calculus for extended EGIs. So let us first consider some simple syntactical problems we have cope with when we want to adopt this calculus.

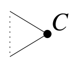
1. May the insertion- or erasure-rule be applied to subgraphs which contain vertices with constants?
2. In the definition of the iteration/deiteration-rule, we hade to use the relation Θ on vertices (for Θ, see Def 15.1 on page 163), and we need a similar relation for vertex-based EGIs as well. We have to investigate whether this relation acts only on generic vertices, or on constant vertices as well.
3. How do we cope with the fact that Ξ does not map *onto* $(\mathcal{VEGI}^{sp})^{\mathcal{A}}$, but only onto $\mathcal{VEGI}^{sp,\mathcal{A}}$?

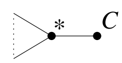
We start with the last problem. Via the mapping Ξ, it is easy to find an adequate calculus for $\mathcal{VEGI}^{sp,\mathcal{A}}$, but not for $\mathcal{VEGI}^{\mathcal{A}}$. So if we had a rule which allows to transform each graph of $\mathcal{VEGI}^{\mathcal{A}}$ into a graph of $\mathcal{VEGI}^{sp,\mathcal{A}}$ (and vice versa), then this calculus can be extended to $\mathcal{VEGI}^{\mathcal{A}}$. This rule shall be provided now.

Definition 24.6 (Separating a Constant Vertex). *Let a vertex-based EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given. Let* $v \in V^{\mathcal{C}}$ *be a vertex. Let* v' *be a fresh vertex and* e' *be a fresh edge. Furthermore let* $c := ctx(v)$*. Now let* $\mathfrak{G}' := (V', E', \nu', \top, Cut, area', \kappa', \rho')$ *be the graph with*

- $V' := V \dot\cup \{v'\}$,
- $E' := E \dot\cup \{e'\}$,
- $\nu := \nu \dot\cup \{(e', (v, v'))\}$
- $area'(c) := area(c) \dot\cup \{v', e'\}$, *and for* $d \in Cut' \cup \{\top'\}$ *with* $d \neq c$ *we set* $area'(d) := area(d)$,
- $\kappa' := \kappa \dot\cup \{(e', \doteq)\}$, *and*
- $\rho' := \rho \backslash \{(v, \rho(v))\} \cup \{(v, *), (v', \rho(v))\}$.

Then we say that $\mathfrak{G}'$ *is obtained from* $\mathfrak{G}$ *by* SEPARATING THE CONSTANT VERTEX v (INTO v AND v') *and* $\mathfrak{G}$ *is obtained from* $\mathfrak{G}'$ *by* DESEPARATING THE CONSTANT VERTEX v'.

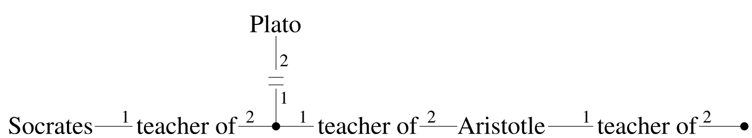
Informally depicted, the device 

is replaced by 

.

For example, if the vertex labeled with 'Plato' is separated, from

we obtain



First we have to show that this rule is meaning-preserving.

Lemma 24.7 ((De)Separating a Constant Vertex is Sound). *If* $\mathfrak{G}$ *and* $\mathfrak{G}'$ *are two vertex-based EGIs such that* $\mathfrak{G}'$ *is obtained from* $\mathfrak{G}$ *by separating the constant vertex* v *(into* v *and* v'*), and if* $\mathcal{M} := (U, I)$ *is a relational structure, we have*

$$\mathcal{M} \models \mathfrak{G} \qquad \Longleftrightarrow \qquad \mathcal{M} \models \mathfrak{G}'$$

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by by separating the object vertex v into v and v'. Let $C := \rho(v)$ and $c := ctx(v)$. We want to apply Lemma 13.8 to c, so let ref be a closed partial valuation for the context c.

Let us first assume that we $\mathcal{M} \models \mathfrak{G}[c, ref]$. That is, there is an extension $\overline{ref}$ of ref to $V \cap area(c)$ with $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}]$. Then $\overline{ref}\,'$ with $\overline{ref}\,' := \overline{ref} \cup \{(v', C)\}$ is an extended valuation for c in $\mathfrak{G}'$ with $\mathcal{M} \models \mathfrak{G}'[c, \overline{ref}\,']$.

If we have on the other hand $\mathcal{M} \models \mathfrak{G}'[c, ref]$, then there is an extension $\overline{ref}\,'$ of ref to $V' \cap area(c)$ with $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}\,']$. Then $\overline{ref} := \overline{ref}\,' \backslash \{(v', C\}$ is an extended valuation for c in $\mathfrak{G}$ with $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}]$.

Now Lemma 13.8 yields the lemma. $\square$

As we have just shown that the separating a constant vertex rule is meaning-preserving, we now know that we can transform with this rule each vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa) \in \mathcal{VEGI}^{\mathcal{A}}$ into a vertex-based EGI $\mathfrak{G}^s \in \mathcal{VEGI}^{sp,\mathcal{A}}$ by separating each constant vertex $v \in V^{\mathcal{C}}$ which is not already separated (and vice versa, we can transform $\mathfrak{G}^s$ into $\mathfrak{G}$). Particularly, for $\mathfrak{G} \in \mathcal{VEGI}^{sp,\mathcal{A}}$ we have $\mathfrak{G}^s = \mathfrak{G}$. As (de)separating vertices is sound, for each vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa) \in \mathcal{VEGI}^{\mathcal{A}}$ and for each model $\mathcal{M}$, we have $\mathcal{M} \models \mathfrak{G} \Leftrightarrow \mathcal{M} \models \mathfrak{G}^s$. Moreover, we have a meaning-preserving bijection mapping $\Xi : \mathcal{EGI}^{\mathcal{A}} \to \mathcal{VEGI}^{sp,\mathcal{A}}$. With these facts, we can now define an adequate calculus for $\mathcal{VEGI}^{\mathcal{A}}$.

To avoid confusion, we use a subscript 'v' for any calculus for $\mathcal{VEGI}$, i.e. we write $\vdash_v$ ('v' for *vertex-based* EGI). Similarly, we use a subscript 'e' for the calculus for $\mathcal{EGI}$ (as in the graphs of $\mathcal{EGI}$, edges are used to denote constants, we could call them *edge-based* EGIs, and now 'e' stands for 'edge-based'). Moreover, for each rule r of the respective calculus, we use r as superscript to refer to this specific rule. For example, we write $\mathfrak{G} \vdash_v^r \mathfrak{G}'$ iff the vertex-based EGI $\mathfrak{G}'$ can be obtained from the vertex-based EGI$\mathfrak{G}$ with the rule r.

In the following, let $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ be a fixed alphabet. Now a canonical idea to define a calculus $\Vdash_v$ for $\mathcal{VEGI}^{\mathcal{A}}$ is as follows (we use the sign '$\Vdash_v$' instead of '$\vdash_v$', as it will turn out that this calculus will not be used, and the sign '$\vdash_v$' is reserved for the calculus which will be employed):

First approach for a calculus for $\mathcal{VEGI}^{\mathcal{A}}$:

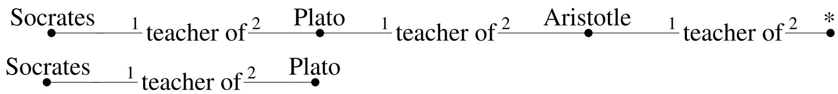
1. $\Vdash_v$ contains the rules 'separating a constant vertex' and 'deseparating a constant vertex', and
2. for all $\mathfrak{G}_a, \mathfrak{G}_b \in \mathcal{VEGI}^{sp,\mathcal{A}}$ and each rule $\Vdash_e^r$ of the calculus for EGIs we define an corresponding rule $\Vdash_v^r$ by setting

$$\mathfrak{G}_a \Vdash_v^r \mathfrak{G}_b \quad :\Longleftrightarrow \quad \Xi^{-1}(\mathfrak{G}_a) \vdash_e^r \Xi^{-1}(\mathfrak{G}_b)$$

The calculus $\Vdash_v$ is sound and complete: First of all, the rules '(de)separating a constant vertex' are sound due to Lem. 24.7, and all other rules are sound, as Ξ is due Lem. 24.5 meaning-preserving and as the rules of $\vdash_e$ are sound. Secondly, the rules of $\vdash_e$ are complete on $\mathcal{EGI}^{\mathcal{A}}$, and Ξ is a bijective mapping from $\mathcal{EGI}^{\mathcal{A}}$ to $\mathcal{VEGI}^{sp,\mathcal{A}}$. So, for $\mathfrak{G}_v, \mathfrak{G}'_v \in \mathcal{VEGI}^{\mathcal{A}}$ with $\mathfrak{G}_v \models \mathfrak{G}'_v$, we can find a proof for $\mathfrak{G}_v \Vdash_v \mathfrak{G}'_v$ as follows: First, we derive $(\mathfrak{G}_v)^s$ from $\mathfrak{G}_v$, then we can find a proof for $(\mathfrak{G}_v)^s \Vdash_v (\mathfrak{G}'_v)^s$ within $\mathcal{VEGI}^{sp,\mathcal{A}}$, and finally, $\mathfrak{G}'_v$ is derived from $(\mathfrak{G}'_v)^s$ by deseparating constant vertices. Thus $\Vdash_v$ is sound and complete, hence we would be done.

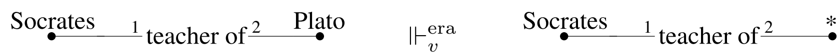
But this calculus is somehow not convenient to be used for vertex-based EGIs, as all rules except the rule 'separating a constant vertex' act only on vertex-based EGIs with separated constant vertices. To see an example, consider the following vertex-based EGIs.

$\mathfrak{G}_1 :=$ $\mathfrak{G}_2 :=$



The general understanding of the erasure-rule is that in positive contexts, we can erase parts of a given graph. So we aim that $\mathfrak{G}_1 \Vdash_v^{\text{era}} \mathfrak{G}_2$ is a application of the erasure-rule (more precisely: A composition of *two* applications of the erasure-rule, namely the erasure of an edge, followed by an erasure of a closed subgraph). But so far, this derivation is not allowed, as neither $\mathfrak{G}_1$ nor $\mathfrak{G}_2$ has separated constant vertices.

Moreover, we have a specific peculiarity with the erasure and insertion rule. The erasure rule for EGIs allows to erase edges labeled with a constant name (and vice versa, the insertion rule allows to insert an edge with a constant name). So we would have



$\Vdash_v^{\text{era}}$

as we have

$\vdash_e^{\text{era}}$

So now with the erasure rule, it is possible to 'generalize' the labels of constant vertices to generic markers in positive context (and vice versa, to 'specialize' generic vertices to constant vertices in negative contexts). Of course, generalizing the labels of vertices in positive contexts must be possible in any complete calculus for vertex-based EGIs, but it is not intended that this is a derivation with a *single application of the erasure-rule.*

We have an approach for a sound and complete calculus for $\mathcal{VEGI}^{\mathcal{A}}$, but in the light of the purpose of the rules, its rules are too weak, and we have an unwanted peculiarity in the erasure- and insertion rule. So we need a different approach which (roughly) extends our first idea and which copes with the before mentioned peculiarity.

The essential methodology for appropriately handling vertex-based EGIs will be: Generic and constant vertices are generally not distinguished, except for special rules for constants. For example, the erasure-rule simply still allows to erase arbitrary subgraphs, no matter whether they contain constant vertices or not. The discussion so far will be made fruitful to show that this calculus is sound and complete.

In order to make the idea clear, all rules will be provided for vertex-based EGIs as (slightly) reformulated rules for EGIs. As a first step, we redefine ligatures and Θ for vertex-based EGIs. The core of the following two definitions directly correspond to Def. 12.8 for ligatures and Def. 15.1 for Θ, but for sake of convenience and comprehensibility, and as some notations are added, the definitions of ligatures and Θ for vertex-based EGIs are completely provided.

For an EGI $(V, E, \nu, \top, Cut, area, \kappa)$, its ligatures had been introduced as subgraphs of (V, E^{id}) (strictly speaking, of $(V, E^{id}, \nu\big|_{E^{id}})$, but the mapping ν was for sake of convenience omitted. See page 126). Now for vertex-based EGIs, we formally do the same. A ligature may both contain generic and constant vertices, and the mapping ρ is not part of a ligature. Thus we do not distinguish between generic and constant vertices. Nonetheless, we assign now to each ligature the set of labels which occur in it.

Definition 24.8 (Ligature). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGI. Then we set $Lig(\mathfrak{G}) := (V, E^{id})$, and $Lig(\mathfrak{G})$ is called the* LIGATURE-GRAPH INDUCED BY $\mathfrak{G}$. *Each connected subgraph of (W, F) of $Lig(\mathfrak{G})$ is called a* LIGATURE OF $\mathfrak{G}$.

If (W, F) is a ligature, we set $\rho(W, F) := \rho(W) := \{\rho(w) \mid w \in W\}$. The set $\rho(W, F)$ is called the a SET OF LABELS OF (W, F). *If we have $\rho(W) = \{*\}$, then (W, F) is called a* GENERIC LIGATURE.

Next we define the relation Θ for vertex-based EGIs. Again, we do not distinguish between constant and generic vertices.

Definition 24.9 (Θ). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an vertex-based EGI. On V, a relation Θ is defined as follows: Let $v, w \in V$ be two vertices. We set $v\Theta w$ iff there exist vertices $v_1, \ldots, v_n$ ($n \in \mathbb{N}$) with*

1. *either $v = v_1$ and $v_n = w$, or $w = v_1$ and $v_n = v$,*
2. *$ctx(v_1) \geq ctx(v_2) \geq \ldots \geq ctx(v_n)$, and*
3. *for each $i = 1, \ldots, n-1$, there exists an identity edge $e_i = \{v_i, v_{i+1}\}$ between v_i and v_{i+1} with $ctx(e_i) = ctx(v_{i+1})$.*

If we have moreover $v_i \in V^*$ for $i = 1, \ldots, n$, we write $v\Theta^* w$.

Please note the following: If $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa) \in \mathcal{VEGI}^{\mathcal{A}}$, if $v_a, v_b \in V$ are vertices, and if v'_a, v'_b are the corresponding vertices in $\mathfrak{G}^s$, then we have $v_a \Theta v_b \Leftrightarrow v'_a \Theta^* v'_b$. This shows that extending Θ to non-generic vertices is not necessary. For example, Θ will be used in the iteration/deiteration rule, but it will turn out that it would be sufficient to consider Θ^* instead. Nonetheless, Θ is extended as the overall idea of the forthcoming calculus is to avoid a too strong distinction between generic and constant vertices.

On the following two pages, the calculus for vertex-based EGIs is provided. All rules are basically the original rules for EGIs (for the transformation rules for ligatures and Peirce's rules for EGIs) resp. the counterparts of the new rules (for constants and functions) for extended EGIs, reformulated for vertex-based EGIs. Nonetheless, as mentioned above, the calculus is completely listed in the well-known semiformal manner. As all rules have already been provided in different previous chapters of this treatise, their formal definitions are omitted now.

Definition 24.10 (Calculus for vertex-based EGIs). Let an vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ over the alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ be given. Then the following transformations may be carried out:

- **Transformation Rules for Ligatures**
  - **isomorphism** $\mathfrak{G}$ may be substituted by an isomorphic copy of itself.
  - **changing the orientation of an identity edge**

    Let $e \in E^{id}$. Then we may change the orientation of e.
  - **adding a generic vertex to a ligature**

    Let $v \in V$ be a vertex which is attached to a hook (e, i). Furthermore let c be a context with $ctx(v) \geq c \geq ctx(e)$. Then the following may be done: In c, a new generic vertex v' and a new identity-edge between v and v' is inserted. On (e, i), v is replaced by v'.
  - **removing a generic vertex from a ligature**

    The rule 'adding a vertex to a ligature' may be reversed.
- **Peirce's Rules for Existential Graphs**
  - **erasure**

    In positive contexts, any directly enclosed edge, isolated vertex, and closed subgraph may be erased.
  - **insertion**

    In negative contexts, any directly enclosed edge, isolated vertex, and closed subgraph may be inserted.
  - **iteration**
    - Let $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0)$ be a (not necessarily closed) subgraph of $\mathfrak{G}$ and let $c \leq ctx(\mathfrak{G}_0)$ be a context such that $c \notin Cut_0$. Then a copy of $\mathfrak{G}_0$ may be inserted into c.

      Moreover, if $v \in V_0$ with $ctx(v) = ctx(\mathfrak{U})$ is a vertex, and if $w \in V_0$ with $ctx(w) = c$ is a vertex with $v \Theta w$, then an identity edge between v and w may be inserted into c.
    - If $v \in V$ is a vertex and $c \leq ctx(v)$ a cut, then a new vertex w and an identity edge between v and w may be inserted into c.
  - **deiteration**

    If $\mathfrak{G}_0$ is a subgraph of $\mathfrak{G}$ which could have been inserted by rule of iteration, then it may be erased.
  - **double cuts**

    Double cuts may be inserted or erased.

- **erasing a vertex**

  An isolated vertex may be erased from arbitrary contexts.
- **inserting a vertex**

  An isolated vertex may be inserted in arbitrary contexts.

• **Rules for Functions**

- **Functional Property Rule (uniqueness of values)**

  Let e and f be two n-ary edges such that $\nu(e) = (v_1, \ldots, v_{n-1}, v_e)$, $\nu(f) = (v_1, \ldots, v_{n-1}, v_f)$, $ctx(e) = ctx(v_e)$, $ctx(f) = ctx(v_f)$, and $\kappa(e) = \kappa(f) = F$. Let c be a context with $c \leq ctx(e)$ and $c \leq ctx(f)$. Then arbitrary identity-links id with $\nu(id) = (v_e, v_f)$ may be inserted into c or erased from c.
- **Total Function Rule (existence of values)**

  Let $v_1, \ldots, v_{n-1}$ be vertices, let $c \leq ctx(v_1), \ldots, ctx(v_{n-1})$ be a context. Then we can add a generic[2] vertex v_n and an edge e to c with $\nu(e) = (v_1, \ldots, v_n)$ and $\kappa(e) = F$. Vice versa, if v_n and e are a vertex and an edge in c with $\nu(e) = (v_1, \ldots, v_n)$ and $\kappa(e) = F$ such that v_n is not incident with any other edge, e and v_n may be erased.

• **Rules for Constants**

- **Constant Identity Rule**

  Let $C \in \mathcal{C}$ be a constant name. Let v, w be vertices with $\rho(v) = \rho(w) = C$. Let c be a context with $c \leq ctx(v)$ and $c \leq ctx(w)$. Then arbitrary identity-links id with $\nu(id) = (v_e, v_f)$ may be inserted into d or erased from c.
- **Existence of Constants Rule**

  Let C be a constant name. In each context c, we may add a fresh vertex v with $\rho(v) = C$. Vice versa, if v is an isolated vertex with $\rho(v) = C$ v may be erased from c.
- **Separating a constant vertex**

  If $v \in V^{\mathcal{C}}$ is a constant vertex, then v may be separated (see Def. 24.6).
- **Deseparating a constant vertex**

  If $\mathfrak{G}$ could be obtained from $\mathfrak{G}'$ by separating a constant vertex v then $\mathfrak{G}'$ may be obtained from $\mathfrak{G}$ by deseparating v.

Similar to the last chapters, for two vertex-based EGIs $\mathfrak{G}_a, \mathfrak{G}_b$, we set $\mathfrak{G}_a \sim \mathfrak{G}_b$ if $\mathfrak{G}_a$ can be transformed into $\mathfrak{G} - b$ with the transformation rules for ligatures, and formal vertex-based EGs are the classes of vertex-based EGIs with respect to $\sim$. For this reason, it was important that in the rules 'adding a generic

[2] Of course, we usually cannot add constant or query vertices with this rule.

vertex to a ligature' and 'removing a generic vertex from a ligature', only generic vertices are considered.

We have eventually defined a calculus for vertex-based EGIs which is convenient, i.e. the general ideas behind the rules for EGIs is adopted for vertex-based EGIs. We now have to show that this calculus is adequate.

Theorem 24.11 ($\vdash_v$ is Sound and Complete for $\mathcal{VEGI}^{\mathcal{A}}$). *For $\mathfrak{H} \cup \{\mathfrak{G}\} \subseteq \mathcal{VEGI}^{\mathcal{A}}$ we have*

$$\mathfrak{H} \models \mathfrak{G} \qquad \Longleftrightarrow \qquad \mathfrak{H} \vdash_v \mathfrak{G}$$

Proof: The soundness of the rules can be proven analogously to the soundness of their counterparts for EGIs. For the only new rule for vertex-based EGIs, i.e. for the rule '(de)separating a constant vertex', its soundness was proven in Lem. 24.7.

So it remains to show that $\vdash_v$ is complete. Due to the discussion for $\Vdash_v$, it is sufficient to show that for vertex-based EGIs with separated constant vertices $\mathfrak{G}_a, \mathfrak{G}_b$ and each rule r of $\Vdash_v^r$, we have

$$\mathfrak{G}_a \Vdash_v^r \mathfrak{G}_b \qquad \Longrightarrow \qquad \mathfrak{G}_a \vdash_v \mathfrak{G}_b$$

That is, as we defined $\mathfrak{G}_a \Vdash_v^r \mathfrak{G}_b \;:\Leftrightarrow\; \Xi^{-1}(\mathfrak{G}_a) \vdash_e^r \Xi^{-1}(\mathfrak{G}_b)$, for each rule $\vdash_e^r$ for extended EGIs, we have to show

$$\Xi^{-1}(\mathfrak{G}_a) \vdash_e^r \Xi^{-1}(\mathfrak{G}_b) \qquad \Longrightarrow \qquad \mathfrak{G}_a \vdash_v \mathfrak{G}_b$$

Recall that $\Xi : \mathcal{EGI}^{\mathcal{A}} \to \mathcal{VEGI}^{\mathcal{A}}$ is bijective (and meaning-preserving). Thus is is sufficient to show that for two extended EGIs $\mathfrak{G}_1, \mathfrak{G}_2$, we have

$$\mathfrak{G}_1 \vdash_e^r \mathfrak{G}_2 \qquad \Longrightarrow \qquad \Xi(\mathfrak{G}_1) \vdash_v \Xi(\mathfrak{G}_2) \tag{24.1}$$

In the following, we will show Eqn. (24.1) for each rule. In most (but not in all) cases, a derivational step can be canonically carried over from EGIs to vertex-based EGIs, i.e., we actually prove the stronger implication

$$\mathfrak{G}_1 \vdash_e^r \mathfrak{G}_2 \;\Rightarrow\; \Xi(\mathfrak{G}_1) \vdash_v^r \Xi(\mathfrak{G}_2) \tag{24.2}$$

- **isomorphism, changing the orientation of an identity edge, adding a vertex to a ligature, removing a vertex from a ligature**

  For these rules, it is easy to see that Eqn. (24.2) holds.

- **erasure/insertion**

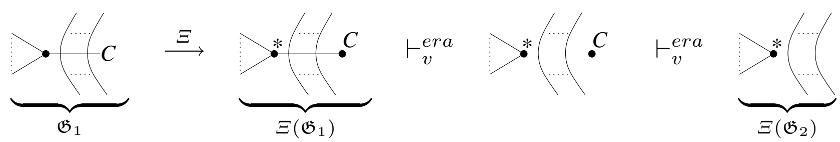
  Due to symmetry reasons, it is sufficient to consider the erasure rule. erasure. We have three cases to distinguish: The erasure of a subgraph of $\mathfrak{G}_1$, the erasure of an edge of $\mathfrak{G}_1$, and the erasure of an isolated vertex of $\mathfrak{G}_1$.

Let $\mathfrak{G}_0$ be a closed subgraph of $\mathfrak{G}_1$ in a positive context, and $\mathfrak{G}_2$ is derived from $\mathfrak{G}_1$ by erasing $\mathfrak{G}_0$. Then there is a corresponding subgraph $\Xi(\mathfrak{G}_0)$ in $\Xi(\mathfrak{G}_1)$, the erasure rule of $\vdash_v$ allows to erase $\Xi(\mathfrak{G}_0)$ from $\Xi(\mathfrak{G}_1^R)$, which yields $\Xi(\mathfrak{G}_2^R)$. Thus Eqn. (24.2) is shown. As isolated vertices are closed subgraphs, the erasure of an isolated vertex in $\mathfrak{G}_1$ can be transferred to the erasure of the corresponding isolated vertex in $\Xi(\mathfrak{G}_1)$ as well.

Now let e be an edge in $\mathfrak{G}_1$ in an isolated vertex. If e is not labeled with a constant name, then there exists a corresponding edge $\Xi(e)$ in $\Xi(\mathfrak{G}_1)$, which can be erased with the erasure rule of $\vdash_v$, i.e. Eqn. (24.2) holds.

So it remains to consider the edges e labeled with an $C \in \mathcal{C}$. Let e be incident with v_e, let v'_e and id_e be the fresh vertex and fresh identity edge which correspond to e in $\Xi(\mathfrak{G}_1)$ (see Def. 24.3). Then we can first erase id_e in $\Xi(\mathfrak{G}_1)$, then we can erase the (now) isolated vertex v'_e, and we obtain $\Xi(\mathfrak{G}_2)$. So Eqn. (24.1) (but not Eqn. (24.2)) holds.

Informally depicted, this procedure can be represented as follows:



- **iteration/deiteration**

  Let $\mathfrak{G}_0 := (V,E,\nu,\top,Cut,area,\kappa)0$ be a subgraph of $\mathfrak{G}_1$ and let $c \leq ctx(\mathfrak{G}_0)$ be a context such that $c \notin Cut_0$, and $\mathfrak{G}_2$ is obtained from $\mathfrak{G}_1$ by iterating $\mathfrak{G}_0$ into c. Then there is a corresponding subgraph $\Xi(\mathfrak{G}_0)$ in $\Xi(\mathfrak{G}_1)$. Moreover, if If $v \in V_0$ with $ctx(v) = ctx(\mathfrak{U})$ is a vertex, and if $w \in V_0$ with $ctx(w) = c$ is a vertex with $v\Theta w$, then we have $\Xi(v)\Theta\Xi(w)$ as well (we even have $\Xi(v)\Theta^*\Xi(w)$, i.e., as already mentioned, it would have been sufficient to consider Θ^* instead of Θ). So we see that $\Xi(\mathfrak{G}_2)$ can be obtained from $\Xi(\mathfrak{G}_1)$ with the iteration rule of $\vdash_v$, i.e., Eqn. (24.2) holds.

  Analogously, if $v \in V$ is a vertex, if $c \leq ctx(v)$ is a cut, and if $\mathfrak{G}_2$ is obtained from $\mathfrak{G}_1$ by inserting a new vertex w and an identity edge between v and w into c with the second clause of the iteration-rule, then $\Xi(\mathfrak{G}_2)$ can be obtained from $\Xi(\mathfrak{G}_1)$ by iterating a copy of $\Xi(v)$ into $\Xi(c)$. Thus Eqn. (24.2) holds again.

- **double cut, erasing a vertex, inserting a vertex, Functional Property Rule, total function rule**

  For these rules it is straightforward to see that Eqn. (24.1) holds.

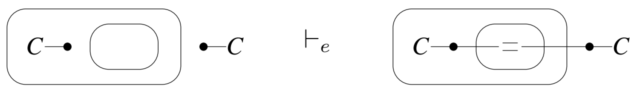
- **Constant Identity Rule**

  Let $\mathfrak{G}_1$ be an EGI, let C be a constant name, let e, f be two unary edges with $\nu(e) = (v_e)$, $\nu(f) = (v_f)$, $ctx(v_e) = ctx(e)$, $ctx(v_f) = ctx(f)$, and

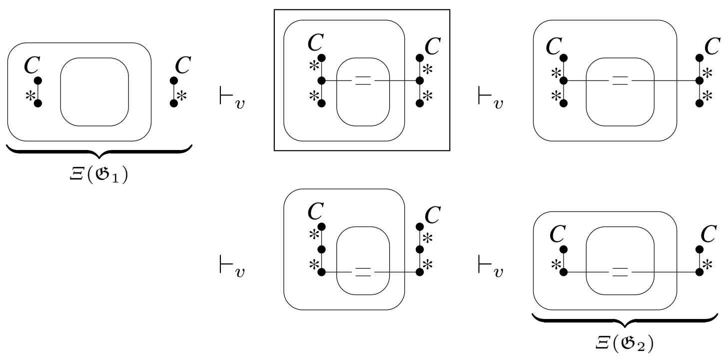
$\kappa(e) = \kappa(f) = C$, let c be a context with $c \leq ctx(e)$ and $c \leq ctx(f)$, and let $\mathfrak{G}_2$ be obtained from $\mathfrak{G}_1$ by inserting an identity edge id into c.

Using the notation from Def. 24.3, let v'_e, id_e be the vertex labeled with C let id_e be the identity edge between v_e and v'_e in $Xi(\mathfrak{G}_1)$ which substitute v_e, and let v'_f, id_f be the corresponding elements for v_f. The constant identity rule allows to add an identity edge $id' = (v'_e, v'_f)$ between v'_e and v'_f in $Xi(\mathfrak{G}_1)$ to c. Now we separate the vertices v'_e and v'_f, replace v'_e by v_e on the hook $(id', 1)$ and replace v'_f by v_f on the hook $(id', 2)$ with Lem. 16.1, and finally we deseparate the vertices v'_e and v'_f. The graph we obtain is $\Xi(\mathfrak{G}_2)$.

Informally depicted with an example, the derivation



with EGIs is replaced by the following derivation with vertex-based EGIs:



The erasure of an identity edge is done analogously.

- **Existence of Constants Rule**:

  If $\mathfrak{G}_2$ is obtained from $\mathfrak{G}_1$ by inserting into a context c a fresh vertex v and an fresh unary edge e with $\nu(e) = (v)$ and $\kappa(e) = C$, we can derive $\Xi(\mathfrak{G}_2)$ from $\Xi(\mathfrak{G}_1)$ by first inserting a constant vertex v with $\rho(v) = C$ into c and then by separating v.

  The erasure of a vertex labeled with C is proven analogously. □

24.4 Ligatures in vertex-based EGIs

In Sec. 16.1, several lemmata had been provided which improved the handling of ligatures. These lemmata had been derived rules, as they have been proven with the iteration- and deiteration-rules of the calculus. Now the question arises how the results of Sec. 16.1 can be transferred to vertex-based EGIs, where ligatures may contain constant vertices as well.

All rules of the calculus for vertex-based EGIs, particularly the iteration- and deiteration-rule, are extensions of the corresponding rules for EGIs. That is, each rule which can be applied to EGIs over an alphabet without constant- or function names can be applied to vertex-based EGIs as well. Thus it is easy to see that the lemmata of Sec. 16.1 can be applied to generic ligatures. Moreover, iff we have a ligature with constant vertices, we can separate all these vertices, and in the resulting graph, the resulting ligature (i.e. the ligature which contains exactly the same vertices and edges, without the fresh vertices and edges we added) is then a generic ligature. In this respect, we can adopt the results of Sec. 16.1 for arbitrary ligatures.

On the other hand, the iteration and deiteration rule rely on the relation Θ. In Def. 24.9 of Θ for vertex-based EGIs, which extends Def. 15.1 of Θ for EGIs, generic and constant vertices are not distinguished, thus we can adopt the proofs of Sec. 16.1 for arbitrary ligatures in vertex-based EGIs. But in these proofs, no application of the iteration or deiteration rule changes the set of labels of the ligature. So, roughly speaking, we can change a ligature with the results of Sec. 16.1, as long as we do not change the set of its labels.

For the sake of convenience, for each lemma of Sec. 16.1, a corresponding lemma for vertex-based EGIs is given. But before we do so, a few new results for ligatures in vertex-based EGIs are provided.

On page 268, when we discussed the first approach for a calculus for vertex-based EGIs, we have seen that it must be possible to 'generalize' the labels of constant vertices to generic markers in positive context (and vice versa, to 'specialize' generic vertices to constant vertices in negative contexts). This can now easily be proven with the calculus for vertex-based EGIs.

Lemma 24.12 (Generalizing and Specializing the Labels of Vertices). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an vertex-based EGI. If $v \in V^{\mathcal{C}}$ is a positively enclosed vertex, then the label from v may be changed to $*$. Vice versa, if $v \in V^*$ is a negatively enclosed vertex, then the label of v may be changed to a constant name $c \in \mathcal{C}$.*

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by generalizing the label of the positively enclosed $v \in V^{\mathcal{C}}$. Then $\mathfrak{G}'$ can be derived from $\mathfrak{G}$ by first separating v, and then by erasing the fresh edge e' and the fresh vertex v'. The proof for specializing a label for a negatively enclosed vertex is done analogously. □

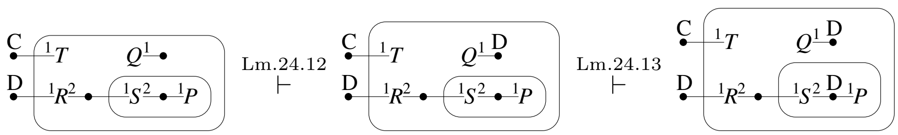
Secondly, it is possible to '(de)iterate constant labels'.

Lemma 24.13 ((De)Iterating Constant Labels). *Let a vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be given, let $v \in V^{\mathcal{C}}$ with $C := \rho(v)$, let $w \in V$ with $ctx(v) \geq ctx(w)$ and $v\Theta w$. If $\rho(w) = C$, then we can change the label of w to $*$. Vice versa, if $\rho(w) = *$, then we can change the label of w to C.*

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by changing the label from w from C to $*$. Then $\mathfrak{G}'$ can be derived from $\mathfrak{G}$ as follows: First, v is iterated into $ctx(w)$. As we have $v\Theta w$, we can add an with an identity edge e' to between v' and w. Now $\mathfrak{G}'$ can be derived from this graph by deseparating v'.

The second proposition of the lemma is proven analogously. □

Next, an example for the last two lemmata is provided. We assume that C, D are constant names and P, Q, R, S, T are relation names.



From the last lemma, we immediately obtain the following corollary.

Corollary 24.14 (Rearranging Labels of a Ligature in a Context). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an vertex-based EGI. Let (W, F) be a ligature which is placed in a context c, i.e., $ctx(w) = c = ctx(f)$ for all $w \in W$ and $f \in F$. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by rearranging the labels of (W, F), i.e. we have $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho')$ with $\rho'\big|_{V\backslash W} = \rho$ and $\rho'(W) = \rho(W)$. Then $\mathfrak{G}$ and $\mathfrak{G}'$ are equivalent.*

Next, the results of Sec. 16.1 (Lem. 16.1, Lem. 16.2, Lem. 16.3 and Def. 16.4) are revised for vertex-based EGIs. The differences of the ongoing lemmata to their counterparts in Sec. 16.1 are, if any, only minor and emphasized by underlining them. For sake of convenience, the lemmata as such are given, but the formal definitions for the transformations, as well as the proofs, are omitted, as they are canonical extensions of the definitions and proofs in Sec. 16.1.

We start with Lemma 16.1. In this lemma, the set of vertices $\rho(W)$ of the considered ligature (W, F) does not change, thus this lemma can be directly adopted for vertex-based EGIs. That is, we obtain:

Lemma 24.15 (Moving Branches along a Ligature in a Context). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be a vertex-based EGI, let v_a, v_b be two vertices with $c := ctx(v_a) = ctx(v_b)$ and $v_a\Theta v_b$, and let e be an edge such that the hook (e, i) is attached to v_a. Let $\mathfrak{G}' := (V, E, \nu', \top, Cut, area, \kappa)$ be obtained from $\mathfrak{G}$ by replacing v_a by v_b on the hook (e, i). Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

Lemma 16.2 allowed to extend or restrict ligatures. More precisely: To a given vertex v of a ligature, new vertices can be attached with identity links, and this transformation can be reversed. For EGIVs, we have to take care that the new vertices are labeled the same as v.

Lemma 24.16 (Extending or Restricting a Ligature in a Context). *Let a vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be given with a vertex $v \in V$. Let V' be a set of fresh vertices and E' be be a set of fresh edges. Let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be obtained from $\mathfrak{G}$ such that all fresh vertices and edges are placed in the context $ctx(v)$, all fresh edges are identity edges between the vertices of $\{v\} \dot\cup V'$ such that we have $v\Theta v'$ for each $v' \in V'$, and we have $\rho(v') = \rho(v)$ for all fresh vertices. Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

Lemma 16.1 and Lemma 16.2 allowed retract a ligature in a context to a single vertex (and vice versa). This is for vertex-based EGIs not possible any more: A ligature (W, F) such that $\rho(W)$ contains more than one constant name cannot be retracted to a single vertex. But if all vertices of W are labeled the same, retracting to a single vertex is still allowed.

Lemma 24.17 (Retracting a Ligature in a Context). *Let a vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be given, let (W, F) be a ligature which is placed in a context c and which satisfies $|\rho(W)| = 1$. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by retracting (W, F) to w_0. Then $\mathfrak{G}$ and $\mathfrak{G}'$ are syntactically equivalent.*

In Def. 16.4, we have summarized the different possibilities to rearrange a ligature in a context, and we concluded from the preceeding lemmata that rearranging a ligature in a context yields equivalent graphs.

For vertex-based EGIs, we first redefine the rearranging of ligatures in the well-known semi-formal manner. The formal elaboration of this definition is a canonical extension of Def. 16.4.

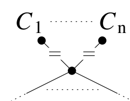
Definition 24.18 (Rearranging Ligatures in a Context). *Let a vertex-based EGI $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ over an alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ be given, let (W, F) be a ligature which is placed in a context c. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ as follows: The ligature (W, F) is replaced by a new ligature (W', F') with $\rho(W) \cap \mathcal{C} = \rho(W') \cap \mathcal{C}$, i.e., all vertices of W and all edges of f are removed from c, the vertices of W' and edges of E' are inserted into c, if an edge $e \in E \backslash F$ was incident with a vertex $w \in W$ of the ligature, it is now connected to a vertex $w' \in W'$ of the new ligature. We say that $\mathfrak{G}'$ is obtained from $\mathfrak{G}$ by* REARRANGING THE LIGATURE (W, F) (TO (W', F')).

For EGIs, is was a direct conclusion from Lem. 16.1–16.3 that rearranging a ligature in a context is a meaning-preserving transformation. For vertex-based EGIs, we have to spend a little further effort to prove a corresponding result.

Theorem 24.19 (Rearranging Ligatures in a Context). *Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by rearranging the ligature (W, F) to (W', F'). Then $\mathfrak{G}$ and $\mathfrak{G}'$ are equivalent.*

Proof: Note that $\rho(W) \cap \mathcal{C} = \rho(W') \cap \mathcal{C}$ is equivalent to $\rho(W)\backslash\{*\} = \rho(W')\backslash\{*\}$. Let us assume $* \notin \rho(W)$. Then by separating w into w and w' and by setting $W' := W \cup \{w'\}$, $F' := F \cup \{(w, w')\}$, we obtain an equivalent graph with a new ligature (W', F') which now satisfies $\rho(W') = \rho(W) \dot\cup \{*\}$. So w.l.o.g. we assume $* \in \rho(W)$.

For $\rho(W) = \{*\}$, we can directly adopt the argumentation after Def. 16.4 to show that $\mathfrak{G}$ and $\mathfrak{G}'$ are equivalent: First retract (W, F) to a single vertex $w_0 \in W$ and then extend w_0 to the new ligature (W', F'). But generally, for $\rho(W) \supsetneq \{*\}$, we cannot retract (W, F) to a single vertex. Instead, we retract (W, F) to the following 'minimal' ligature:



Here, $C_1, \ldots, C_n$ is an enumeration of the constant names in $\rho(W)$, and the edges below the vertex spot shall be the (former) edges between vertices of W and vertices of $V\backslash W$. More formally, we assign to $\mathfrak{G}$ a vertex-based EGI $\mathfrak{G}_r$ as follows: First, for each $C \in \rho(W)\backslash\{*\}$, let w_C be a fresh vertex, and let $w_* \in W$ be an arbitrary generic vertex (the reason not to choose a *fresh* vertex w_* is a simple matter of convenience: It eases the ongoing proof). For each $C \in \rho(W)\backslash\{*\}$, let f_C be a fresh edge. Let $W_r := \{w_r \mid C \in \rho(W)\}$ and $F_r := \{f_r \mid C \in \rho(W)\}$. Now $\mathfrak{G}_r$ is defined as follows:

- $V_r := V\backslash W \cup W_r$
- $E_r := E\backslash F \cup F_r$
- For ν_r is defined as follows:
  - For $e \in E\backslash F$, $\nu(e) = (v_1, \ldots, v_n)$, let $\nu_r(e) := (v'_1, \ldots, v'_n)$, where $v'_i := v_i$, if $v_i \notin W$, and $v'_i = w_*$, if $v_i \in W$.
  - For $f_r \in F_r$, let $\nu_r(f_r) := (w_*, w_r)$.
- $\top_r := \top$ and $Cut_r := Cut$
- $area_r(d) := \begin{cases} area(d) \text{ for } d \neq c \\ area(c)\backslash(W \cup F) \cup (W_r \cup F_r) \text{ for } d = c \end{cases}.$
- $\kappa_r := \kappa\big|_{E\backslash F} \cup F_r \times \{\doteq\}$
- $\rho := \rho\big|_{V\backslash W} \cup \{(w_r, C) \mid C \in \rho(W\})$

We can transform $\mathfrak{G}_r$ into $\mathfrak{G}$ with the following steps:

1. With L. 24.13, we can transform $\mathfrak{G}$ into $\mathfrak{G}_1 := (V, E, \nu, \top, Cut, area, \kappa, \rho_1)$ such that for each $C \in \rho_1(W)$, there is exactly one vertex $w \in W$ with $\rho_1(w) = C$.
2. Let w_* be an arbitrary vertex $w \in W$ with $\rho_1(w) = *$. Now for each $C \in \rho_1(W)\backslash\{*\}$, the vertex $w \in W$ with $\rho_1(w) = C$ is separated into

w and w_C. The identity-link added by the separation shall be denoted f_C. For $W_r := \{w_C \mid C \in \rho(W)\}$ and $F_r := \{f_C \mid C \in \rho(W)\}$, we obtain a vertex-based EGI $\mathfrak{G}_2 := (V_2, E_2, \nu_2, \top, Cut, area_2, \kappa_2, \rho_2)$ with $V_2 = V \cup W_r$ and $E_2 = E \cup F_r$. In this graph, $(W \cup W_r, F \cup F_r)$ is a ligature, having (W, F) as sub-ligature which satisfies $\rho_1(W) = \{*\}$.

3. Finally, (W, F) is retracted to w_* with Lemma 24.17.

The resulting graph is $\mathfrak{G}_r$, and as all steps can be carried out in both directions, we obtain that $\mathfrak{G}$ and $\mathfrak{G}_r$ are equivalent.

We can similarly obtain a graph $\mathfrak{G}'_r$ from $\mathfrak{G}'$ by retracting the ligature (W', F'). Thus the graphs $\mathfrak{G}_r$ and $\mathfrak{G}'_r$ are isomorphic, thus we obtain that $\mathfrak{G}$ and $\mathfrak{G}'$ are equivalent. $\Box$

25

Relation Graphs

In the last two chapters, we added object- and function-names as new syntactical elements to Peirce's existential graphs. Object- and function-names belong to each standard symbolic form of $\mathcal{FO}$, and the purpose of the last two chapters was to show how Peirce's system has be extended in order to encompass these new elements. But Peirce's himself considered only relations (not objects or functions) as the constituting elements of existential graphs. This is no accident or gap: As already been mentioned in the introduction, Peirce was convinced that relations are the most elementary elements of reasoning. In this chapter, EGIs are extended so that they describe relations. Thus the resulting graphs – relation graph instances (RGIs) – can still be understood as a formal elaborations of Peirce's diagrammatic logic.

Burch elaborates in [Bur91a] a formal algebraic system from which he claims that it is an 'either accurate or at least approximate representation of Peirce's thinking' (page viii). This system is called *Peircean Algebraic Logic* (PAL). But although the elaboration of PAL is driven by the diagrammatic representation of relations, Burch develops a linear and algebraic notation for PAL. Not until the last chapter of his book it is roughly discussed how this linear notation is related to its diagrammatic representation. The relation graphs of this treatise can be understood as a formal, diagrammatic system for representing PAL by means of graphs (to be more precise: The relation graphs cover PAL, but as it will turn out in the next section, not every relation graph has a corresponding PAL-term).

Peirce's understanding of relations is still disputed among Peirce experts. Nowadays, relations are understood in a purely extensional manner as sets of tuples. Burch argues that this view does not suffice for Peirce's conception of relations, and he develops an extensional *and intensional* semantics for relations. Zeman acknowledges in [Zem95] very much this intensional semantics, but he advocates that Peirce had the extensional view on relations as well. Indeed, there are several places in Peirce's writings where he describes dyadic relations as classes of pairs. But Peirce's view cannot be nailed down to this understanding: In other places, Peirce provides different descriptions for relations (see below). The formal relation graphs whill will be developed in this chapter suite both the extensional and the intensional interpretation of relations, as they are elaborated by Burch.

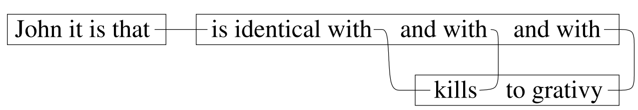
Besides the extensional view, Peirce often described a relation as a proposition with free places called *blanks*. In 3.465, he writes (in this and the following quotation, emphasis is done by Peirce).

> In a complete proposition there are no blanks. It may be called a *medad*, or *medadic relative* [...] A non-relative name with a substantive verb, as ”– is a man,” or ”man that is –” or ”–'s manhood” has one blank; it is a *monad*, or *monadic relative*. An ordinary relative with an active verb as ”– is a lover of –” or ”the loving by – of –” has two blanks; it is a *dyad*, or *dyadic relative*. A higher relative similarly treated has a plurality of blanks. It may be called a *polyad*.

Shortly after this, he concludes in 3.466:

> A *relative*, then, may be defined as the equivalent of a word or phrase which, either as it is (when I term it a *complete* relative), or else when the verb ”is” is attached to it (and if it wants such attachment, I term it a *nominal* relative), becomes a sentence with some number of proper names left blank.

We have already seen an example for this view on page 103, where the relations ”– kills – to gratify –” or ”John is –” are diagrammatically depicted. For sake of convenience, Figure 4 of 3.471 is repeated:



In this example, it becomes clear that the blanks of a relation correspond to the hooks in EGIs. In Peirce's existential graphs, they have to be filled by lines of identity, resp. in EGIs, vertices are attached to the hooks.

It is well known that the diagrams from chemistry for atoms and molecules are a main inspiration for the diagrammatic form of existential graphs. For example, in 3.421 Peirce writes that '*A rhema is somewhat closely analogous to a chemical atom or radicle with unsaturated bonds.* ', and in 3.460 we find '*A chemical atom is quite like a relative in having a definite number of loose ends or ”unsaturated bonds,” corresponding to the blanks of the relative*'. In fact, in 4.471 he writes that Figure 4 '*correspond to prussic acid as shown in Figure 5.*' (see Fig. 25.1).

The similarity to chemistry is not only the diagrammatic representation of relations. Atoms of molecules with free valences can be compound to new

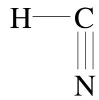


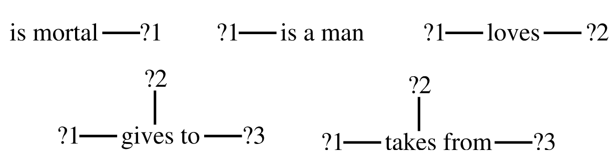
Fig. 25.1. Fig. 5 of 4.471

molecules. Similarly, relations can be *joined* (the join of relations is one of the operations of PAL). This exemplifies Peirce in 3.421 as follows:

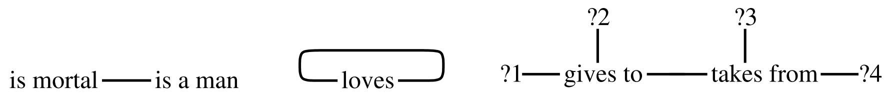
> So, in chemistry, unsaturated bonds can only be saturated by joining two of them, which will usually, though not necessarily, belong to different radicles. If two univalent radicles are united, the result is a saturated compound. So, two non-relative rhemas being joined give a complete proposition. Thus, to join "– is mortal" and "– is a man," we have "X is mortal and X is a man," or some man is mortal. So likewise, a saturated compound may result from joining two bonds of a bivalent radicle; and, in the same way, the two blanks of a dual rhema may be joined to make a complete proposition. Thus, "– loves –" "X loves X," or something loves itself.

Similarly, the graph of Figure 4 of 4.471 is the result of appropriately joining the relations "John it is that –", "– is identical with – and with – and with –", and "– kill – to gratify –". In joining relations, it is allowed that blanks are left, i.e., the result does not need to be a proposition: It might be a relation again. An example by Peirce can be found in 3.421 as well, where Peirce writes: '*Thus, "– gives – to –" and "– takes – from –" give "– gives – to somebody who takes – from –," a quadruple rhema.*"

In this chapter, EGIs are extended by a new syntactical element which is used to denote the blanks (or *loose ends*, as Peirce also calls them) of (possibly compound) relation. These new syntactical elements will be devices query markers with an index i, i.e., devices '?i'. For example, the relations "– is mortal", "– is a man,", "– loves –" "– gives – to –" and "– takes – from –" can de (so far informally) depicted as follows:

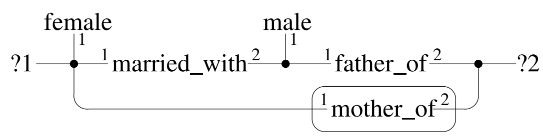


So the results of joining these relations as described by Peirce ("X is mortal and X is a man," or some man is mortal; "X loves X," or something loves itself; and "– gives – to somebody who takes – from –,") yield the following graphs:



In this chapter, relation graphs will be formally elaborated. This will done by means of relation graph instances, which are an extension of EGIs in the understanding of Chpt. 23 (i.e., EGIs over an extended alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$). This is done by additionally allowing that edges to be labeled with new names '$?i$'. Adding these names to vertex-based EGIs (in the understanding of Chpt. 24), i.e. by allowing that vertices to be labeled with new names '$?i$', could be done in a similar manner.

To provide a simple, so-far informal example, we consider the following EGI with the query markers ?1 and ?2. This graph describes the 2-ary relation `is_stepmother_of`.



These graphs will be called *relation graph instances (RGIs)*.[1] We will even evaluate RGIs without query markers to relations, namely to one of the two 0-ary relations $\{\}$ and $\{\{\}\}$. The 0-ary relation $\{\}$ can be identified with the truth-value `tt`, the 0-ary relation $\{\{\}\}$ can be identified with the truth-value `ff`. Thus this approach is not a change, but an extension of the semantics for EGIs, i.e., relation graphs are not an substitute, but an extension of existential graphs.

In the next two sections, two different kinds of relation graphs are introduced. In Sec. 25.1, a general form of relation graph instances called called *semi relation graph instances* is defined. Roughly speaking, they are EGIs were we additionally allow edges to be labeled with the query markers $?i$. These edges correspond to the loose ends, i.e. blanks, of relations. It will be possible that different edges are labeled with the same query marker, and it is moreover allowed that these edges are in the area of cuts. For Peirce, each loose end of a relation graph corresponds to the unsaturated bond of a chemical atom or radicle. So for him, a loose end may appear only once, and it must placed on the sheet of assertion. For this reason, the following graphs are to general to encompass Peirce's notion of relation graphs. Due to this, they are called *semi* relation graph instances. For semi RGIs, the syntax, semantics, and a sound and complete calculus is provided.

In Sec. 25.2, the graphs are syntactically restricted such that a query marker $?i$ may only appear once, and this must be on the sheet of assertion. These

[1] Please do not mistake relation graphs with the underlying structure of all graphs we consider, i.e., relation*al* graphs (with cuts).

graphs are much closer to Peirce's notion of relation graphs then the general form of the first section. The restriction does not yield a loss of expressiveness, and it is shown that the calculus for semi RGIs remains complete.

Other authors already worked on graphs describing relations too. For example, for their framework of Relational Logic and inspired by the work of Burch, Pollandt and Wille invented and investigated such graphs as well (see [Pol01, Pol02, Wil01]). Their graphs are termed *relation graphs* as well, and similar to the graphs in this treatise, they are graph-based formalizations of Peirce's graphs. Similar to Pollandt and Wille, Hereth-Correia and Pöschel in [HCP04, HCP06] and Hereth-Correia and myself in [DHC06] investigated such graphs. But I want to stress that that in these works, different formalizations (than in this treatise) for the graphs are introduced.

All the these works mainly focus on the expressiveness of the graphs and on *operations* on the graphs. That is, they investigate the *algebra* of relations and graphs. Some of them extend and prove Peirce's famous *reduction thesis*, as it is elaborated by Burch in [Bur91a]. In this chapter, we will focus on the *logic* of relation graphs. The next chapter will turn to the algebra of relation graphs, and using the result of [HCP06], a relation graph version of Peirce's reduction thesis, which strictly extends the result of Burch, will be provided.

25.1 Semi Relation Graph Instances

In this section, the syntax, semantics, and calculus for semi relation graph instances are introduced. The basic idea is to extend the alphabet by additional signs $?1, ?2, \ldots$ as free variables. Syntactically, we will treat these signs sometimes like object names, sometimes like free variables. They are first incorporated into an alphabet, then semi relation graph instances are defined.

Definition 25.1 (Extension of Alphabets). *Let $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ be an alphabet. Let $\mathcal{A}^? := (\mathcal{C}^?, \mathcal{F}, \mathcal{R}, ar)$ with $\mathcal{C}^? := \mathcal{C} \dot{\cup} \{?1, ?2, \ldots \mid n \in \mathbb{N}\}$ (we assume $\mathcal{C} \cap \{?1, ?2, \ldots \mid n \in \mathbb{N}\} = \emptyset$). We call $\mathcal{A}^?$ the* QUERY-MARKER-EXTENSION OF $\mathcal{A}$.

Definition 25.2 (Semi Relation Graph Instances). *Let $\mathcal{A}$ be an alphabet and $\mathcal{A}^?$ its query-marker-extension.*

For an EGI $\mathfrak{G}$ over $\mathcal{A}^?$, we set $FV(\mathfrak{G}) := \{?i \mid \exists e \in E : \kappa(e) = ?i\}$. Let $E^? := \{e \in E \mid \kappa(e) = ?i$ for an $i \in \mathbb{N}\}$. Each edge $e \in E^?$ is called PENDING EDGE.

A SEMI RELATION GRAPH INSTANCE (SEMI RGI) OVER $\mathcal{A}$ *is an EGI $\mathfrak{G}$ over $\mathcal{A}^?$ such that there is an n with $FV(\mathfrak{G}) = \{?1, ?2, \ldots, ?n\}$. We set $n := ar(\mathfrak{G})$ and call n the* ARITY OF $\mathfrak{G}$.

For an edge e with $ar(e) = 1$, let v_e be the vertex incident with e.

If we have an alphabet $\mathcal{A}$ and its query marker extension $\mathcal{A}^?$, the query markers have a double syntactical meaning:

- We can consider EGIs over $\mathcal{A}^?$. Then the query markers are object names.
- We can consider semi RGIs over $\mathcal{A}$. Then the query markers are treated as new syntactical names –besides the names for constants, relations, or functions– denoting free valences of an semi RGI.

In the following, the intended understanding will be clear from the context.

There are several ways to define n-tuples, and hence relations, over a ground set U. For the following definitions of the semantics of relation graphs, and particularly for the next chapter, it is useful to provide a precise definition of tuples. Moreover, for the definition of the semantics it is helpful to consider relations, where the tuples are not finite, but infinite sequences.

Definition 25.3 (Relation). *An I-ARY RELATION OVER U is a set $\varrho \subseteq U^I$, i. e. a set of mappings from I to U.*

If we have $I = \{1, \ldots, n\}$, we write $\varrho \subseteq U^n$ instead, and $n := ar(\varrho)$ is now called the ARITY *of ϱ.*

Each relation over a set $I = \{1, \ldots, n\}$ with $n \in \mathbb{N}$ is called FINITARY RELATION*. The set of all finitary relations over a set U is denoted by $\mathsf{Rel}(U)$.*

In the following, we will use the letters ϱ and σ to denote finitary relations. Moreover, we will usually consider only finitary relations, i.e., the attribute 'finitary' is often omitted. When we have to distinguish between finitary and non-finitary relations, this will be explicitely mentioned.

The models for $\mathcal{A}^?$ are the models for $\mathcal{A}$, where we additionally assign objects to the query markers. This is fixed by the following definition.

Definition 25.4 (Extension of Models). *Let $\mathcal{M} := (U, I)$ be a model for $\mathcal{A}$. An* ASSIGNMENT FOR THE QUERY MARKERS *is a mapping $f : \{?1, ?2, \ldots \mid n \in \mathbb{N}\} \to U$.*

Then let $\mathcal{M}[f] := (U, I^f)$ be the model for $\mathcal{A}^?$ with:

$$I^f := (I^f_{\mathcal{C}}, I_{\mathcal{F}}, I_{\mathcal{R}}) \text{ , where } I^f_{\mathcal{C}}(c) := \begin{cases} I_{\mathcal{C}}(c) \text{ for } c \in \mathcal{C} \\ f(c) \text{ for } c = ?i \text{ , } i \in \mathbb{N} \end{cases}$$

$\mathcal{M}[f]$ is called AN QUERY-MARKER-EXTENSION OF *$\mathcal{M}$, and $\mathcal{M}$ is called* THE QUERY-MARKER-RESTRICTION OF *$\mathcal{M}[f]$.*

If $\mathcal{M}$ is a model for $\mathcal{A}$ and $f : \{?1, ?2, \ldots \mid n \in \mathbb{N}\} \to U$, then $\mathcal{M}[f]$ is obviously an model for $\mathcal{A}^?$. Moreover, each model for $\mathcal{A}^?$ is the query-marker-extension of a model for $\mathcal{A}$.

Based on the notation of extension of models, we can now easily define the semantics of semi RGIs.

Definition 25.5 (Semantics for Semi Relation Graph Instances). *Let $\mathfrak{G}$ be a semi RGI over $\mathcal{A}$. We set:*

$$\mathfrak{R}_{\mathcal{M},\mathfrak{G}} := \{f : \mathbb{N} \to U \mid \mathcal{M}[f] \models \mathfrak{G}\}$$

Now let $\mathfrak{H} \cup \{\mathfrak{G}\}$ be a set of semi RGIs. We set

$$\mathfrak{H} \models_? \mathfrak{G} \quad :\Longleftrightarrow \quad \text{for all models } \mathcal{M} \text{ we have } \bigcap_{\mathfrak{G}' \in \mathfrak{H}} \mathfrak{R}_{\mathcal{M},\mathfrak{G}'} \subseteq \mathfrak{R}_{\mathcal{M},\mathfrak{G}}$$

In this definition, for the case $\mathfrak{H} = \emptyset$, we use the convention $\bigcap \emptyset := U^{\mathbb{N}}$. Particularly, we write $\mathcal{M} \models_? \mathfrak{G}$ if we have $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = U^{\mathbb{N}}$.

In this definition, we used sets of query marker assignments, i.e., $\mathbb{N}$-ary relations, instead of finitary relations for a mere matter of convenience. But for a single graph $\mathfrak{G}$, the relation $\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$ can be understood to be a finitary relation with arity $ar(\mathfrak{G})$. We have the following property: If $\mathfrak{G}$ is an EGI over $\mathcal{A}^?$ with $n := ar(\mathfrak{G})$, if $\mathcal{M}$ is a model for $\mathcal{A}$ and if f, g are two query marker assignments with $f\big|_{\{1,\ldots,n\}} = g\big|_{\{1,\ldots,n\}}$, then we have

$$\mathcal{M}[f] \models \mathfrak{G} \quad \Longleftrightarrow \quad \mathcal{M}[g] \models \mathfrak{G}$$

So for $\varrho := \{f\big|_{\{1,\ldots,n\}} \mid f \in \mathfrak{R}_{\mathcal{M},\mathfrak{G}}\}$, we obtain a finitary relation ϱ with arity $ar(\mathfrak{G})$ such that

$$\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = \{f : \mathbb{N} \to U \mid f\big|_{\{1,\ldots,n\}} \in \varrho\}$$

So we can have the following alternative definition for $\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$:

$$\begin{aligned} \mathfrak{R}_{\mathcal{M},\mathfrak{G}} &= \{(u_1, \ldots, u_n) \mid \mathcal{M}[f] \models \mathfrak{G} \text{ for all } f \text{ with } (u_1, \ldots, u_n) \in f\big|_{\{1,\ldots,n\}}\} \\ &= \{f\big|_{\{1,\ldots,n\}} \mid \mathcal{M}[f] \models \mathfrak{G}\} \end{aligned}$$

In the following, we will implicitly identify the infinitary relation $\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$ and the n-ary relation ϱ. i.e., $\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$ will refer to both the finitary and infinitary relation described by $\mathfrak{G}$. It will be clear from the context which meaning is intended. But note that the relation $\bigcap_{\mathfrak{G}' \in \mathfrak{H}} \mathfrak{R}_{\mathcal{M},\mathfrak{G}'}$ of Def. 25.5 is usually not a finitary relation.

If we have an EGI $\mathfrak{G}$ over an extended alphabet $\mathcal{A}^?$, there are two possible ways to evaluate it in a model: We can understand it as EGI over $\mathcal{A}^?$, thus using the entailment-relation $\models$, or we understand it as semi RGI over $\mathcal{A}$, thus using the entailment-relation $\models_?$. Of course, there is an intimate relationship between these two evaluations, which is made precise through the following lemma.

Theorem 25.6 (Semantical Entailment is Preserved for Semi Relation Graphs Instances). *Let $\mathfrak{H} \cup \{\mathfrak{G}\}$ be a set of semi RGIs over an alphabet $\mathcal{A}$. Then we have:*

$$\text{(semi RGIs over } \mathcal{A}) \qquad \mathfrak{H} \models_? \mathfrak{G} \quad \Longleftrightarrow \quad \mathfrak{H} \models \mathfrak{G} \qquad \text{(EGIs over } \mathcal{A}^?)$$

Proof: $\mathfrak{H} \models_? \mathfrak{G}$ (semi RGIs over $\mathcal{A}$)

$\iff$ for all models $\mathcal{M} := (U, I)$ over $\mathcal{A}$: $\bigcap_{\mathfrak{G}' \in \mathfrak{H}} \mathfrak{R}_{\mathcal{M},\mathfrak{G}'} \subseteq \mathfrak{R}_{\mathcal{M},\mathfrak{G}}$

$\iff$ for all models $\mathcal{M} := (U, I)$ over $\mathcal{A}$, for all $f : \mathbb{N} \to U$:

$(\forall \mathfrak{G}' \in \mathfrak{H} : f \in \mathfrak{R}_{\mathcal{M},\mathfrak{G}'}) \Rightarrow f \in \mathfrak{R}_{\mathcal{M},\mathfrak{G}}$

$\iff$ for all models $\mathcal{M} := (U, I)$ over $\mathcal{A}$, for all $f : \mathbb{N} \to U$:

$(\forall \mathfrak{G}' \in \mathfrak{H} : \mathcal{M}[f] \models \mathfrak{G}') \Rightarrow \mathcal{M}[f] \models \mathfrak{G}$

$\iff$ for all models $\mathcal{M}^?$ over $\mathcal{A}^?$:

$(\forall \mathfrak{G}' \in \mathfrak{H} : \mathcal{M} \models \mathfrak{G}') \Rightarrow \mathcal{M} \models \mathfrak{G}$

$\iff$ for all models $\mathcal{M}^?$ over $\mathcal{A}^?$:

$\mathcal{M} \models \mathfrak{H} \Rightarrow \mathcal{M} \models \mathfrak{G}$

$\iff \mathfrak{H} \models \mathfrak{G}$ (EGIs over $\mathcal{A}^?$) □

In the syntax, and in the semantic as well, query markers are basically handled like new object names. If we carry over this idea to the transformation rules, we obtain an adequate calculus. First of all, the calculus is defined as follows:

Definition 25.7 (Calculus for Semi Relation Graph Instances). *The calculus for semi RGIs over $\mathcal{A}$ consists of the rules for EGIs over $\mathcal{A}^?$, where*

- *the query markers $?i$ are treated exactly like object names, and*
- *a rule may only be applied to a semi RGI $\mathfrak{G}$ over $\mathcal{A}$, if the result $\mathfrak{G}'$ is again a well-formed semi RGI over $\mathcal{A}$, i.e., if there is an n with $FV(\mathfrak{G}') = \{?1, ?2, \ldots, ?n\}$.*

We will use the sign '$\vdash_?$' to refer to this calculus.

Let two semi RGIs $\mathfrak{G}_a, \mathfrak{G}_b$ over an alphabet $\mathcal{A}$ with $\mathfrak{G}_a \models_? \mathfrak{G}_b$ be given. Each proof for $\mathfrak{G}_a \vdash_? \mathfrak{G}_b$ is a sequence of semi RGIs over $\mathcal{A}$, so it is a proof for $\mathfrak{G}_a \vdash \mathfrak{G}_b$ in the system of EGIs over $\mathcal{A}^?$ as well. On the other hand, in a proof for $\mathfrak{G}_a \vdash \mathfrak{G}_b$ in the system of EGIs over $\mathcal{A}^?$, there might be EGIs over $\mathcal{A}^?$ which are no semi RGIs over $\mathcal{A}$, as they violate the condition $\mathrm{FV}(\mathfrak{G}) = \{?1, ?2, \ldots, ?n\}$. Nonetheless, by augmenting the graphs in the proof with some query vertices, we can easily obtain a proof in the system of semi RGIs over $\mathcal{A}$. This idea will be used in the next theorem.

Theorem 25.8 (Soundness and Completeness of $\vdash_?$). *Let an alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ be given. Then we have:*

- *The calculus $\vdash_?$ is adequate for the system of all semi RGIs over $\mathcal{A}$.*
- *Given a fixed number n, the calculus $\vdash_?$ is adequate for the system of all semi RGIs $\mathfrak{G}$ over $\mathcal{A}$ with $ar(\mathfrak{G}) = n$.*

Proof: Let first be two semi RGIs $\mathfrak{G}_a, \mathfrak{G}_b$ be given with $\mathfrak{G}_a \vdash_? \mathfrak{G}_b$. Seen as EGIs over $\mathcal{A}^?$, we have $\mathfrak{G}_a \vdash \mathfrak{G}_b$. As $\vdash$ is sound, we obtain $\mathfrak{G}_a \models \mathfrak{G}_b$, thus Lem. 25.6 yields $\mathfrak{G}_a \models_? \mathfrak{G}_b$. That is, $\vdash_i$ is sound.

Now let $\mathfrak{H} \cup \{\mathfrak{G}\}$ be a set of semi RGIs over an alphabet $\mathcal{A}$ with $\mathfrak{H} \models_? \mathfrak{G}$. Then there are $\mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H}$ with $\mathfrak{G}_{\mathfrak{H}} \vdash \mathfrak{G}$. where $\mathfrak{G}_{\mathfrak{H}}$ is the juxtaposition of $\mathfrak{G}_1, \ldots, \mathfrak{G}_n$. Let $FV := \mathrm{FV}(\mathfrak{G}) \cup \mathrm{FV}(\mathfrak{G}_{\mathfrak{H}})$. Considered as EGIs over the alphabet $\mathcal{A}^?$, we have a proof

$$(\mathfrak{G}_{\mathfrak{H}} := \mathfrak{G}_1, \mathfrak{G}_2, \ldots \mathfrak{G}_n := \mathfrak{G})$$

for $\mathfrak{G}_{\mathfrak{H}} \vdash \mathfrak{G}$ with $\mathrm{FV}(\mathfrak{G}_k) \subseteq FV$ for each $k = 1, \ldots, n$. Now for $i = 1, \ldots n$, let $\mathfrak{G}'_i$ be the graph obtained from $\mathfrak{G}'$ by adding for each i with $?i \in FV$ a device $\bullet\!\!-\!\!-?i$ to the sheet of assertion. Then $(\mathfrak{G}'_1, \mathfrak{G}_2, \ldots \mathfrak{G}'_n)$ is again a proof, and we have $\mathrm{FV}(\mathfrak{G}_k) = FV$ for each $k = 1, \ldots, n$.

If we have $\mathrm{FV}(\mathfrak{G}_{\mathfrak{H}}) \subsetneq FV$, e.g. $FV = \mathrm{FV}(\mathfrak{G}_{\mathfrak{H}}) \,\dot\cup\, \{?l, ?(l+1), \ldots, ?m\}$, we can derive $\mathfrak{G}'_1$ from $\mathfrak{G}_1$ with the existence of constants rule by adding successively the devices $\bullet\!\!-\!\!-?l$, $\bullet\!\!-\!\!-?l{+}1$, ..., $\bullet\!\!-\!\!-?m$ to the sheet of assertion.

If we have $\mathrm{FV}(\mathfrak{G}) \subsetneq FV$, e.g. $FV = \mathrm{FV}(\mathfrak{G}) \,\dot\cup\, \{?l, ?(l+1), \ldots, ?m\}$, we can derive $\mathfrak{G}_n$ from $\mathfrak{G}'_n$ with the existence of constants rule the devices by removing successively the devices $\bullet\!\!-\!\!-?m$, $\bullet\!\!-\!\!-?m{-}1$, ..., $\bullet\!\!-\!\!-?m$ from the sheet of assertion.

So we have a proof for $\mathfrak{G}_{\mathfrak{H}} \vdash \mathfrak{G}'_1$, for $\mathfrak{G}'_1 \vdash \mathfrak{G}'_n$, and for $\mathfrak{G}'_n \vdash \mathfrak{G}$, thus for $\mathfrak{G}_{\mathfrak{H}} \vdash \mathfrak{G}$. Each graph $\mathfrak{G}_i$ in this proof is an RGI over $\mathcal{A}$. Moreover, due to the construction of the proof, for $\mathrm{FV}(\mathfrak{G}_{\mathfrak{H}}) = \mathrm{FV}(\mathfrak{G}) = FV$, we have $\mathrm{FV}(\mathfrak{G}_i) = FV$ for each graph $\mathfrak{G}_i$. This proves both completeness claims of the theorem. □

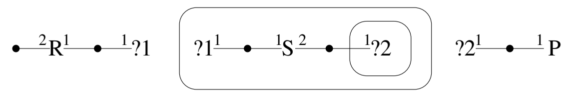
25.2 Relation Graph Instances

Semi RGIs evaluate to relations. For Peirce, to each blank of the relation, we have exactly one pending edge, and this edge is placed on the sheet of assertion. In the following, we restrict the system of RGIs in order to get a class of graphs which corresponds more closely to Peirce's understanding.

Definition 25.9 (Relation Graph Instances). *Let $\mathcal{A}$ be an alphabet. A* RELATION GRAPH INSTANCE (RGI) $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ OVER $\mathcal{A}$ *is a semi RGI over $\mathcal{A}$ which satisfies $ctx(e) = \top$ for each $e \in E^?$ and $|\{e \in E^? \mid \kappa(e) = ?i\}| = 1$ for each $?i \in FV\mathfrak{G})$.*

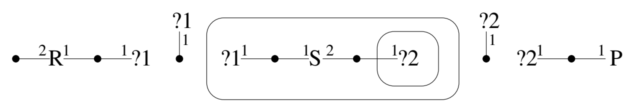
RGIs can be understood as semi RGIs in some normal form. In order to emphasize that a graph is an RGI, not only a semi RGI, we will call it sometimes NORMED. Now we assign to each semi RGI $\mathfrak{G}$ a normed RGI $norm(\mathfrak{G})$, the *normalization* of $\mathfrak{G}$. The normalization is first described as a procedure using the rules of the calculus, and it is exemplified with one graph. Then the formal definition is given.

1. Our example is the graph

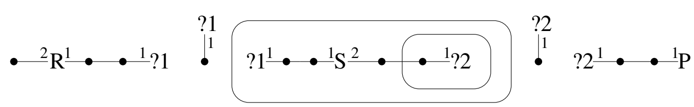


   In the diagram, for unary edges, we omit to label the edge-line with 1.

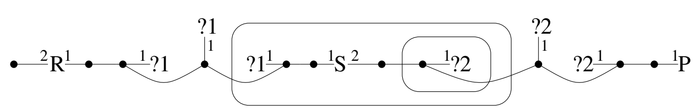
2. For each $?i \in \mathrm{FV}(\mathfrak{G})$, we add a new vertex $v_{?i}$ and a new edge $e_{?i}$ with $\nu(e_{?i}) = (v_{?i})$ and $\kappa(e_{?i}) = ?i$ to the sheet of assertion. Within the calculus, this is done with the existence of constants rule.



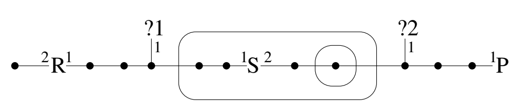
3. For each edge e with $\kappa(e) = ?i$ and $e \neq e_{?i}$ for a $?i \in \mathrm{FV}(\mathfrak{G})$, we use the rule 'adding a vertex to a ligature' to add a new vertex w_e to $ctx(e)$. The new identity edge between $v(e)$ (the vertex incident with e in $\mathfrak{G}$) and w_e will be denoted id_e^1.



4. For each edge e with $\kappa(e) = ?i$ and $e \neq e_{?i}$ for a $?i \in \mathrm{FV}(\mathfrak{G})$, an identity link id_e^2 between $v_{?i}$ and w_e is added to $ctx(e)$ with the constant identity rule.



5. Finally, each edge e with $\kappa(e) = ?i$ for a $?i \in \mathrm{FV}(\mathfrak{G})$ and $e \neq e_{?i}$ is removed. Within the calculus, this can be done as follows: First, an new vertex w'_e and a new identity-link id'_e between w_e and w'_e is added to to $ctx(e)$ with the rule 'adding a vertex to a ligature'. Then the subgraph consisting of e and w'_e is deiterated, and in this deiteration, the identity-link id'_e is removed as well.



Now the formal definition of the normalization of a semi RGI is provided.

Definition 25.10 ($\boldsymbol{norm(\mathfrak{G})}$). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be a semi RGI.*

- *For each $?i \in FV(\mathfrak{G})$, let $v_{?i}$ be a fresh vertex and $e_{?i}$ be a fresh edge.*
- *For each $?i \in FV(\mathfrak{G})$ and $e \in E^{?i}$, let let w_e be a fresh vertex and id_e^1, id_e^2 be fresh edges. We set $W_n := \{w_e \mid e \in E^?\}$, $F_n^1 := \{id_e^1 \mid e \in E^?\}$ and $F_n^2 := \{id_e^2 \mid e \in E^?\}$.*

Now we define $norm(\mathfrak{G}) := (V_n, E_n, \nu_n, \top_n, Cut_n, area_n, \kappa_n)$ as follows (note that $v(e)$ in the definition of ν_n is the vertex incident with e in $\mathfrak{G}$):

- $V_n := V \mathbin{\dot\cup} W_n \mathbin{\dot\cup} \{v_{?i} \mid ?i \in FV(\mathfrak{G})\}$
- $E_n := E \backslash E^? \mathbin{\dot\cup} F_n^1 \mathbin{\dot\cup} F_n^2 \mathbin{\dot\cup} \{e_{?i} \mid ?i \in FV(\mathfrak{G})\}$
- $\nu_n := \nu\big|_{E \backslash E^?} \mathbin{\dot\cup} \{(w_e, v(e)), (w_e, v_{?i}) \mid e \in E^?\}$
- $\top_n := \top$,
- $Cut_n := Cut$,
- *$area_n$ is defined as follows: For $c \in Cut$ we set*

$$area_n(c) := area(c) \backslash (E^? \cap area(c)) \mathbin{\dot\cup} \{w_e, id_e^1, id_e^2 \mid e \in E^? \cap area(c)\}, \text{ and}$$
$$area_n(\top) := area(\top) \backslash (E^? \cap area(\top)) \mathbin{\dot\cup} \{w_e, id_e^1, id_e^2 \mid e \in E^? \cap area(\top)\} \mathbin{\dot\cup} \{v_{?i}, e_{?i} \mid ?i \in FV(\mathfrak{G})\}$$

- $\kappa_n := \kappa\big|_{E \backslash E^?} \mathbin{\dot\cup} \{(id_e^1, \doteq) \mid id_e^1 \in F_n^1\} \mathbin{\dot\cup} \{(id_e^2, \doteq) \mid id_e^2 \in F_n^2\}$

$norm(\mathfrak{G})$ is called THE NORMALIZATION OF $\mathfrak{G}$.

As we have seen, in the transformation of $\mathfrak{G}$ to its normalization we used only rules of the calculus which can be carried out in both directions. Thus we immediately obtain the following lemma:

Lemma 25.11 ($\mathfrak{G}$ and $\boldsymbol{norm(\mathfrak{G})}$ are Equivalent). *Let $\mathfrak{G}$ be a semi RGI. Then $norm(\mathfrak{G})$ is an RGI which is syntactically equivalent to $\mathfrak{G}$.*

Without proof

Due to Lemma 25.11, the full system of semi RGIs is semantically equivalent to the restricted system of RGIs. Moreover, it is clear that the calculus is still sound, if consider the restricted system of RGIs. But if we apply one of the rules iteration/deiteration, erasure/insertion, or the constant identity rule to a normed RGI, we might obtain an semi RGI which is not normed. Thus it is not clear that the calculus is still complete. The next theorem shows that that we do not loose the completeness of the calculus.

Theorem 25.12 (Completeness of $\vdash_?$ for RGIs). *Let $\mathfrak{G}, \mathfrak{G}'$ be two semi RGIs such that $\mathfrak{G}'$ is derived from $\mathfrak{G}$ by applying one of the rules of the calculus $\vdash_?$. Then we have $norm(\mathfrak{G}_a) \vdash_? norm(\mathfrak{G}_b)$, where the proof contains only RGIs. That is, the calculus $\vdash_?$ is complete for RGIs.*

Proof: First of all, if r is one of the rules 'isomorphism', 'changing the orientation of an identity edge', 'adding a vertex to a ligature' and 'removing a vertex from a ligature', 'double cuts', 'erasing a vertex' and 'inserting a vertex', 'functional property rule', 'total function rule', and the 'constant identity rule',[2] and if we have $\mathfrak{G} \vdash_?^r \mathfrak{G}'$, it is easy to see that we have $norm(\mathfrak{G}) \vdash_?^r norm(\mathfrak{G}')$ as well.

It remains to consider the rules 'erasure' and 'insertion', 'iteration' and 'deiteration', and the existence of constants rule. First of all, we use the notation of Def. 25.10. Moreover, in the ongoing proof, we will split and merge vertices (see Def. 16.6). In Lemma 16.7, we have shown that splitting or merging vertices transforms a graph in a syntactically equivalent graph by providing a formal proof for the transformation. It can easily be checked that if we start with a normed graph, all graphs in this proof are normed as well. So we are allowed to apply Lemma 16.7 in the ongoing proof.

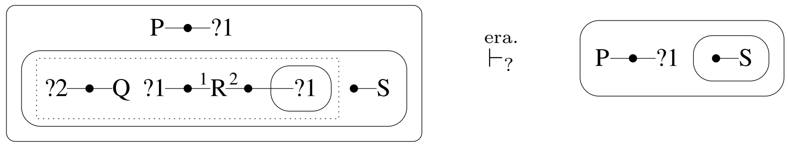
We start with the pair 'erasure' and 'insertion'. Due to the symmetry of the calculus, it is sufficient to prove this theorem only for the erasure rule.

Let first $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by erasing the edge e. If $e \notin E^?$, we have $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$ as well. So let e be an edge with $\kappa(e) = ?i$. In $norm(\mathfrak{G})$, we first erase id_e^1, then id_e^2, and then w_e. If e is the only edge in $\mathfrak{G}$ with $\kappa(e) = ?i$, we furthermore erase the (now closed) subgraph consisting of $v_{?i}$ and $e_{?i}$. The resulting graph is $norm(\mathfrak{G})$, thus we again have $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$.

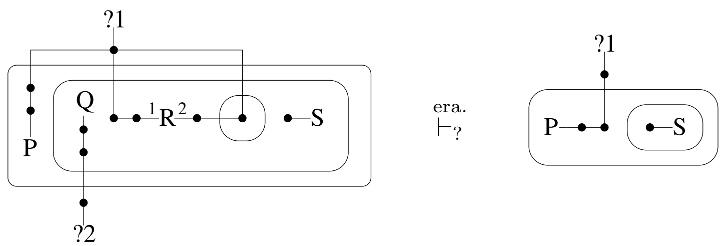
Now let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by erasing the closed subgraph edge $\mathfrak{G}_s := (Vs, Es, \nu s, \top s, Cuts, areas, \kappa s)$ (particularly, $\top_s$ is the context of $\mathfrak{G}_s$). We provide a formal proof for $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$, which will be illustrated by an example.

Our example is given below. The subgraph is indicated by a dotted subgraph-line.

[2] Recall that constants are treated like 1-ary functions, and the constant identity rule and the existence of constants rule are the counterparts of the functional property rule and the total function rule. So one might think that if we have $norm(\mathfrak{G}) \vdash_?^r norm(\mathfrak{G})$ for the rules for functions, this should hold trivially for the rules for constants as well. But on the other hand, as query vertices are exactly treated like constant vertices, and as query vertices are affected by the normalization of graphs, we have to take care for the constant rules. And in fact, it turns out that the existence of constants rule has to be treated separately.

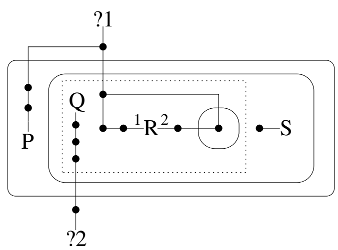


Thus for the normalizations of these graphs, we have to show



First, for each $?i \in \mathrm{FV}(\mathfrak{G}_s)$, we split $v_{?i}$. The new vertex $w_{?i}$ is placed in $\top_s$ (thus we have a new identity edge $id_{?i}$ in $\top_s$ between $v_{?i}$ and $w_{?i}$ as well). It shall be incident with all vertices $w_e \in E^? \cap E_s$.

For our example, this yields:

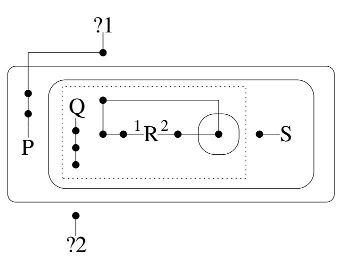


Let us denote this intermediate graph by $\mathfrak{G}^i$. The sets

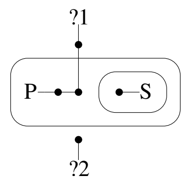
- $V_s^i := V_s \,\dot{\cup}\, \{w_{?i} \mid ?i \in \mathrm{FV}(\mathfrak{G}_s)\} \,\dot{\cup}\, \{w_e \mid e \in E^? \cap E_s\}$ of vertices,
- $E_s^i := E^s \backslash E^? \,\dot{\cup}\, \{id_{?i} \mid ?i \in \mathrm{FV}(\mathfrak{G}_s)\} \,\dot{\cup}\, \{id_e^1, id_e^2 \mid e \in E^? \cap E_s\}$ of edges,
- $Cut_s^i := Cut_s$ of cuts

give rise to a subgraph $\mathfrak{G}_s^i := (V_s^i, E_s^i, \nu_s^i, \top_s^i, Cut_s^i, area_s^i, \kappa_s^i)$ of $\mathfrak{G}^i$ which so-to-speak corresponds to the subgraph $\mathfrak{G}_s$ of $\mathfrak{G}$.

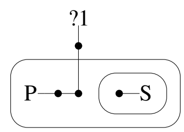
Now, for each $?i \in \mathrm{FV}(\mathfrak{G}_s)$, we erase $id_{?i}$.



We erased only edges of $\mathfrak{G}_s^i$. The remaining subgraph is now closed and can be erased.



Finally, for each $?i \in \mathfrak{G}_s$ with $\{e \in E | \kappa(e) = ?i\} \subseteq E^s$, the subgraph with the vertex $v_{?i}$ and the edge $e_{?i}$ is closed and can erased.

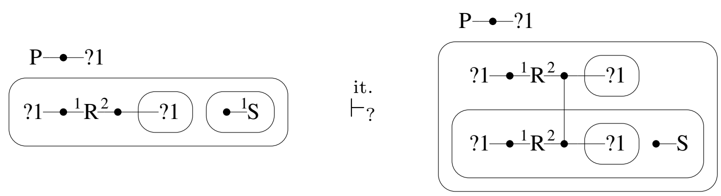


We obtain $norm(\mathfrak{G}')$, thus we are done.

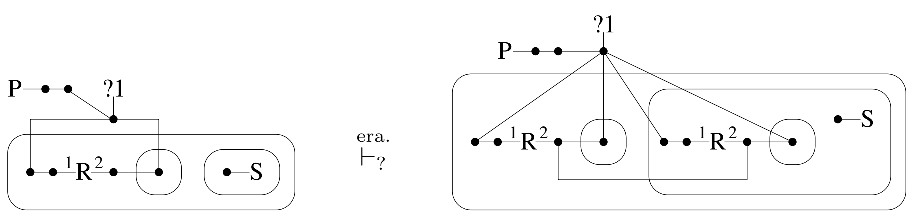
After proving the theorem for the rule 'erasure', thus for 'insertion' as well, we consider next the iteration and deiteration rule. Again due to the symmetry of the calculus, we prove this theorem only for the iteration rule.

Again we provide a formal proof for $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$ and an example. The subgraph $\mathfrak{G}_s$ iterated into d.

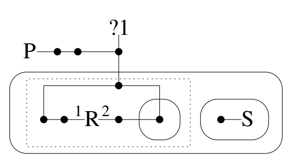
Our example is given below. The subgraph is indicated by a dotted subgraph-line. In the application of the iteration rule, we add one new identity link.



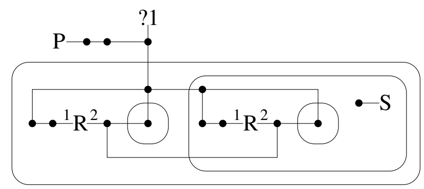
Thus for the normalizations of these graphs, we have to show



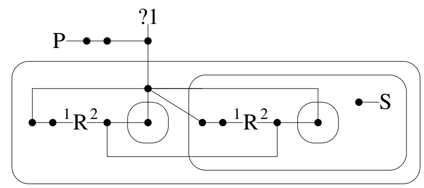
First, for each $?i \in \mathrm{FV}(\mathfrak{G}_s)$, we split $v_{?i}$ exactly like in the proof of the erasure rule. Again, we obtain an intermediate graph by $\mathfrak{G}^i$ with a subgraph $\mathfrak{G}_s^i$.



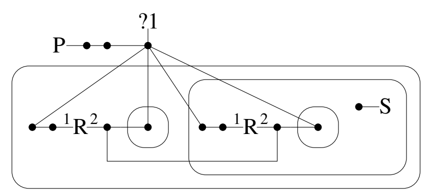
Now the subgraph $\mathfrak{G}_s^i$ is iterated into d similar to the iteration of $\mathfrak{G}_s$ (particularly, if in iteration of $\mathfrak{G}_s$ an identity edge it added between a vertex $v \in V_s$ and its copy v', we do the same in the iteration of $\mathfrak{G}_s^i$). Moreover, for each vertex $w_{?i}$ for an $?i \in \mathrm{FV}(\mathfrak{G}_s)$, we add an identity-link $id'_{?i}$ between $w_{?i}$ and its copy $w'_{?i}$.



For each $?i \in \mathrm{FV}(\mathfrak{G}_s)$, we merge $w'_{?i}$ into $w_{?i}$.



For each $?i \in \mathrm{FV}(\mathfrak{G}_s)$, we merge $w_{?i}$ into $v_{?i}$.

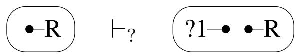


We obtain $norm(\mathfrak{G}')$, thus we are done.

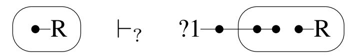
After proving the theorem for the iteration and deiteration rule, we finally prove this theorem for the existence of constants rule. We prove the theorem for adding a fresh vertex v and an fresh unary edge e with $\nu(e) = (v)$ and $\kappa(e) = C$ to a context c, where C is a constant name. We distinguish three cases.

- $C \in \mathcal{C}$ (i.e., C is no query marker). Then we have $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$ as well.
- $C = ?i$ for an $i \in \mathbb{N}$, and $?i \notin \mathrm{FV}(\mathfrak{G})$. We obtain $norm(\mathfrak{G}')$ from $norm(\mathfrak{G})$ as follows: First, we add with the existence of constants rule. a new vertex $v_{?i}$ and a new edge $e_{?i}$ with $\nu(e_{?i}) = (v_{?i})$ and $\kappa(e_{?i}) = ?i$ to the sheet of assertion. Then $v_{?i}$ is iterated into c. The copy of $v_{?i}$ shall be denoted w_e. During the iteration, an identity edge id_e^2 is added between $v_{?i}$ and w_e. Finally, w_e is iterated into c. The copy of w_e shall be denoted v. During the iteration, an identity edge id_e^1 is added between w_e and v. The graph we obtain is $norm(\mathfrak{G}')$.

An example for this rule is given below. On the left side, we have $\mathfrak{G} \vdash_? \mathfrak{G}'$, on the right side, we have $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$.



, thus



- $C = ?i$ for an $i \in \mathbb{N}$, and $?i \in \mathrm{FV}(\mathfrak{G})$. This case is handled like the last case, except that the first step of adding $v_{?i}$ and $e_{?i}$ is omitted.

As we have finally shown that each rule with $\mathfrak{G} \vdash_?^r \mathfrak{G}'$ implies $norm(\mathfrak{G}) \vdash_? norm(\mathfrak{G}')$, we are done. $\square$

26

Peirce's Reduction Thesis

26.1 Introduction

> Now I call your attention to a remarkable theorem. Every polyad higher than a triad can be analyzed into triads, though not every triad can be analyzed into dyads.
>
> Peirce, Detached Ideas Continued, 1898[1]

In a footnote at the beginning of Sec. 3.2, we already shorty mentioned Peirce's three fundamental categories called *firstness*, *secondness*, and *thirdness*. For him, these three categories form a complete metaphysical scheme. Moreover, for Peirce, the basic constituting element of human reasoning are relations. The above given quotation can be understood as the application of Peirce's categories to relations ('polyads', as Peirce calls them in this place). So, to put in into contemporary terminology, each relation of arity higher then three can be constructed from ternary relations, but not every ternary relation can be constructed from dyadic relations only. But, of course, this claim needs an examination what Peirce means by 'being analyzed', i.e., it has to be investigated which operations on relations have to be taken into account.

A first approach to a mathematical elaboration of Peirce's reduction thesis has been provided by Herzberger in [Her81]. In this work, no Cartesian product of relations is allowed. This approach has been extended by Burch in [Bur91a], where he allows to use the Cartesian product of relations as last or second-last operation in the construction of relations. Inspired by [Bur91a], Hereth Correia and Pöschel in turn have extended Burch's result in [HCP06], as they allow to use the Cartesian product in arbitrary steps in the construction of relations.

In the following sections, we will use the result of Hereth Correia and Pöschel and adopt it for relation graphs. In all sections, we will only consider alphabets $\mathcal{A} := (\mathcal{R}, ar)$, i.e., we will not take constant or function names into account. In

Sec. 26.2, the operations on relations, as they have been used in [HCP06], are provided. Then, in Sec. 26.3, we will define corresponding operations on the graphs. It will turn out that (nearly) all relation graphs can be constructed with these operations. Then, in Sec. 26.4, we elaborate a version of Peirce's reduction thesis which suits the relation graphs of this treatise. In this section, we first show the first part of Peirce's thesis. The given result is more precise than Peirce's claim in the given quotation, as we will see that each relation can be constructed from dyadic relations and the teridentity. Similarly, Peirce's claim that '*not every triad can be analyzed into dyads*'will be made precise through the result of [HCP06], where it is shown that the teridentity Id_3^U on a given universe of discourse cannot be constructed out of dyadic relations. Finally, in Sec. 26.5, we will discuss more deeply the differences between the approaches of Herzberger, Burch, and Hereth Correia and Pöschel.

26.2 Peircean Algebraic Logic

The following two definitions, which are adopted from [HCP06], fix the operations of the Peircean Algebraic Logic. This definition of PAL is a strict extension of the version of PAL which Burch provided in [Bur91a].

Definition 26.1 (PAL-Operations). *Let U be an arbitrary set. On the set of all finitary relations on U, we define the following operations:*

PAL1 (Product): *If ϱ is an m-ary relation and S is an n-ary relation, then*

$$\varrho \times \sigma := \{(r_1, \ldots, r_m, s_1, \ldots, s_n) \mid (r_1, \ldots, r_m) \in \varrho \wedge (s_1, \ldots, s_n) \in \sigma\}$$

is the PRODUCT OF ϱ AND σ.

PAL2 (Join): *If ϱ is an n-ary relation and $1 \leq i < j \leq n$, then*

$$\delta^{i,j}(\varrho) := \{(r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n) \mid \\ \exists u \in U : (r_1, \ldots, r_{i-1}, u, r_{i+1}, \ldots, r_{j-1}, u, r_{j+1}, \ldots, r_n) \in \varrho\}$$

is the JOIN OF THE i-TH AND j-TH BLANK (OR PLACE) OF ϱ.

PAL3 (Complement): *If ϱ is an n-ary relation, then*

$$\neg\varrho := A^n \backslash \varrho = \{(u_1, \ldots, u_n) \in U^n \mid (u_1, \ldots, u_n) \notin \varrho\}$$

is the COMPLEMENT OF ϱ.

PAL4 (Permutation): *If ϱ is an n-ary relation and if α is a permutation on $\{1, \ldots, n\}$, then*

$$\pi_\alpha(\varrho) := \{(r_1, \ldots, r_n) \mid (r_{\alpha(1)}, \ldots, r_{\alpha(n)}) \in \varrho\}$$

is the α-PERMUTATION OF ϱ.

Definition 26.2 (Closure resp. PAL). *Let U be a set and $\Sigma \subseteq \mathsf{Rel}(U)$ be a set of finitary relations on U. Then let $\langle\Sigma\rangle_{PAL^{-\doteq_3}}$ be the smallest set which contains Σ and which is closed under the PAL-operations of Def. 26.1. Similarly, let $\langle\Sigma\rangle_{PAL}$ be the smallest set which contains $\Sigma \cup \{\doteq_3\}$ and which is closed under the PAL-operations.*

Let $\mathcal{M} := (U, I)$ be a model over an alphabet $\mathcal{A} := (\mathcal{R}, ar)$. Then we set

$$\langle\mathcal{M}\rangle_{PAL^{-\doteq_3}} := \langle\{I(\varrho) \mid \varrho \in \mathcal{R}\}\rangle_{\mathsf{PAL}\setminus\{\doteq_3\}}$$
$$\langle\mathcal{M}\rangle_{PAL} := \langle\{I(\varrho) \mid \varrho \in \mathcal{R}\}\rangle_{PAL}$$

Recall that for a given set U, we defined $Id_3^U := \{(u,u,u) \mid u \in U\}$ (see page 183). In [HCP06], Hereth Correia and Pöschel have proven the following algebraic version of the second part of Peirce's reduction thesis. This is the main result we will use in this chapter.

Theorem 26.3 (An Algebraic Version of Peirce's Reduction Thesis (Hereth Correia and Pöschel 2006)). *Let $|U| \geq 2$ and let Σ be the set of all 1- and 2-ary relations over U. Then we have*

$$Id_3^U \notin \langle\Sigma\rangle_{PAL^{-\doteq_3}} \qquad ,thus \qquad \langle\Sigma\rangle_{PAL^{-\doteq_3}} \subsetneq \langle\Sigma\rangle_{PAL}$$

26.3 Graphs for Peircean Algebraic Logic

In the last section, the operations of PAL, which act on relations, had been defined, and an algebraized version of the Peircean Reduction Thesis had been provided. In this section, we transfer this result to relation graphs.

We start our scrutiny by transferring the (algebraic) PAL-operations of Def. 26.1 to (syntactical) operations on RGIs.

Definition 26.4 (PAL Operations on Graphs). *On the set of all RGIs over an alphabet $\mathcal{A}$, we define the following atomar graphs and operations:*

1. ***Atomar graphs:*** *Let R be an n-ary relation name. The graph*

$$\begin{aligned}\mathfrak{G}_R := (&\{v_1, \ldots, v_n\}, \{e_1, \ldots, e_n, e_R\},\\ &\{(e_1,(v_1)), \ldots, (e_n,(v_n)), (e_R,(v_1,\ldots,v_n))\},\\ &\top, \emptyset, \emptyset, \{(e_1, ?1), \ldots, (e_n, ?n), (e_R, R)\})\end{aligned}$$

is the atomar RGI corresponding to R. Moreover, the graph

$$\mathfrak{G}_{?1} := (\{v\}, \{e\,\{(e,(v))\}, \top, \emptyset, \emptyset, \{(e, ?1)\})$$

is an atomar graph.

2. ***Product:*** *Let two RGIs* $\mathfrak{G}_1 := (V_1, E_1, \nu_1, \top_1, Cut_1, area_1, \kappa_1)$, $\mathfrak{G}_2 := (V_2, E_2, \nu_2, \top_2, Cut_2, area_2, \kappa_2)$ *with* $m = ar(\mathfrak{G}_1)$ *and* $n = ar(\mathfrak{G}_2)$ *be given. Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be the juxtaposition of* $\mathfrak{G}_1$ *and* $\mathfrak{G}_1$, *where the labeling mapping* κ *is changes as follows: For* $e \in E_1$, *we set* $\kappa(e) := \kappa_1(e)$, *and for* $e \in E_2$, *we set*

$$\kappa(e) := \begin{cases} \kappa_2(e) & \text{if } e \notin (E_2)^? \\ ?(i+m) & \text{if } \kappa_2(e) = ?i \end{cases}$$

Then $\mathfrak{G}$ *is called the* PRODUCT OF $\mathfrak{G}_1$ AND $\mathfrak{G}_2$ *and denoted by* $\mathfrak{G}_1 \times \mathfrak{G}_2$.

3. ***Join:*** *Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be an* n*-ary RGI, and let* $1 \leq i < j \leq n$. *Let* v_i, v_j *be the edges with* $\kappa(e_i) = ?i$ *and* $\kappa(e_j) = ?j$, *resp., and let* v_i *and* v_j *be the vertices incident with* e_i *and* e_j, *resp. Let* v *be a fresh vertex. Let* $\mathfrak{G}'$ *be the following graph:*
   - $V' := V \backslash \{v_i, v_j\} \dot{\cup} \{v\}$,
   - $E' := E \backslash \{e_i, e_j\}$,
   - *We define* ν' *as follows: For* $\nu(e) = (w_1, \ldots, w_m)$, *let* $\nu'(e) = (w'_1, \ldots, w'_m)$ *with* $w'_k := \begin{cases} w_k & \text{for } w_k \notin \{v_i, v_j\} \\ v & \text{for } w_k \in \{v_i, v_j\} \end{cases}$.
   - $\top' := \top$
   - $Cut' := Cut$
   - $area'(c) := area(c)$ *for each* $c \neq \top'$, *and* $area'(\top') := area(\top') \backslash \{v_i, v_j, e_i, e_j\} \dot{\cup} \{v\}$,
   - $\kappa'(e) := \kappa(e)$ *for each* $e \notin E^?$, *and*

$$\kappa'(e) := \begin{cases} ?k & \text{for } \kappa(e) = ?k \text{ and } k < i \\ ?(k-1) & \text{for } \kappa(e) = ?k \text{ and } i < k < j \\ ?(k-2) & \text{for } \kappa(e) = ?k \text{ and } j < k \end{cases}$$

We say that we obtain $\mathfrak{G}'$ *from* $\mathfrak{G}$ *by* JOINING (THE BLANKS) i AND j. *Sometimes we will say 'by joining* $?i$ *and* $?j$*' as well.* $\mathfrak{G}'$ *denoted by* $\delta^{i,j}(\mathfrak{G})$.

4. ***Complement:*** *Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be an* n*-ary RGI. Let* c *be a fresh cut (i.e.,* $c \notin E \cup V \cup Cut \cup \{\top\}$*). Let* $V^?$ *be the set of all vertices which are incident with an* $e \in E^?$. *Now let* $\neg\mathfrak{G}$ *be the RGI* $(V, E, \nu, \top, Cut', area', \kappa)$ *with* $Cut' := Cut \cup \{c\}$, *where* $area'$ *is defined as follows:*

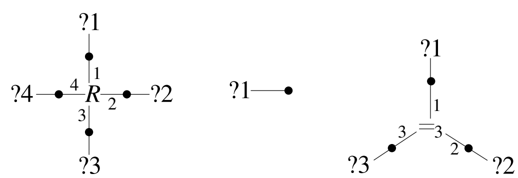
$$area'(d) := \begin{cases} E^? \cup V^? & \text{for } d = \top \\ area(c) \backslash (E^? \cup V^?) & \text{for } d = c \\ area(c) & \text{else} \end{cases}$$

This graph is an n*-ary RGI. Then* $\mathfrak{G}'$ *is called* THE NEGATION OF $\mathfrak{G}$ *and denoted by* $\neg\mathfrak{G}$.

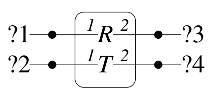
5. ***Permutation:*** *Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be an* n*-ary RGI, let* α *be a permutation of* $\{1, \dots, n\}$*. Let* $\mathfrak{G}' := (V, E, \nu, \top, Cut, area, \kappa')$ *be the RGI with*
$$\kappa'(e) := \begin{cases} \kappa(e) & \textit{for } e \notin E^? \\ ?\alpha(i) & \textit{for } e \in E^? \textit{ with } \kappa(e) = ?i \end{cases}$$
This graph is an n*-ary RGI. Then* $\mathfrak{G}'$ *is called* THE α-PERMUTATION OF $\mathfrak{G}$ *and denoted by* $\pi_\alpha(\mathfrak{G})$*.*

Before we proceed, first some simple examples for this definition are provided.

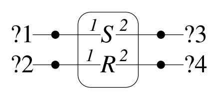
1. Atomar graphs: Below, the graph $\mathfrak{G}_R$ for a relation name R with $ar(R) = 4$, as well $\mathfrak{G}_{?1}$, is depicted. Moreover, as the teridentity will play a crucial role in the definition of PAL-graphs, the graph $\mathfrak{G}_{\dot{=}_3}$ is depicted as well.



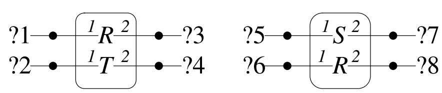
2. Product: Example: From



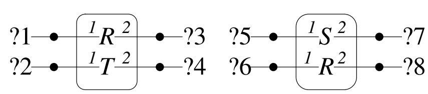
and



we obtain

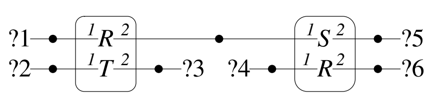


3. Join: For example, with joining 3 and 5, from

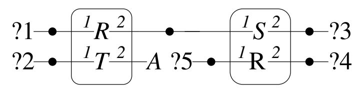


we obtain

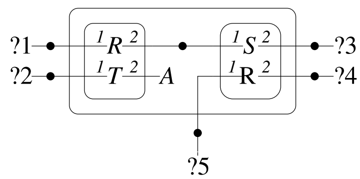
.



4. Complement: For example, from



we obtain



.

Def. 26.4 is the syntactical counterpart of Def. 26.1 for RGIs. After Def. 26.1, we defined in Def. 26.2 the closure of sets of relations which respect to the PAL-operations. As the teridentity plays the central role in the formal elaboration of Peirce's reduction thesis, we defined two closures where in one case, besides the given relations, the teridentity is included in the closure process.

The next definition is the syntactical counterpart of of Def. 26.2, i.e., we define classes of graphs which are closed under the syntactical operations of Def. 26.4. Again with respect to teridentity, we will define PAL$^{-\dot{=}_3}$-graph instances and PAL-graph instances.

Recall that in Chpt. 16, where the method of separating ligatures at branching points has been introduced, we augmented a given alphabet $\mathcal{A}$ by new names $\dot{=}_k$ for the k-ary identity. The resulting alphabet has been denoted $\mathcal{A}^{\dot{=}}$. For the purpose we have now in mind, we will augment $\mathcal{A}$ only with the name $\dot{=}_3$ for the teridentity. The resulting alphabet is denoted $\mathcal{A}^{\dot{=}_3}$. As we then have $\dot{=}_3 \in \mathcal{R}$, the atomar graph $\mathfrak{G}_{\dot{=}_3}$ can be used in the construction of the graph instances. That is, similar as the teridentity served to distinguish between the closures $\langle \Sigma \rangle_{\mathrm{PAL}^{-\dot{=}_3}}$ and $\langle \Sigma \rangle_{\mathrm{PAL}}$ in Def. 26.2, now $\mathfrak{G}_{\dot{=}_3}$ will serve to distinguish between PAL$^{-\dot{=}_3}$-graph instances and PAL-graph instances.

Definition 26.5 (PAL Graph Instances). *Let $\mathcal{A} := (\mathcal{R}, ar)$ be a given alphabet. We set:*

- *A* PAL$^{-\dot{=}_3}$-GRAPH INSTANCE OVER $\mathcal{A}$ *is defined to be an RGI which can be obtained from finitely many atomar graphs $\mathfrak{G}_R$ with $R \in \mathcal{R}$ by finitely many applications of the operations product, join, complement, and permutation, and*
- *a* PAL-GRAPH INSTANCE OVER $\mathcal{A}$ *is defined to be a PAL$^{-\dot{=}_3}$-graph instance over $\mathcal{A}^{\dot{=}_3}$.*

As usual, PAL$^{-\dot{=}_3}$-graph instances and PAL-graph instances are abbreviated by PAL-GIs and PAL-GIs.

Recall that we defined branching points as vertices which are attached to more than two hooks (see Def. 12.9). Up to some minor syntactical restrictions, the $\mathrm{PAL}^{-\dot{=}_3}$-GIs over $\mathcal{A}$ are the RGIs over $\mathcal{A}$ without branching points. This is the subject of the next lemma.

Lemma 26.6 (The PAL-graphs are the normed RGIs without branching points). *A graph $\mathfrak{G}$ is a $\mathrm{PAL}^{-\dot{=}_3}$-GI over $\mathcal{A}$ if and only if it is an normed RGI over $\mathcal{A}$ such that*

1. *$\mathfrak{G}$ has no branching points,*
2. *$\mathfrak{G}$ has no isolated vertices, and*
3. *$\mathfrak{G}$ has no vertices which are incident with two different edges $e, f \in E^?$.*

Proof: Let us call the normed RGIs which satisfy the three conditions of the lemma PAL*-GRAPH INSTANCE INSTANCES. That is, we have to show that each RGI $\mathfrak{G}$ is a $\mathrm{PAL}^{-\dot{=}_3}$-GI if and only if it is and PAL*-graph instance.

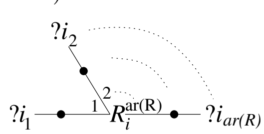
It can easily shown by induction over the construction of $\mathrm{PAL}^{-\dot{=}_3}$-GIs that each $\mathrm{PAL}^{-\dot{=}_3}$-GI is a PAL*-graph instance. So it remains to prove the opposite direction.

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be a PAL*-graph instance. Then each vertex $v \in V$ satisfies exactly one of the following conditions:

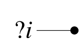
- v is attached to two different hooks (e, i) and (f, j) ($e = f$ is possible) with $e, f \notin E^?$. The class of these vertices shall be denoted $T_1(V)$.
- v is attached to exactly one hook (e, i) with $e \notin E^?$. The class of these vertices shall be denoted $T_2(V)$.
- v is attached to two different hooks (e, i) and (f, j) with $e \in E^?$ and $f \notin E^?$. The class of these vertices shall be denoted $T_3(V)$.
- v is attached to exactly one hook (e, i) with $e \in E^?$. The class of these vertices shall be denoted $T_4(V)$.

Thus we have $V = T_1(V) \dot{\cup} T_2(V) \dot{\cup} T_3(V) \dot{\cup} T_4(V)$. Now we assign to $\mathfrak{G}$ its PAL-COMPLEXITY $palc(\mathfrak{G}) := 3 \cdot |T_1(V)| + |T_2(V)| + |Cut|$. The lemma is now proven over the PAL-complexity of PAL*-graph instances, i.e., by induction over $palc(\mathfrak{G})$. In the proof, we set $n := ar(\mathfrak{G})$.

We first consider an PAL*-graph instance $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ with $palc(\mathfrak{G}) = 0$. Then $\mathfrak{G}$ has no cuts, and each vertex is either attached to two different hooks (e, i) and (f, j) with $e \in E^?$ and $f \notin E^?$, or to exactly one hook (e, i) with $einE^?$. That is, $\mathfrak{G}$ is the juxtaposition of graphs of the following form (where $R \in \mathcal{R}$):

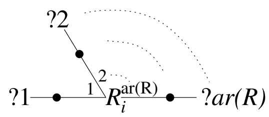


and

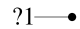


.

Then $\mathfrak{G}$ can be obtained by the successive join of the corresponding atomar PAL*-graph instances



and



,

followed by an appropriate permutation.

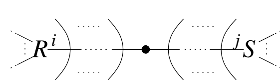
Now let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an PAL*-graph instance with $palc(\mathfrak{G}) > 0$.

Assume first we have a vertex $v \in T_1(V)$ with $ctx(v) = \top$. Let v be attached to the hooks (e_1, i_1) and (e_2, i_2). The graph $\mathfrak{G}$ can be obtained from another graph $\mathfrak{G}'$ by a join operation. The graph $\mathfrak{G}'$ is defined as follows:

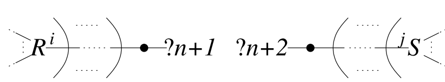
Let $v_{?(n+1)}$,$v_{?(n+2)}$ be fresh vertices and $e_{?(n+1)}$,$e_{?(n+2)}$ be fresh edges. Let $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be the following graph:

- $V' := V \backslash \{v\} \dot\cup \{v_{?(n+1)}, v_{?(n+2)}\}$,
- $E' := E \backslash \dot\cup \{e_{?(n+1)}, e_{?(n+2)}\}$,
- We define ν' as follows: For $\nu(f) = (w_1, \ldots, w_m)$, let $\nu'(f) = (w'_1, \ldots, w'_m)$ with $w'_k := \begin{cases} v_{?(n+1)} & \text{for } (f,k) = (e_1, i_1) \\ v_{?(n+2)} & \text{for } (f,k) = (e_2, i_2) \\ w_k & \text{else} \end{cases}$.
- $\top' := \top$
- $Cut' := Cut$
- $area'(c) := area(c)$ for each $c \neq \top'$, and $area'(\top') := area(\top') \backslash \{v\} \dot\cup \{v_{?(n+1)}, v_{?(n+2)}, e_{?(n+1)}, e_{?(n+2)}\}$,
- $\kappa' := \kappa \dot\cup \{(e_{?(n+1)}, ?n+1), (e_{?(n+2)}, ?n+2)\}$

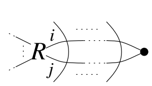
Informally depicted, with $R = \kappa(e_1)$ and $S = \kappa(e_2)$,



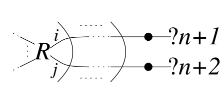
is replaced by



resp. for $e_1 = e_2$ and $\kappa(e) = R$,



is replaced by



Then we have $\mathfrak{G} = \delta^{n+1,n+2}(\mathfrak{G}')$ and $palc(\mathfrak{G}') = palc(\mathfrak{G}) - 1$, i.e. for this case, the induction is done.

Next, we consider the case that we have a vertex $v \in T_2(V)$ with $ctx(v) = \top$. This case is handled similarly to the last case. Let v be attached to the hook (e, i). We define another graph $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ as follows:

- $V' := V$
- $E' := E \,\dot{\cup}\, \{e_{?(n+1)}\}$,
- $\nu' := \nu \,\dot{\cup}\, \{(v, e_{?(n+1)})\}$
- $\top' := \top$
- $Cut' := Cut$
- $area'(c) := area(c)$ for each $c \neq \top$, and $area'(\top) := area(\top) \,\dot{\cup}\, \{e_{?(n+1)}\}$
- $\kappa' := \kappa \,\dot{\cup}\, \{(e_{?(n+1)}, ?(n+1))\}$

Then we have $\mathfrak{G} = \delta^{n+1,n+2}(\mathfrak{G}' \times \mathfrak{G}_{?1})$ and $palc(\mathfrak{G}') = palc(\mathfrak{G}) - 1$. Thus for this case, the induction is done as well.

So we can now assume that we have $T_1(V) \cap area(\top) = \emptyset = T_2(V) \cap area(\top)$. As we have $palc(\mathfrak{G}) > 0$, we have $Cut \neq \emptyset$. Let $c \in area(\top) \cap Cut$ be a fixed cut. Let us consider an edge e which is enclosed by c, i.e., $e < c$. Note that $e \notin E^?$. If e is incident with a vertex v which is not enclosed by c, this vertex must be placed on $\top$ and satisfies $v \in T_1(V)$. So if we define

$$V_c^+ := \{v \in V \cap area(\top) \mid v \text{ is incident with an edge } e \text{ with } e < c\}$$
$$E_c^+ := \{e \in E \cap area(\top) \mid e \text{ is incident with e vertex } v \in V_c^+\}$$

the sets $V_c := (\leq[c] \cap V) \cup V_c^+$, $E_c := (\leq[c] \cap E) \cup E_c^+$, and $Cut_c := Cut$ give rise to a closed subgraph $\mathfrak{G}_c := (V_c, E_c, \nu_c, \top_c, Cut_c, area_c, \kappa_c)$ of $\mathfrak{G}$, which is placed on the sheet of assertion. As $\mathfrak{G}_c$ is closed, the sets $V_{-c} := V \backslash V_c$, $E_{-c} := E \backslash E_c$, and $Cut_{-c} := Cut \backslash Cut_c$ give rise to a closed subgraph $\mathfrak{G}_{-c}$ of $\mathfrak{G}$ as well (this graph may be empty). Now $\mathfrak{G}$ is the juxtaposition of $\mathfrak{G}_c$ and $\mathfrak{G}_{-c}$. By choosing appropriate permutations α, β, γ, we get

$$\mathfrak{G} = \pi_\alpha(\pi_\beta(\mathfrak{G}_c) \times \pi_\gamma(\mathfrak{G}_{-c}))$$

Moreover, we have $palc(\mathfrak{G}_{-c}) < palc(\mathfrak{G})$.

Finally, let $\pi_\beta(\mathfrak{G}_c) := (V_c', E_c', \nu_c', \top_c', Cut_c', area_c', \kappa_c')$. Now we set $\mathfrak{G}' := (V_c', E_c', \nu_c', \top', Cut', area', \kappa_c')$ with $Cut' := Cut_c' \backslash \{c\}$, $area'(d) := area_c'(d)$ for each $d \neq \top$, and $area'(\top) = area_c'(\top) \cup area_c'(c)$ (roughly speaking, $\mathfrak{G}'$ is obtained from $\pi_\beta(\mathfrak{G}_c)$ by removing c). Then by definition of $\mathfrak{G}_c$, we have $\beta(\mathfrak{G}_c) = \neg\mathfrak{G}'$, and we have $palc(\mathfrak{G}') = palc(\pi_\beta(\mathfrak{G}_c)) - 1 \leq palc(\mathfrak{G})$.

So for the last case where $T_1(V) \cap area(\top) = \emptyset = T_2(V) \cap area(\top)$, $\mathfrak{G}$ can be composed of PAL*-graph instances with a lower PAL-complexity, too. This finishes the proof. □

Please note that the lemma can be applied to PAL-ohne instances over $\mathcal{A}^{\doteq_3}$ as well, i.e, PAL$^{-\doteq_3}$-GIs are the RGIs over $\mathcal{A}$ which have no branching points, as well as no isolated vertices, and no vertices which are incident with two different edges.

The second and third condition for PAL$^{-\doteq_3}$-GIs are of minor technical nature. More importantly for is is that PAL$^{-\doteq_3}$-GIs have no branching points. But in Chpt. 16, where the method of separating ligatures at branching points has been discussed, it has been, roughly speaking, shown that instead of branching points, it is (semantically) sufficient to have the relation name $\doteq_3$ in our alphabet. That is, we obtain the following lemma:

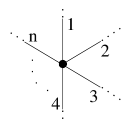
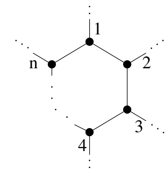
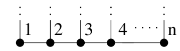
Lemma 26.7 (PAL$^{-\doteq_3}$- and PAL-GIs are equivalent to RGIs without or with branching points). *Let $\mathcal{A}$ be a given alphabet. Then we have:*

1. *For each RGI $\mathfrak{G}$ without branching points over $\mathcal{A}$, there exists a corresponding PAL$^{-\doteq_3}$-GI $\mathfrak{G}'$ which expresses the same relation, i.e., we have $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = \mathfrak{R}_{\mathcal{M},\mathfrak{G}'}$ for all models $\mathcal{M}$.*
2. *For each RGI $\mathfrak{G}$ over $\mathcal{A}$, there exists a corresponding PAL-GI $\mathfrak{G}'$ which expresses the same relation.*

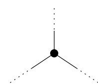
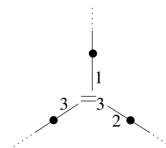
Proof:

1. Each RGI without branching points $\mathfrak{G}$ can be easily transformed into a syntactically equivalent, thus semantically equivalent, PAL$^{-\doteq_3}$-GI $\mathfrak{G}'$ as follows:
   a) With the iteration-rule, each isolated vertex $\bullet$ of $\mathfrak{G}$ is replaced by $\bullet\!- = -\!\bullet$ (compare this to Def. 19.2).
   b) Now let $v \in V$ be a vertex which is incident with two different edges $e, f \in E^?$. With the rule 'adding a vertex to a ligature', we add a vertex v' and a new identity link between v and v', and on $(e, 1)$, v is replaced by v'.

   The resulting graph has still no branching point, it is equivalent to $\mathfrak{G}$ and satisfies now the second and third condition of Lem. 26.6, too. That is, it is a PAL$^{-\doteq_3}$-graph instance.
2. According to 1), we first assume that $\mathfrak{G}$ satisfies the second and third condition of Lem. 26.6. We have shown in Lemma 16.3 that a branching point with more than three branches can be converted into a 'wheel' or a 'fork' (see page 173), where only branching points with three branches occur. Then, for each vertex v which is a branching point with three branches, we separate a ligature at v (see page 184 ff). That is, each device


is first is replaced by

or


and after that, each device


is replaced by


As both conversions respect the second and third condition of Lem. 26.6, the resulting graph is a PAL-graph $\mathfrak{G}'$ which is syntactically, thus semantically, equivalent to $\mathfrak{G}$. $\Box$

So the class of all RGIs without branching points over $\mathcal{A}$ has the same expressiveness as the class of all PAL$^{-\dot{=}_3}$-GIs over $\mathcal{A}^{=}$, and the class of all RGIs, where branching points are allowed, the same expressiveness as the class of all PAL-GIs over $\mathcal{A}$.

The operations of Def. 26.1 are *semantical* operations on relations, the operations of Def. 26.4 are *syntactical* operations on graphs. Thus before we use Thm. 26.3 in the next section to prove the graph-version of Peirce's reduction thesis, we still have to show that the syntactical operations of Def. 26.4 correspond semantically to the operations of Def. 26.1.

Lemma 26.8 (Inductive Semantics of PAL$^{-\dot{=}_3}$-GIs). *Let $\mathfrak{G}$, $\mathfrak{G}_1$, $\mathfrak{G}_2$ be PAL$^{-\dot{=}_3}$-GIs and let $\mathcal{M} := (U, I)$ be a model. Then we have:*

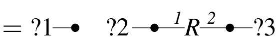
1. ***Atomar graphs:*** $\mathfrak{R}_{\mathcal{M},\mathfrak{G}_R} = I(R)$ *and* $\mathfrak{R}_{\mathcal{M},\mathfrak{G}_{?1}} = U$*. Particularly we have* $\mathfrak{R}_{\mathcal{M},\mathfrak{G}_{\dot{=}_3}} = Id_3^U$.
2. ***Product:*** $\mathfrak{R}_{\mathcal{M},\mathfrak{G}_1 \times \mathfrak{G}_2} = \mathfrak{R}_{\mathcal{M},\mathfrak{G}_1} \times \mathfrak{R}_{\mathcal{M},\mathfrak{G}_2}$
3. ***Join:*** $\mathfrak{R}_{\mathcal{M},\delta^{i,j}(\mathfrak{G})} = \delta^{i,j}(\mathfrak{R}_{\mathcal{M},\mathfrak{G}})$
4. ***Complement:*** $\mathfrak{R}_{\mathcal{M},\neg\mathfrak{G}} = \neg\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$
5. ***Permutation:*** $\mathfrak{R}_{\mathcal{M},\pi_\alpha(\mathfrak{G})} = \pi_\alpha(\mathfrak{R}_{\mathcal{M},\mathfrak{G}})$

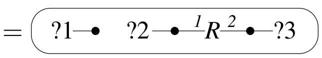
Proof: The proof is carried out by induction over the construction of PAL$^{-\dot{=}_3}$-GIs. The lemma is easily seen for atomar graphs, and the operations product, join, and permutation. It remains to show $\mathfrak{R}_{\mathcal{M},\neg\mathfrak{G}} = \neg\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$.

Let $\mathfrak{G}^c$ be the graph obtained from $\mathfrak{G}$ by completely enclosing it with the fresh cut c. That is, $\mathfrak{G}^c := (V, E, \nu, c, Cut \,\dot{\cup}\, \{\top\}, area^c, \kappa$ with $area^c(c) := \{\top\}$ and $area^c(d) := area(d)$ for each $d \in Cut \cup \{\top\}$ (note that for $\mathrm{FV}(\mathfrak{G}) \neq \emptyset$, the RGI

$\mathfrak{G}^c$ is not normed, particularly not a PAL$^{-\doteq_3}$-GI). We have $\mathfrak{R}_{\mathcal{M},\mathfrak{G}^c} = \neg\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$. Now we show that $\mathfrak{G}^c$ and $\neg\mathfrak{G}$ are syntactically, thus semantically, equivalent, by providing a proof for $\mathfrak{G}^c \vdash_? \neg\mathfrak{G}$ which uses only rules which can be carried out in both directions. The transformation is similar (but not identical) to the normalization of an RGIs (see Def. 25.9).

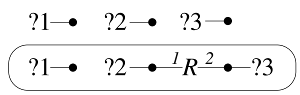
1. Our example is the graph

   $\mathfrak{G} :=$ 

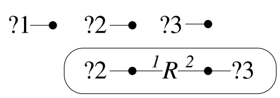
   , thus $\mathfrak{G}^c =$ 

   For each $?i \in \mathrm{FV}(\mathfrak{G}^c)$, the edge of $\mathfrak{G}^c$ labeled with $?i$ is denoted $e_{?i}$, and the vertex incident with $e_{?i}$ is denoted $v_{?i}$.

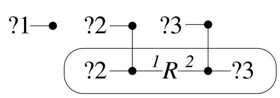
   For the following steps, please note that as $\mathfrak{G}$ is a PAL$^{-\doteq_3}$-GI, each vertex $v_{?i}$ of $V^?$ belongs either to $T_3(V^c)$ or to $T_4(V^c)$ (we use the notation of the proof of Lem. 26.6).
2. For each $?i \in \mathrm{FV}(\mathfrak{G}^c)$, we add a new vertex $w_{?i}$ and a new edge $f_{?i}$ with $\nu(f_{?i}) = (v_{?i})$ and $\kappa(f_{?i}) = ?i$ to the sheet of assertion. Within the calculus, this is done with the existence of constants rule.

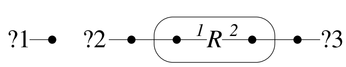
3. For each vertex $v_{?i}$ with $v_{?i} \in T_4(V^c)$, we erase the subgraph consisting of $v_{?i}$ and $e_{?i}$ from $\top$ (recall that $\top$ is a *cut* in $\mathfrak{G}^c$).

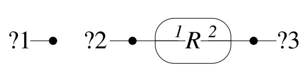
4. For each vertex $v_{?i}$ with $v_{?i} \in T_3(V^c)$, we add an new identity edge $id_{?i}$ between $w_{?i}$ and $v_{?i}$ to $\top$.

5. Similarly to the fifth step in the normalization of a graph, for each $?i$ with $v_{?i} \in T_3(V^c)$, the edge $e_{?i}$ is deiterated from $\top$.

6. Finally, for each $?i$ with $v_{?i} \in T_3(V^c)$, the identity edge $id_{?i}$ and the vertex $v_{?i}$ is erased with the rule 'removing a vertex from a ligature'.

The last graph is (isomorphic to) $\neg\mathfrak{G}$, thus we are done. $\square$

Please note that the last step of the proof relies of $\mathfrak{G}$ being a PAL$^{-\dot{=}_3}$-GI. To see this, consider the following graphs:

Let $\mathfrak{G} := ?1 — \bullet — ?2$, thus $\neg\mathfrak{G} = ?1 — \bullet — ?2 \ \boxed{}$ and $\mathfrak{G}^c = \boxed{?1 — \bullet — ?2}$.

Then we have $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = \{(u,v) \in U^2 \mid u = v\}$ $(= I(\dot{=}))$, $\mathfrak{R}_{\mathcal{M},\neg\mathfrak{G}} = \emptyset$, and $\mathfrak{R}_{\mathcal{M},\mathfrak{G}^c} = \{(u,v) \in U^2 \mid u \neq v\}$. So, even for the normed RGI $\mathfrak{G}$, we have $\mathfrak{R}_{\mathcal{M},\mathfrak{G}^c} \neq \mathfrak{R}_{\mathcal{M},\neg\mathfrak{G}}$.

Comment: On page 64, it was argued why we defined graphs not inductively, but in one step. For PAL$^{-\dot{=}_3}$-GIs, their inductive definition is canonical. If we had so far no semantics for these graphs, the five propositions of Lem. 26.8 could serve as an inductive definition of their semantics. But PAL$^{-\dot{=}_3}$-GIs do not have an unique derivational history. For example, the constructions

$$\delta^{2,3}(\delta^{4,5}((\mathfrak{G}_R \times \mathfrak{G}_S) \times \mathfrak{G}_T)) \qquad \text{and} \qquad \pi_{(1,2)}(\delta^{1,4}(\delta^{2,3}(\mathfrak{G}_S \times \mathfrak{G}_T) \ \times \ \mathfrak{G}_R))$$

both yield the same graph

$$\mathfrak{G} \ := \ ?1 — \bullet — {}^1R^2 — \bullet — {}^1S^2 — \bullet — {}^1T^2 — \bullet — ?2 \quad .$$

Thus for a model (U,I), an inductively semantics would assign (at least) the two relations $\delta^{2,3}(\delta^{4,5}((I(R) \times I(S)) \times I(T)))$ and $\pi_{(1,2)}(\delta^{1,4}(\delta^{2,3}(I(S) \times I(T)) \ \times \ I(R)))$ to $\mathfrak{G}$. So the semantics is not well-defined unless it is proven that these terms yield indeed the same relation.

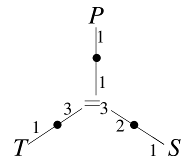
Lem. 26.8 can now be understood to provide such a proof, as it shows that an inductive definition of the semantics for PAL$^{-\dot{=}_3}$-graphs yields simply the same semantics as defined in Def. 25.5. Particularly, it is well-defined.

26.4 Peirce's Reduction Thesis for Relation Graphs

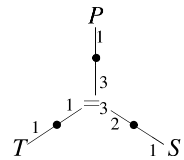
After providing the algebraic version of Peirce's reduction thesis in Sec. 26.2, and the counterparts for RGIs of the algebraic operations in Sec. 26.3, we can now elaborate the full account the graph-version of Peirce's reduction thesis.

Before we do so, we will shortly discuss branching points in PAL-GIs. Strictly speaking, in the understanding that branching points are vertices which are attached to more than three hooks, PAL-GIs do not have branching points. But of course, as we have it already used in the proof of Lem. 26.7, edges which are labeled with the name $\dot{=}_3$ for the teridentity can be understood as branching points. When we first introduced the names $\dot{=}_k$ on page 184 ff, this was done in the context to separate ligatures into single object ligatures. For this purpose, it was reasonable that in the graphical representation of formal EGs, we really used the signs $\dot{=}_k$ in their graphical representations (see for example the formal EGs in Figs. 16.5 and 16.3).

Now, formal PAL$^{-\dot{=}_3}$-graphs and formal PAL-graphs are defined in the usual manner, i.e. by factorizing the class of all PAL$^{-\dot{=}_3}$-GIs and formal PAL-GIs by means of the transformation rules for ligatures. Moreover, we agree that changing the order of the hooks of an edge labeled with $\doteq_3$ is a further allowed transformation rule for ligatures. For example, the PAL-GIs

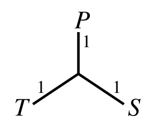


and



are in the class of the same formal PAL-graph.

For the graphical representation of PAL-GIs, we now agree that an edge labeled with $\doteq_3$ is represented as a branch of three heavily drawn lines, without using the sign $\doteq_3$. I.e., we agree that the formal formal PAL-graph generated by either one of the just given PAL-GIs is depicted as follows:



In the following, we first show that each relation of arity higher than two can be expressed with PAL-GIs by means of relations of arity one and two. To make this more precise: If $\mathcal{M} := (U, I)$ is a model such that

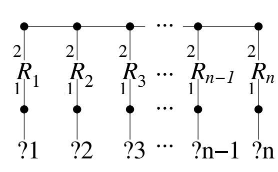
- for each relation ϱ of arity one or two, there exists a corresponding relation name R with $I(R) = \varrho$, and
- we have a name $\doteq_3$ for the teridentity (i.e., $(\doteq_3) = Id_3^U$),

then for each relation ϱ there exists a PAL-graph $\mathfrak{G}$, where –besides $\doteq_3$– only relation names of arity one or two occur, and which satisfies $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = \varrho$.

We have to distinguish two cases. Let us first consider the case where U is infinite. Let $\varrho \subseteq U^n$ be a relation with $n \geq 3$. Using an index set I for enumerating the pairs in ϱ, we have:

$$\varrho = \{(u_{1,i}, u_{2,i}, \ldots, u_{n,i}) \mid i \in I\}$$

As U is infinite, we have $|\varrho| \leq |U^n| = |U|$. So w.l.o.g. we can assume $I \subseteq U$. Now for $k = 1, \ldots, n$, let $\varrho_k := \{(u_{k,i}, i) \mid i \in I\}$, and let R_k be a relation name with $I(R_k) = \varrho_k$. Then the RGI



evaluates in $\mathcal{M}$ to ϱ. The given graph is not a PAL-GI, as it has $n - 2$ branching points. By separating the ligatures at these vertices, this graph can

can be transformed into a semantically equivalent PAL-GI. This PAL-GI, and moreover the corresponding formal PAL graph, is provided in Fig. 26.1.

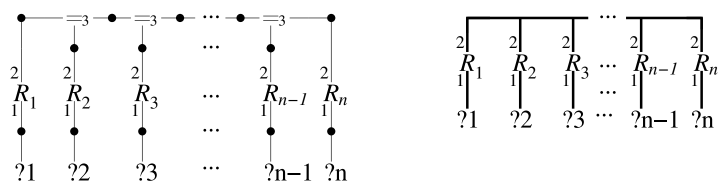


Fig. 26.1. A PAL-GI and the corresponding PAL-graph which evaluate to ϱ

Comment: For the algebraic version of PAL, as it was developed in Sec. 26.2, there exists a corresponding term which evaluates to ϱ. If we use $\circ$ as the usual notation for the composition of the unary functions $\delta^{i,j}$, $\bigodot$ for an indexed representation of this composition, and $\bigotimes$ for an indexed representation of the product of relations, the PAL-expression is as follows:

$$\varrho = \left(\left[\bigodot_{i=0}^{n-3} \delta^{2+5i,4+5i} \circ \delta^{4+5i,6+5i} \right] \circ \delta^{2+5(n-2),4+5(n-2)} \right) \left(\varrho_1 \times \bigotimes_{i=0}^{n-3} \varrho_{i+1} \times \doteq_3 \right)$$

For example, for $ar(\varrho) = 4$; we have

$$\varrho = \delta^{2,4}(\delta^{4,6}(\delta^{7,9}(\delta^{9,11}(\delta^{12,14}(\delta^{14,16}(\delta^{17,19}(R_1 \times R_2 \times \doteq_3 \times R_3 \times \doteq_3 R_4)))))))$$

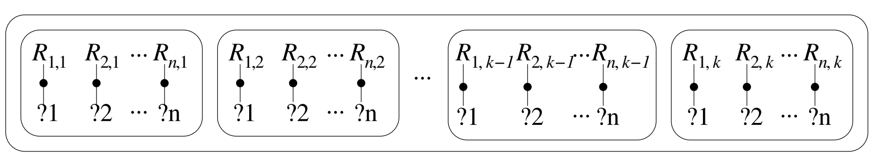
This shows that although they have the same expressiveness, the PAL-graphs are much easier to read than the corresponding terms of the algebraic version of PAL.

If U is finite, we might have $|\varrho| > |U|$. Thus the 'technical trick' to use some elements of U to refer to the tuples of ϱ does only work if the universe U is infinite. So for us, a different approach is needed.

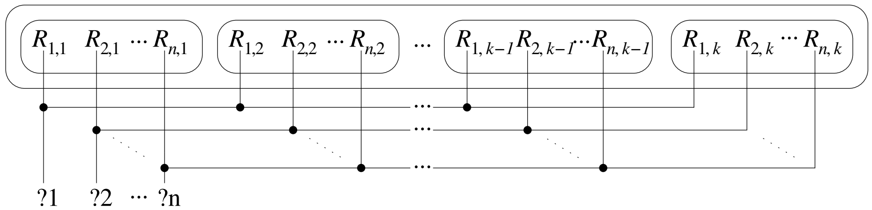
Now let U be finite and ϱ be an n-ary relation with k elements, i.e.,

$$\varrho = \{(u_{1,i}, \ldots, u_{n,i}) \mid i = 1, \ldots, k\}$$

In contrast to Herzberger or Burch, we can express finite unions of relations within the system of PAL-graphs. This will be used to find a PAL-graph $\mathfrak{G}$ with $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = \varrho$. First, for $j = 1, \ldots, n$ and $i := 1, \ldots, k$, we set $\varrho_{j,i} := \{(u_{j,i})\}$. For $i = 1, \ldots, k$, the product $\varrho_{1,i} \times \varrho_{2,i} \times \ldots \times \varrho_{n,i}$ yields the n-ary relation which contains exactly the ith tuple of ϱ as element. So ϱ is the union of these relations, and we see that the semi RGI



evaluates in $\mathcal{M}$ to ϱ. This graph is not an RGI, but we can consider its (semantically equivalent) normalization instead. This graph is, after we reduced the set of vertices with the rule 'removing a vertex from a ligature':



Similar to the last case, in Fig. 26.2 a semantically equivalent PAL-GI and the corresponding formal PAL-graph are depicted.

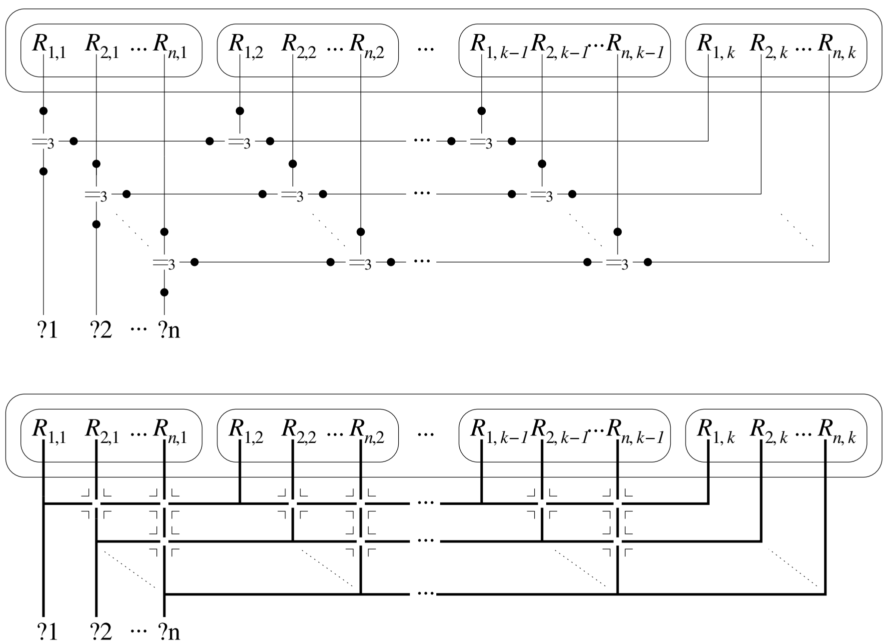


Fig. 26.2. A PAL-GI and the corresponding PAL-graph which evaluate to ϱ

Comment: For this construction, it is even more tedious to write down the corresponding PAL-term. Instead of the general term, we provide only the term for case $ar(\varrho) = 3$ and $|\varrho| = 4$. It is:

$$\begin{aligned}\varrho = \big(&\delta^{1,2} \circ \delta^{2,5} \circ \delta^{3,8} \circ \delta^{4,12} \circ \delta^{5,13} \circ \delta^{6,15} \circ \delta^{7,17} \circ \delta^{8,19} \circ \delta^{9,21} \circ \\ &\delta^{10,21} \circ \delta^{11,24} \circ \delta^{12,27} \circ \delta^{14,20} \circ \delta^{17,23} \circ \delta^{20,26} \circ \delta^{24,29} \circ \delta^{27,33} \circ \delta^{30,37}\big) \\ &\Big(\neg\big[\neg(R_{1,1} \times R_{2,1} \times R_{3,1}) \times \neg(R_{1,2} \times R_{2,2} \times R_{3,2}) \\ &\times\neg(R_{1,3} \times R_{2,3} \times R_{3,3}) \times \neg(R_{1,4} \times R_{2,4} \times R_{3,4})\big] \\ &\times \doteq_3 \times \doteq_3 \times \doteq_3 \times \doteq_3 \times \doteq_3 \times \doteq_3 \times \doteq_3 \times \doteq_3 \times \doteq_3 \Big)\end{aligned}$$

We have shown that with PAL-graphs and relations of arity 1 and 2, each relation can be expressed. But in these graphs, the teridentity, i.e. a ternary relation, is needed as well. The essence of Thm. 26.3 is that, using the PAL$^{-\doteq_3}$-operations on relations, the teridentity cannot be expressed with unary and dyadic relations only. On the side of the graphs, if we consider PAL$^{-\doteq_3}$-graphs over relations of arity 1 or 2, we cannot express the teridentity. This is the subject of the next theorem which is the graph-based counterpart of Thm. 26.3.

Theorem 26.9 (Peirce's Reduction Thesis for Relation Graph Instances). *Let an alphabet $\mathcal{A} := (\mathcal{R}, ar)$ be given with $ar(R) \in \{1,2\}$ for each $R \in \mathcal{R}$. Let $\mathfrak{G}$ be an normed RGI without branching points over $\mathcal{A}$ and let $\mathcal{M} := (U, I)$ be a model over $\mathcal{A}$ with $|U| \geq 2$. Then we have*

$$\mathfrak{R}_{\mathcal{M},\mathfrak{G}} \in \langle \mathcal{M} \rangle_{PAL^{-\doteq_3}} \qquad \text{, particularly} \qquad \mathfrak{R}_{\mathcal{M},\mathfrak{G}} \neq Id_3^U$$

Proof: Due to Lem. 26.7, we can w.l.o.g. assume that $\mathfrak{G}$ is a PAL$^{-\doteq_3}$-GI. Let Σ be the set of all 1- and 2-ary relations over U. Lem. 26.8 immediately yields $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} \in \langle \mathcal{M} \rangle_{\mathrm{PAL}^{-\doteq_3}} \subseteq \langle \Sigma \rangle_{\mathrm{PAL}^{-\doteq_3}}$. Now Thm. 26.3 yields $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} \neq Id_3^U$. □

As relation graphs are classes of RGIs, we immediately obtain the following corollary.

Corollary 26.10 (Peirce's Reduction Thesis for Relation Graphs). *Let an alphabet $\mathcal{A} := (\mathcal{R}, ar)$ be given with $ar(R) \in \{1,2\}$ for each $R \in \mathcal{R}$. Let $\mathfrak{G}$ be an normed relation graph without branching points over $\mathcal{A}$ and let $\mathcal{M} := (U, I)$ be a model over $\mathcal{A}$ with $|U| \geq 2$. Then we have*

$$\mathfrak{R}_{\mathcal{M},\mathfrak{G}} \in \langle \mathcal{M} \rangle_{PAL^{-\doteq_3}} \qquad \text{, particularly} \qquad \mathfrak{R}_{\mathcal{M},\mathfrak{G}} \neq Id_3^U$$

Please note that it is mandatory to consider *normed* relation graphs. The following semi relation graph has no branching points and trivially evaluates to the teridentity in each model.

?1——?2 ?2——?3

26.5 The Contributions of Herzberger and Burch

We have used the result of Hereth Correia and Pöschel in [HCP06] to provide a graph version of Peirce's reduction thesis. Their version of the reduction thesis strictly extends the results of Herzberger [Her81] and Burch [Bur91a]. The differences between these three works shall be discussed in this section. We start with the negative part of the reduction thesis, i.e., that not every ternary relation can be constructed from unary and binary relations only.

To the best of my knowledge, Herzberger was the first to provide mathematical elaboration of Peirce's reduction thesis. Roughly speaking, in Herzberger's approach, the Cartesian product of relations is not considered as an operation on relations. That is, On the side of the graphs, the juxtaposition of graphs is not allowed.

At the beginning of the last chapter, it was already discussed that relations can are joined. According to Herzberger, this was for Peirce the most fundamental operation on relations. So Herzberger first considers a 'miniature setting' (this term is used by Herzberger) where only the join of relations is allowed. To be more precisely: Herzberger uses two join operations: One is taking one relation as argument and joins two blanks of this relation, the second is taking two relation as arguments and joins one blank of the first relation and one blank of the second relation. In both cases, the join of two relations of arity ≤ 2 yields again a relation of arity ≤ 2. Thus it is easy to see that no ternary relation can be constructed by means of unary or binary relations. So the negative part of Peirce's reduction thesis trivially holds in this setting. Herzberger then extends the miniature setting by considering permutations and complements of relations as well. But as these relations do not change the arity of a graph, the negative part of Peirce's reduction thesis still holds for the same simple arity argument.

Burch extends in [Bur91a] the approach of Herzberger by considering the product of relations as well. But in contrast to Hereth Correia and Pöschel, he allows to build the product of relations only as last step or before last step (when the last step is building the complement of the relation) in the construction of relations. Thus, in Burch's setting, again using the arity argument, each ternary relation is the Cartesian product of three unary relations or of an unary and a binary relation, or it is the complement of such an Cartesian product. It can easily be seen that in models with more than one element, the teridentity cannot be of this form, so the negative part of the reduction thesis still holds in this setting.

Hereth Correia and Pöschel finally allow to build the product of relations in *arbitrary* steps in the construction of a relation. The simple arity argument of Herzberger and Burch cannot be applied in this setting anymore, and the proof of the reduction thesis in contrast becomes exceedingly difficult.[2]

It is hard to decide which approach is closest to Peirce's thinking. Herzberger gives good reasons that Peirce did not want to consider operations on relations which violate the arity argument. In fact, Peirce himself argues in some places for the reduction thesis only with this argument (for example in 1.346). Herzberger introduces the notion of 'valency-regular operations', and argues that Peirce aimed to show that valency-irregular operations are dispensable. The Cartesian product of relations is such a valency-irregular operation, so by introducing this operation, Burch departs from Herzberger's point of view. An argument Burch gives is that his approach is closer to Peirce's existential graphs. Nonetheless, as it will be shortly discussed, not all existential graphs can be obtained from Burch's operations.

To clarify matters, let us introduce Herzberger and Burch graphs. Similar to the approach in this chapter, we could introduce operations on RGIs according to the Herzberger's and Burch's operations on relations. Let us call an RGI a HERZBERGER-RGI resp. a BURCH-RGI, if it can be constructed from atomar graphs with Herzberger's or Burch's operations. Obviously, we can lift this definition to RGs by saying that an RG is a HERZBERGER GRAPH resp. a BURCH GRAPH, if the the underlying RGIs are HERZBERGER-RGI resp. a BURCH-RGI. Finally, let us call a RGI $(V, E, \nu, \top, Cut, area, \kappa)$ CONNECTED iff (V, E, ν) is connected, and a relation graph is called CONNECTED if the underlying RGIs are connected. It can be easily proven that all Herzberger graphs are connected. Similarly, each Burch graph is the juxtaposition of connected RGs, or the negation of such a juxtaposition.

Not all graphs Peirce provided in his writings are of this form. In the following, some examples from Peirce are given. In a comment on page 115, we investigated how Peirce discusses in 4.449 how heavy lines crossing a cut are understood. Below, you find another graph of this discussion Peirce provides. Another example can be found on page 22 of [PS00], where Peirce exemplifies the iteration rule (we already used this example on page 153). Two further examples are taken from 4.502, where Peirce provides a total of 25 graphs to explain the transformation rules. Here Figs. 169 and 177 of these graphs are provided (I slightly changed the diagram of Fig. 169: In the diagram in [HB35], the right ligature which is attached to 'respects' and 'knows' ends on the cut, instead of crossing it. This is a degenerate cut. But as Points on a cut are considered outside the cut, I extended the ligature outwardly. See Sec. 11.4). All these graphs are given in Fig. 26.3. None of these graphs is a Herzberger graph or a Burch graph, but they can both be constructed with the operations we provided in Def. 26.4. This shows that neither Herzberger's

[2] In total, it took Hereth Correia nearly three years to find the proof (personal communication).

nor Burch's approach captures all existential or relation graphs Peirce had in mind.[3]

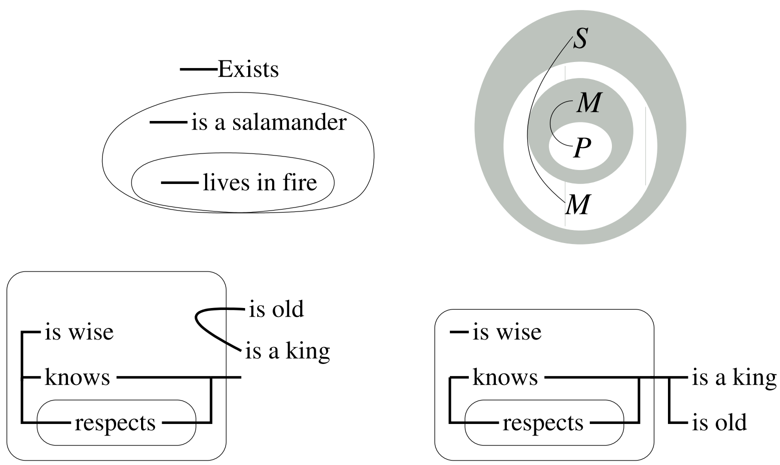


Fig. 26.3. Fig. 84 of 4.449 and a graph of page 22 of [PS00]

Let us now come to the positive part of the reduction thesis, namely that an arbitrary relation ϱ of arity ≥ 4 can be constructed from unary, binary and ternary relations.

In the last section, we have proven the positive part of the reduction thesis by providing two PAL-graphs: The graph of Fig. 26.1 was used for infinite universes, the graph of Fig. 26.2 was used for finite universes. But as the just provided discussion shows, the graph of Fig. 26.2 is not a Herzberger or Burch graph. Both Herzberger and Burch use the graph of Fig. 26.1 to represent a finite relation ϱ as well. Recall that the underlying idea in this graph was to enumerate all tuples of *varrho* by means of elements of the universe. That is, we can represent a relation with the graph of Fig. 26.1 only if the universe U contains at least as many elements as the number of tuples in the relation. For this reason, Herzberger first proves in theorem 4 a restricted version of the positive part, where a relation of arity ≥ 4 can be reduced within *sufficiently large universes*, i.e., for universes U with $|U| \geq |\varrho|$ (particularly, for infinite universes). Nonetheless, even for the case $|U| < |\varrho|$ both Herzberger and Burch use the graph of Fig. 26.1 to represent ϱ. To enumerate the elements of ϱ, they *augment (by means of Peirce's hypostatic abstraction) the universe U*

[3] Moreover, the PAL-system provided in this treatise is closer to the system of relational algebra: The equivalence of the PAL-system and relational algebra has been shown in [HCP04].

with new elements to denote the tuples of ϱ, i.e., they extend the universe U to a larger universe U^+. In fact, Peirce argues similarly for the positive part of his reduction thesis (see for example 1.363). With the herein considered PAL-graphs, this 'trick' is dispensable, as we have shown that for each model $\mathcal{M} = (U, I)$ and each relation $\varrho \subseteq U^n$ (with $n \geq 3$), there exists a PAL-graph $\mathfrak{G}$ with $\mathfrak{R}_{\mathcal{M},\mathfrak{G}} = \varrho$. That is, we can express ϱ *within* $\mathcal{M}$, i.e., without extending U.

Appendix

Referenced Authors

Barwise, 2, 51
Burch, 3, 99, 108, 158, 244, 281, 282, 285, 297–298, 314–316

Chein, 109, 187

Dipert, 26

Etchemendy, 2

Frege, 1, 2, 7, 37

Haack, 37
Hammer, 22
Hereth Correia, 109, 187, 297–299, 314
Herzberger, 297–298, 314–316
Hookway, 26
Howse, 46

Keeler, 6, 25, 26

Mugnier, 109, 187

Pöschel, 109, 187, 244, 285, 297–299, 314
Pape, 26
Pietarinen, 8
Pollandt, 109, 187, 285
Putnam, 2

Roberts, 3, 7, 32, 39, 41, 42, 55, 56, 99, 105, 118
Russell, 3, 34

Shimojima, 21
Shin, 3, 7, 22, 33, 39–42, 46, 55–58, 64, 186
Sowa, 3, 34, 39, 92, 105

Tennant, 1

Whitehead, 3, 34
Wille, 285

Zeman, 3, 8, 39, 55–57, 64, 108, 157, 281

Symbols

(U,I), **141**, **246**
(V,E^{id}), **127**, **269**
$<$, 64, **65**, **125**
E, **124**
$E^{?}$, **285**
$E^{\mathcal{C}}$, **263**
E^{id}, **126**
E^{nonid}, **126**
E_v, **124**
I, **141**
I^f, **286**
$I^f_{\mathcal{C}}$, **286**
$I_{\mathcal{C}}$, **246**
$I_{\mathcal{F}}$, **246**
$I_{\mathcal{R}}$, **141**, **246**
$T_1(V)$, **303**
$T_2(V)$, **303**
$T_3(V)$, **303**
$T_4(V)$, **303**
U, **141**
V, **63**, **124**
V^*, **206**
V^{Var}, **206**
V_e, **124**
$[\mathfrak{G}]_\sim$, **139**, 148
$\mathbf{ff}$, 284
$(V,\top,Cut,area,\kappa)$, **63**
$\mathcal{C}$, **245**, **285**
$\mathcal{F}$, **245**, **285**
$\mathcal{R}$, **126**, 198, **245**, **285**
$\mathcal{A}$, **126**, **245**, **285**
$\mathcal{A}^{=}$, **183**
$\mathcal{A}^{?}$, **285**
$\mathcal{A}^{\doteq_3}$, **302**
Cut, **63**, **124**
$\mathcal{EGI}$, **126**, **246**
$(V,E,\nu,\top,Cut,area,\kappa,\rho)$, **206**, **260**
$\mathcal{VEGI}$, **260**
$\mathcal{VEGI}^{sp,\mathcal{A}}$, **264**
$\cong$, **210**
FV, **197**, **285**
$\mathfrak{G}[c, ref \mathbin{\dot\cup} val]$, **213**
$\mathfrak{G}[c,ref]$, **142, 143**
$\mathfrak{G}[c]$, **75**
$\mathfrak{G}_1 \times \mathfrak{G}_2$, **299**
$\mathfrak{G}_{[v\mapsto\alpha]}$, **213**
Id^U_k, **183**
$\mathcal{M}$, **141**, **246**, **286**
$\mathcal{M}[f]$, **286**
Neg_f, **200**
$\mathrm{Neg}_f \cup \{\top_f\}$, **200**
$(V,E,\nu,\top,Cut,area)$, **124**
$\mathsf{Rel}(U)$, **286**
$\mathfrak{R}_{\mathcal{M},\mathfrak{G}}$, **287**
$\top$, **63**, **124**
$area$, **63**, **124**
$\leq[c]$, **65**, 66, 67, **125**
$\lneq[c]$, **65**, 66, 67, **125**
ctx, **63**, 67, **124**
$\doteq$, **126**, 142, 212, 245, 262
$\doteq_k$, 104, 117, **183**, 184–187
$\models_{class}$, **142**
$\models_{endo}$, 143, **213**
Φ, **209**, 212, 215, 227
Φ_α, **93**
Π, **254**

Ψ, **206**, 213, 215, 226, 227, 253
Ψ_V, **255**
Ψ_α, **92**
$\Psi_\forall$, 218, **218**
Ψ_{ext}, **255**
$\mathtt{tt}$, 284
Θ, **163**, 265, **269**
Θ^*, **269**
$\Vdash_v$, **267**
Ξ, 262, **263**
α_{empty}, **209**
β, **65**, **125**
$\delta^{i,j}(\mathfrak{G})$, **299**
$\delta^{i,j}(\varrho)$, **298**
$\overline{ref}$, **142, 143**, **213**
κ, **63**, **126**, 246, 260, 285
$\langle\mathcal{M}\rangle_{\mathrm{PAL}}$, **299**
$\langle\mathcal{M}\rangle_{\mathrm{PAL}-\doteq_3}$, **299**
$\langle\Sigma\rangle_{\mathrm{PAL}}$, **299**
$\langle\Sigma\rangle_{\mathrm{PAL}-\doteq_3}$, **299**
$\leq$, **65**, 67, 125, **125**, **200**
$\lneq$, **65**, **125**
$\models$, **75**
$\models_?$, 287
$\models_{endo}$, **143**
$\models_{val}$, 201
$\neg\mathfrak{G}$, **299**
$\neg\varrho$, **298**
ν, **124**
$\overline{area}$, **231**
$\pi_\alpha(\mathfrak{G})$, **299**
$\pi_\alpha(\varrho)$, **298**
$?i$, 283, **285**
ρ, **206**, **260**
σ, **286**
$\sim$, **139**, 148, **271**
$\sqsubseteq$, **232**
ϱ, **286**
$\varrho \times \sigma$, **298**
$\vdash$, **79**, **167**, **202**
$\vdash_?$, **288**
$\vdash_e$, **267**
$\vdash_e^r$, **267**
$\vdash_v$, **267**
$\vdash_v^r$, **267**
ar, **126**, **245**
$ar(FDGraph)$, **285**
$ar(\varrho)$, **286**
$f[x_j/x_i]$, **199**
$norm(\mathfrak{G})$, **291**
$palc(\mathfrak{G})$, 303
ref, 142, **142, 143**, 212, **213**, 262
$ref \mathbin{\dot\cup} val$, **213**
v_e, **124**, 249, 285
val, **142**, 212, **213**, **262**
$\mathtt{ff}$, **75**
$\mathtt{tt}$, **75**

Index

$\boldsymbol{\alpha}$-Conversion, **199**

Abduction, 31
adequate, 259
Alpha, 2, 8–11
Alpha Graphs (Shin's Definition), 40
Alphabet, 49, **126**, **245**, **285**
  Query-Marker-Extension, **285**
Area, 9, 10
Area-Space, 128–133
Arity
  of a Function, **245**
  of a Relation, 286
  of a Semi Relation Graph Instance, 285

Begriffsschrift, 1, 7, 37
Beta, 2, 8, 11–14
Beta Graphs (Shin's Definition), 40
Blank, 103, **282**
Blank Form, 108
Branch, **128**
Branching Point, 12, 41, 98, **100**, 101, **101–103**, **128**, 173, 174, 183, 303, 306, 309–310
  Calculus, 157
  Moving, 157, 169, 175
  Moving across Cuts, 158–159
Broken Cut, 15
Burch Graph, 315

Calculus, 10, 33
  Alpha, 10, 76–77
  Beta, 13
  Equivalence Rules, 150
  Existential Graph Instance, **163**
  First Order Logic, **203**, 202–203
  Formal Alpha Graphs, **78**
  Formal Existential Graphs, **167**
  Generalization Rules, 150
  Graphical, 235–237
  Peirce's Understanding, 33
  Semi Relation Graph Instances, **288**
  Shin's Version, 57
  Vertex-Based Existential Graph Instance, **270**
  Vertex-Based Existential Graph Instances, 267–268
Calculus Ratiocinator, 33
Categories, 18, 297
Classical Evaluation, *see* Evaluation
Classification of Sciences, 29
Close, 9
Closed Subgraph, **134**, 134–135
Collected Papers, **6**
Colligation, 33
Comment, **6**
Community, 31, 32
Completeness
  Existential Graph Instance, **228**
  extended Existential Graph Instance, **253**
  First Order Logic, **203**
  Formal Alpha Graphs, **88**, 86–89
  Relation Graph Instances, **292**
  Semi Relation Graph Instances, **288**

Vertex-Based Existential Graph Instances, **272**
Conclusion, 14
Conjunction, 9, 10, 12, 206
Connect, **127**
Connected, **127**
Consistent, **86**
Constant Name, 102, **245**, **285**
Rules, **252**
Constant Vertex, **260**
Context, 66, 67, 125, 150
Existential Graph Instance, **124**
Formal Alpha Graphs, **63**
Negative, **72**, **125**, 150
Ordering, **65**, **125**
Positive, **72**, **125**, 150
Convention No. Zero, 44
Cut, 8, 36
Alpha, 62
Existential Graph Instance, **124**
Formal Alpha Graphs, **63**
Cut Condition, 191
Classical Evaluation, **142**
Endoporeutic Evaluation, **143**
Existential Graph Instance with Variables, **213**
Formal Alpha Graphs, **75**
Isomorphism, **136**
Cut-And-Paste-Theorem, 80
Cut-Line, **128**, 128–133
Cuts, 111–121
Cycle, **127**

Deduction, 31
Deduction Theorem, 80
Degenerated Graph, **117**, 120, 121
Deiterating Constant Labels, 275
Delta, 8
Derive
Existential Graph Instance, **167**
First Order Logic, **202**
Formal Alpha Graphs, **79**
Diagram, 17–18, **21**, 35, 39–53
Barwise and Etchemendy, 2
Chemistry, 282
Existential Graph Instances, 128–133, 229–240
Formal Alpha graphs, 61–62, **68**, 67–70
Formal Existential Graph, 132–133, 229–240
Vertex-Based Existential Graph Instance, 260–261
Diagrammatic Reasoning, 1, 2, 17, 25
Directly Enclosed, 134
Alpha, **65**
Existential Graph Instance, **125**
Dominating Nodes, **125**, 143
Double Cut, 10, 13, 159–161

Edge, 109–110
Existential Graph Instance, **124**
Pending, 285
Edge Condition, 191
Classical Evaluation, **142**
Endoporeutic Evaluation, **143**
Existential Graph Instance with Variables, **213**
Isomorphism, **136**
Edge-Line, **129**, 128–133
Enclosed, 128, 134, 231
Alpha, 62, **65**
Directly, 134
Evenly, 10, **72**, **125**, 134
Existential Graph Instance, 125, **125**
Oddly, 10, **72**, **125**, 134
Endoporeutic Evaluation, **143**, **213**
Endoporeutic Method, 9, 142, **143**, **213**
Erasure, 33
Euclidean Plane, 45, 46
Euler Circles, 1
Evaluation, **143**, **213**
Classical, **142**
Equivalence, 144
First Order Logic, **201**
Formal Alpha Graphs, **75**
Propositional Logic, **92**
Evenly Enclosed, *see* Enclosed
Existential Graph, 210
Existential Graph Instance, **126**, 126–133
Representation, 128–133
Existential Graph Instance (extended), **246**
Existential Graph Instance with Variables, 206

First Order Logic, 2, 8, 49, 142, 197–203, 210
Firstness, 18, 297
Forest, **127**
Formal Existential Graph, **139**
Formula
  Structure, 200
Formulas, 49
  Equivalence Relation, **210**
  First Order Logic, **197**
  Propositional Logic, **91**
  Subformula Occurrences, 199
  Universal Closure, 218
Free Ride, 21
Fresh, 138
Function Name, **245**, **285**
  Rules, **249**

Gamma, 2, 8, 14–16
Generalizing Labels of Vertices, 275
Generic Marker, **206**
Generic Vertex, 206, **260**
Grapheus, 37
Graphist, 37

Heavily Drawn Line, 97, 105
  Crossing a Cut, 111–121
Heavily Marked Point, 98, 99, 107, 161–162
  On a Cut, 112
Herzberger Graph, 315
Hook, 108, **108**, **128**
  Attached to, **128**
  Replaced on, **128**, 169
Hypostatic Abstraction, 15, 316
Hypothetical Universe, 28, 29, 34, 37

Icon, 18, **19, 20**
Identity, 97–100, 104, 109, 111–117, **126**, 179, 183, 203, **245**, **285**
Identity Spot, 99, 100, 106, 107
  On a Cut, 112
Identity-Edge, **126**, 129
Implication, 28, 36
Index, 18, **19**
Induction, 30
Inductive Definition, 63, 309
Informal Definition, **6**
Interpretant, 19, 20
Interpretation, 141
Isomorphism, **68**, **135**
  Except Cuts, **71**, **136**, 145
  Partial, **71**, **136**, 145
Iterating Constant Labels, 275
Iteration, 33

Join of Relatives, *see* Relatives
Judgement, 8
Juxtaposition, 9, 35, 81, 206
  Existential Graph Instance, **136**
  Formal Alpha Graphs, **73**

Knowledge, 27

Leaf, **127**
Ligature, 12, 98–107, **127**, 169–187, **269**
  Calculus, **153**, **155**, 150–159
  Crossing a Cut, 111–121
  Extending or Restricting, 172, 277
  Generic, **269**
  Idiotic, 100
  Joining, 184
  Rearranging, 174, 277
  Retracting, 173, 174, 181, 277
  Separating, 183–187
  Single-Object, 179, **180**, 181, 184–187
  Vertex-Based Existential Graph Instances, 274–279
Ligature-Graph, **127**, **269**
Line of Identity, 11, 35, 98–107
  'Crossing a Cut', 111–121
Logic, 26
Loop, **127**
Loose End, 103, 155, **282**

Mathematical Reasoning, 1, 2, 17, 27, 29, 32
Medad, 8
Merging two Vertices, **175**, 176
Meta-Level Proposition, 15
Modal Logic, 3, 15
Model, 28
  Beta, 201
  Existential Graph Instance, **141**
  extended Existential Graph Instance, **246**
  Formal Alpha Graphs, 75
  Relational, **141**, **246**
Modus Ponens, 10, 218

Necessary Reasoning, 27–29, 32, 36
Negation, 9, 10, 12
Negative Context, *see* Context
Nesting, 9
  Alpha, 64
Non-Degenerated Graph, **117**, 120, 121, 230
Normalization, 289, **291**

Object Vertex, **260**
Oddly Enclosed, 10, 134, *see* Enclosed

PAL, 244, 281
  Complexity, 303
  Operations on Graphs, **299**
  Operations on Relations, **298**
PAL Graph Instance, **302**, 303, **306**
  Inductive Semantics, **307**
$\mathrm{PAL}^{-\{\dot{=}_3\}}$ Graph Instance, 302, 306
  Inductive Semantics, **307**
Partial Isomorphism, *see* Isomorphism
Partial Valuation, *see* Valuation
Path, **127**
Peirce's Reduction Thesis, 101, 103, 285
  Algebraic Version, **299**
  for Relation Graph Instances, **313**
  for Relation Graphs, **313**
Peircean Algebraic Logic, *see* PAL
Positive Context, *see* Context
Pragmatism, 25
Predicate, 11, 107–110
Predicate Spot, 108
Principia Mathematica, 3, 34
Proof
  Existential Graph Instance, **167**
  First Order Logic, **202**
  Formal Alpha Graphs, **79**
Proposition, 8, 10, 35, 108
Propositional Logic, 2, 10, **91**, 91–95
Propositional Variables, **61**, 62, **91**
  Occurences, 62
Provably Equivalent
  Existential Graph Instance, **167**
  First Order Logic, **202**
  Formal Alpha Graphs, **79**
Psychologism, 37

Quantification, 12, 102, 142, 198, 203, 218
  Peirce and Mitchell, 2
Query Marker
  Assignment, **286**
  Calculus, **288**
  Extension of a Model, **286**
  Extension of an Alphabet, **285**
  Restriction of a Model, **286**

Rational Communication, 31
Reasoning, 1, 2, 25, 27–30, 32, 33, 37
  Patterns, 33
  Self-Correcting, 30, 31
Relation, 107–110, 282–283, **286**
  Arity, **286**, **314**
  Complement, **298**
  Finitary, **286**
  Join, 282–283, **298**, 314
  Permutation, **298**
  Product, **298**
Relation Graph Instance, 284, **289**, 306
  Atomar Graph, **299**
  Complement, **299**
  Join, **299**
  Normed, **289**, 303
  Permutation, **299**
  Product, **299**
Relation Name, **126**, 198, **245**, **285**
Relational Graph with Cuts, 246, 260
Relational Graphs with Cuts, **124**, 123–125
Relational Structure, *see* Model
Relations, 107, 243–244, 281
  Product, 297
Relatives, 103, 104, 107, 243–244, 281–283
Replica, **23**, 39, **44**, 41–45, 230
Representation Problem, 45, 49–51
Reversion Theorem, 80
Rule of Deformation, 44

Scribing, 8
Scroll, 10, 36, 41, 208
Secondness, 18, 297
Semantics
  Existential Graph Instance, **142, 143**
  Existential Graph Instance with Variables, **213**
  First Order Logic, **201**
  Formal Alpha Graphs, **75**

Formal Existential Graphs, **148**
Semi Relation Graph Instance, 284, **285**
Semiotics, 17–23, 26
Sep, 8, 44
Separated Constant Vertices
seeVertex, 317
Sheet of Assertion, 8, 10, 34, 124
Existential Graph Instance, **124**
Formal Alpha Graphs, **63**
Sign, **23**, 17–23, 27
Single-Object Ligature, 179, **180**, 181, 184–187
Soundness
Constant Vertex Rule, **266**
Existential Graph Instance, **196**, 189–196
First Order Logic, **203**
Formal Alpha Graphs, **85**, 83–85
Function Rule, **250**
Main Lemma for Alpha, 83
Main Lemma for Beta, 145, 147
Relation Graph Instances, **291**
Semi Relation Graph Instances, **288**
Transformation Rules for Ligatures, **147**
Vertex-Based Existential Graph Instances, **272**
Specializing Labels of Vertices, 275
Splitting a Vertex, **175**, 176
Standard-Form, **208**, 227
Standardization, **208**
Star, **183**
Subformulas
First Order Logic, **197**
Subgraph, 42, **127**, **134**, 134–135
Alpha, **70**, 70–71
Beta, **134**, 134–135
Calculus, 151–152
Closed, **134**
Existential Graph Instance, **134**
Representation, 230–235
Subgraph-Line, 232, **233, 234**
Substitutions, **199**
Syllogism, 13, 29
Symbol, 7, 18, **19**, 27
Symbolic Logic, 1, 2, 7, 48–51
Syntactical Entailment, 218, 226
Existential Graph Instance, **167**
First Order Logic, **202**
Formal Alpha Graphs, **79**

Teridentity, 99, 101–103, 117, 157, 174, 184, **299**, 309–310
Terms, 197, 253–258
Thirdness, 18, 297
Tinctures, 15
Token, **23**, 45
Token-Structure, **45**
Total Valuation, *see* Valuation
Transformation Rules, 32, 33
Transformation Rules for Ligatures, **138**, 147
Translation, 263
Translations, 91–95, 205–215
Meaning-Preserving, 94, 212, 213, 215, 264
Truth Values, **28**, **75**
Type, **23**, 45
Type-Equivalence, **46**
Type-Structure, **45**
Type-Token Issue, 23

Universal Language, 37
Universe of Discourse, 16, 34, 141

Valid, **75**, 92, 201
Valuation
Extended Partial, **142**, **212**, **262**
First Order Logic, **201**
Partial, **75**, **142**, 145, **212**, **262**
Total, **75**, **142**, **212**, **262**
Variable Vertex, 206
Variables, **197**, 201, 205, 206
Bound, **197**
Free, **197**
Venn Diagrams, 1
Vertex, 109
Existential Graph Instance, **124**
Formal Alpha Graphs, **63**
Generic, 206, 271
Separated Constant, **264**
Variable, 206
Vertex Condition
Formal Alpha Graphs Graphs, **75**
Vertex-Based Existential Graph Instances, 259, **260**
with Separated Constant Vertices, **264**
Vertex-Spot, **128**, 128–133

References

[AB96] G. Allwein and J. Barwise, editors. *Logical Reasoning With Diagrams*. Oxford University Press, 1996.

[ACH00] Michael Anderson, Peter Cheng, and Volker Haarslev, editors. *Theory and Application of Diagrams: First International Conference, Diagrams 2000*, volume 1889 of *Lecture Notes in Computer Science*. Springer, Berlin – Heidelberg – New York, 2000.

[Arn01] Markus Arnold. Einführung in die kontextuelle relationenlogik. Master's thesis, Darmstadt University of Technology, 2001.

[Bar93] John Barwise. Heterogenous reasoning. In Mineau et al. [MMS93], pages 64–74.

[BE98] J. Barwise and J. Etchemendy. Computers, visualization, and the nature of reasoning. In Terrell Ward Bynum, Don D. Roberts, and James H. Moor, editors, *The Digital Phoenix: How Computers are Changing Philosophy*, pages 93–116. Blackwell, London, 1998.

[BF97] Jacqueline Brunning and Paul Forster, editors. *The Rule of Reason: The Philosophy of Charles Sanders Peirce*. University of Toronto Press, 1997.

[BMS04] Alan Blackwell, Kim Marriott, and Atsushi Shimojima, editors. *Diagrammatic Representation and Inference: Third International Conference, Diagrams 2004*, volume 2980 of *LNAI*. Springer, Berlin – Heidelberg – New York, 2004.

[Bra01] R. B. Brandom. *Begründen und Begreifen. Eine Einführung in den Inferentialismus.* Suhrkamp, Frankfurt, 2001.

[Bru76] W. Brugger. *Philosophisches Wörterbuch.* Herder Verlag, Freiburg, 1976. 14. Auflage.

[Bur91a] Robert W. Burch. *A Peircean Reduction Thesis: The Foundation of Topological Logic*. Texas Tech. University Press, Texas, Lubbock, 1991.

[Bur91b] Robert W. Burch. Valency, adicity, and adity in peirce's ms 482. *Transactions of the Charles S. Peirce Society*, 27(2):237–244, 1991.

[Bur92] Robert W. Burch. Valental aspects of peircean algebraic logic. *Computers and Mathematics with Applications*, 23(6–9):665–677, 1992.

[Bur94a] Robert W. Burch. *Die Logischen Operationen in Peirce' 'Beschreibung einer Notation fuer die Logik der Relative'*, pages 77–113. Suhrkamp, Frankfurt, 1994.

[Bur94b] Robert W. Burch. Game theoretical semantics for peirce's existential graphs. *Synthese*, 99:361–375, 1994.

[Bur97a] Robert W. Burch. Peirce on the application of relations to relations. In Houser et al. [HRE97].

[Bur97b] Robert W. Burch. Peirce's reduction thesis. In Houser et al. [HRE97].

[Bur97c] Robert W. Burch. A tarski-style semantics for peirce's beta graphs. In Brunning and Forster [BF97].

[CM92] M. Chein and M.-L. Mugnier. Conceptual graphs: Fundamental notions. *Revue d'Intelligence Artificiell*, 6:365–406, 1992.

[CM95] M. Chein and M.-L. Mugnier. Conceptual graphs are also graphs. Technical report, LIRMM, Université Montpellier II, 1995. Rapport de Recherche 95003.

[Dau03a] Frithjof Dau. Concept graphs without negations: Standardmodels and standardgraphs. In de Moor et al. [dMLG03], pages 243–256. This paper is a part of [Dau03b] as well.

[Dau03b] Frithjof Dau. *The Logic System of Concept Graphs with Negations and its Relationship to Predicate Logic*, volume 2892 of *LNAI*. Springer, Berlin – Heidelberg – New York, November 2003.

[Dau04a] Frithjof Dau. Background knowledge in concept graphs. In Eklund [Ekl04], pages 156–171.

[Dau04b] Frithjof Dau. Query graphs with cuts: Mathematical foundations. In Blackwell et al. [BMS04], pages 32–50.

[Dau04c] Frithjof Dau. Types and tokens for logic with diagrams: A mathematical approach. In Karl Erich Wolff, Heather D. Pfeiffer, and Harry S. Delugach, editors, *Conceptual Structures at Work: 12th International Conference on Conceptual Structures*, volume 3127 of *Lecture Notes in Computer Science*, pages 62–93. Springer, Berlin – Heidelberg – New York, 2004.

[Dau06a] Frithjof Dau. Fixing shin's reading algorithm for peirce's existential graphs. In Dave Barker-Plummer, Richard Cox, and Nik Swoboda, editors, *Diagrams*, volume 4045 of *LNAI*, pages 88–92. Springer, Berlin – Heidelberg – New York, 2006.

[Dau06b] Frithjof Dau. The role of existential graphs in peirce's philosophy. In Peter Øhrstrøm, Henrik Schärfe, and Pascal Hitzler, editors, *ICCS*, pages 28–41. Aalborg University Press, 2006.

[Dau06c] Frithjof Dau. Some notes on proofs with alpha graphs. In Øhrstrøm et al. [ØSH06], pages 172–188.

[Dev91] K. Devlin. *Logic and Information*. Cambridge University Press, Cambridge, 1991.

[DHC03] Frithjof Dau and Joachim Hereth Correia. Nested concept graphs: Mathematical foundations and applications for databases. In Ganter and de Moor [GdM03], pages 125–142.

[DHC06] Frithjof Dau and Joachim Hereth Correia. Two instances of peirce's reduction thesis. In Rokia Missaoui and Jürg Schmid, editors, *ICFCA*, volume 3874 of *Lecture Notes in Computer Science*, pages 105–118. Springer, 2006.

[Dip04] Randall Dipert. Peirce's deductive logic: Its development, influence, and philosphical significance. In Cheryl L. Misak, editor, *The Cambridge Companion to Peirce*, pages 287–324. Cambridge University Press, Cambridge, 2004.

[DK03] Frithjof Dau and J. Klinger. From formal concept analysis to contextual logic. In Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors, *Formal Concept Analysis: The State of the Art*, volume 3626 of *Lecture Notes in Computer Science*. Springer, Berlin – Heidelberg – New York, 2003.

[dMLG03] Aldo de Moor, Wilfried Lex, and Berhard Ganter, editors. *Conceptual Structures for Knowledge Creation and Communication*, volume 2746 of *LNAI*. Springer, Berlin – Heidelberg – New York, 2003.

[DS01] Harry S Delugach and Gerd Stumme, editors. *Conceptual Structures: Broadening the Base*, volume 2120 of *LNAI*, Stanford, USA, July,, 2001. Springer, Berlin – Heidelberg – New York.

[EEM96] P. W. Eklund, G. Ellis, and G. Mann, editors. *Conceptual Structures: Knowledge Representation as Interlingua*, volume 1115 of *LNAI*. Springer, Berlin – Heidelberg – New York, 1996.

[Eis76] C. Eisele, editor. *Charles Sanders Peirce: The new Elements of Mathematics*. Hague: Mouton Publishers; Atlantic Highlands, N.J.: Humanities Press, 1976.

[Ekl04] Peter W. Eklund, editor. *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*, volume 2961 of *Lecture Notes in Computer Science*. Springer, Berlin – Heidelberg – New York, 2004.

[ELRS95] G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors. *Conceptual Structures: Applications, Implementation, and Theory*, volume 954 of *LNAI*. Springer, Berlin – Heidelberg – New York, 1995.

[Fri92] U. Friedrichsdorf. *Einführung in die klassische und intensionale Logik*. Vieweg, 1992.

[GdM03] Bernhard Ganter and Aldo de Moor, editors. *Using Conceptual Structures: Contributions to ICCS 2003*, Borovets, Bulgaria, July, 15–19, 2003. Shaker Verlag, Aachen.

[GM00] Bernhard Ganter and Guy W. Mineau, editors. *Conceptual Structures: Logical, Linguistic and Computational Issues*, volume 1867 of *LNAI*, Darmstadt, Germany, 2000. Springer, Berlin – Heidelberg – New York.

[GM01] B. Ganter and G. Mineau, editors. *Conceptual Structures: Extracting and Representing Semantics*. Manuscript Stanford, 2001.

[GW99] Bernhard Ganter and Rudolf Wille. Contextual attribute logic. In Tepfenhart and Cyre [TC99], pages 377–388.

[Haa78] Susan Haack. *Philosophy of Logics*. Cambridge University Press, 1978.

[Ham95] Eric M. Hammer. *Logic and Visual Information*. CSLI Publications, Stanford, California, 1995.

[Ham98] E. M. Hammer. Semantics for existential graphs. *Journal Philosophical Logic*, 27:489–503, 1998.

[Haw94] Derik Hawley. Logic in pictures: An examination of diagrammatic representations, graph theory and logic. Master's thesis, Waterloo, Canada, 1994. Available at: `http://etd.uwaterloo.ca/etd/dhawley1994.pdf`.

[HB35] Weiss Hartshorne and Burks, editors. *Collected Papers of Charles Sanders Peirce*, Cambridge, Massachusetts, 1931–1935. Harvard University Press.

[HCP04] Joachim Hereth Correia and Reinhard Pöschel. The power of peircean algebraic logic (pal). In Eklund [Ekl04], pages 337–351.

[HCP06] Joachim Hereth Correia and Reinhard Pöschel. The teridentity and peircean algebraic logic. In Øhrstrøm et al. [ØSH06].

[Her81] Hans G. Herzberger. Peirces remarkable theorem. In F. Wilson L. W. Sumner, J. G. Slater, editor, *Pragmatism and Purpose: Essays Presented to Thomas A. Goudge*. University of Toronto Press, Toronto, 1981.

[Hin85] J. Hintikka. *The Game of Language: Studies in Game-Theoretical Semantics and its Applications*. D. Reidel, Dordrecht, 1985.

[Hin97] J. Hintikka. The place of c. s. peirce in the history of logical theory. In Brunning and Forster [BF97], pages 13–33.

[HMN02] Mary Hegarty, Bernd Meyer, and N. Hari Narayanan, editors. *Diagrammatic Representation and Inference: Second International Conference, Diagrams 2002, Proceedings*, volume 2317 of *Lecture Notes in Computer Science*. Springer, Berlin – Heidelberg – New York, 2002.

[HMST02] J. Howse, F. Molina, Sun-Joo Shin, and J. Taylor. On diagram tokens and types. In Hegarty et al. [HMN02], pages 146–160.

[Hoo85] Christopher Hookway. *Peirce*. London: Routledge and Kegan Paul, 1985.

[HRE97] Nathan Houser, Don D. Roberts, and James Van Evra, editors. *Studies in the Logic of Charles Sanders Peirce*. Indiana University Press, 1997.

[Jec78] T. Jech. *Set Theory*. Academic Press, 1978.

[Kee] Mary Keeler. The philosophical context of peirce's existential graphs. Available at:
`http://www.dipf.de/projekte/Paed_Sem_HCI/Texte/Keeler_context.htm`.

[Ket90] Kenneth Laine Ketner. *Elements of Logic: An Introduction to Peirce's Existential Graphs*. Texas Tech University Press, Lubbock, Texas, 1990.

[KGS90] L. Kreiser, S. Gottwald, and W. Stelzer. *Nichtklassische Logik*. Akademie–Verlag Berlin, 1990.

[Kul94] Zenon Kulpa. Diagrammatic representation and reasoning. *Machine Graphics and Vision*, 3(1/2):77–103, 1994.

[Lev79] A. Levy. *Basic Set Theory*. Springer, Berlin – Heidelberg – New York, 1979.

[Luk97] D. Lukose, editor. *Conceptual Structures: Fulfilling Peirce's Dream*, volume 1257 of *LNAI*. Springer, Berlin – Heidelberg – New York, 1997.

[M9̈7] Ralf Müller. Peirce's existential graphs: First inquiries towards a proper understanding. In Michael van Heerden Jaap van Brakel and, editor, *C.S. Peirce, categories to constantinople, proceedings of the international symposium of Peirce*. Leuven University Press, 1997.

[Mat69] B. Mates. *Elementare Logik*. Vandenhoeck & Ruprecht, 1969.

[MC98] Marie-Laure Mugnier and Michel Chein, editors. *Conceptual Structures: Theory, Tools and Applications, 6th International Conference on Conceptual Structures, ICCS '98, Montpellier, France, August 10-12, 1998, Proceedings*, volume 1453 of *LNAI*. Springer, Berlin – Heidelberg – New York, 1998.

[Men64] E. Mendelson. *Introduction to Mathematical Logic*. D. Van Nostrand Company, Princeton, 1964.

[ML] Robert Marty and Bern Lang. 76 definitions of the sign by c. s. peirce. Available at:
`http://members.door.net/arisbe/menu/library/rsources/76defs/76defs.htm`.

[MMS93] Guy Mineau, Bernard Moulin, and John F. Sowa, editors. *Conceptual Graphs for Knowledge Representation*, volume 699 of *LNAI*, Quebec City, Canada, August 1993. Springer, Berlin – Heidelberg – New York.

[Mon69] J. D. Monk. *Introduction to Set Theory*. McGraw–Hill, 1969.

[Nor04] Jesse Norman. The iconic logic of peirce's graphs. *Mind*, 113(452):783–787 (5), 2004.

[ØBD99] Peter Øhrstrøm, Torben Bräuner, and Claus Donner. A software system for learning peircean graphs. In Tepfenhart and Cyre [TC99], pages 184–197.

[Oeh93] K. Oehler. *Charles Sanders Peirce*. Beck'sche Reihe, 1993.

[Øh95] Peter Øhrstrøm. Logic and existential graphs. *Topics in Cognitive Science and HCI*, 6:137–154, 1995.

[Øh96] Peter Øhrstrøm. Existential graphs and tense logic. In Eklund et al. [EEM96].

[Øh97] Peter Øhrstrøm. C. s. peirce and the quest for gamma graphs. In Lukose [Luk97], pages 357–370. Available at:
`http://www.hum.auc.dk/ poe/ARTIKLER/GammaHTML/GammaGraphs.html`.

[ØSH06] Peter Øhrstrøm, Henrik Schärfe, and Pascal Hitzler, editors. *Conceptual Structures: Inspiration and Application*, volume 4068 of *Lecture Notes in Computer Science*. Springer, Berlin – Heidelberg – New York, 2006.

[ØSvdB94] Peter Øhrstrøm, Jan Schmidt, and Harmen van den Berg. Some peircean problems regarding graphs for time and modality. In Tepfenhart et al. [TDS94], pages 78–92.

[Pap83] Helmut Pape. *Charles S. Peirce: Phänomen und Logik der Zeichen*. Suhrkamp Verlag Wissenschaft, Frankfurt am Main, Germany, 1983. German translation of Peirce's Syllabus of Certain Topics of Logic.

[PCA02] Uta Priss, Dan Corbett, and Galia Angelova, editors. *Conceptual Structures: Integration and Interfaces*, volume 2393 of *LNAI*, Borovets, Bulgaria, July, 15–19, 2002. Springer, Berlin – Heidelberg – New York.

[Pei35] Charles Sanders Peirce. *MS 478: Existential Graphs.* Harvard University Press, 1931–1935. Partly published in of [HB35] (4.394-417). Complete german translation in [Pap83].

[Pei92] Charles Sanders Peirce. Reasoning and the logic of things. In K. L. Kremer and H. Putnam, editors, *The Cambridge Conferences Lectures of 1898*. Harvard Univ. Press, Cambridge, 1992.

[Pie04] Ahti-Veikoo Pietarinen. Peirce's diagrammatic logic in if perspecive. In Blackwell et al. [BMS04], pages 97–111.

[Pie06] Ahti-Veikoo Pietarinen. The endoporeutic method. Published on 'the digital Peirce'. Available at:
`http://www.digitalpeirce.fee.unicamp.br/endo.htm`., 2006.

[Pol01] Silke Pollandt. Relational constructions on semiconcept graphs. In Ganter and Mineau [GM01].

[Pol02] Silke Pollandt. Relation graphs: A structure for representing relations in contextual logic of relations. In Priss et al. [PCA02], pages 24–48.

[Pre98] Susanne Prediger. *Kontextuelle Urteilslogik mit Begriffsgraphen – Ein Beitrag zur Restrukturierung der Mathematischen Logik*. Shaker Verlag, Aachen, 1998. Dissertation, Darmstadt University of Technology.

[Pre00] Susanne Prediger. Mathematische logik in der wissensverarbeitung: Historisch-philosophische gründe für eine kontextuelle logik. *Mathe. Semesterberichte*, 47, 2000.

[PS00] Charles Sanders Peirce and John F. Sowa. Existential Graphs: MS 514 by Charles Sanders Peirce with commentary by John Sowa, 1908, 2000. Available at: `http://www.jfsowa.com/peirce/ms514.htm`.

[Put82] Hilary Putnam. Peirce the logician. *Historia Mathematica*, 9:290–301, 1982.

[Rau96] W. Rautenberg. *Einführung in die mathematische Logik.* Vieweg Lehrbuch Mathematik, 1996.

[Rob73] Don D. Roberts. *The Existential Graphs of Charles S. Peirce*. Mouton, The Hague, Paris, 1973.

[Rob92] Don D. Roberts. The existential graphs. *Computers Math. Appl..*, 23(6–9):639–63, 1992.

[Shi] Sun-Joo Shin. Diagrams and a theory of seeing. Available at: `http://www.cs.tcd.ie/Tim.Fernando/B/shin.pdf`.

[Shi94] Sun-Joo Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.

[Shi96] Atsushi Shimojima. *On the Efficacy of Representation*. PhD thesis, The Department of Philosophy, Indiana University, 1996. Available at: `http://www.jaist.ac.jp/ ashimoji/e-papers.html`.

[Shi99] Sun-Joo Shin. Reconstituting beta graphs into an efficacious system. *Journal of Logic, Language and Information*, 8(3), 1999.

[Shi00] Sun-Joo Shin. Reviving the iconicity of beta graphs. In Anderson et al. [ACH00].

[Shi02a] Sun-Joo Shin. *The Iconic Logic of Peirce's Graphs*. Bradford Book, Massachusetts, 2002.

[Shi02b] Sun-Joo Shin. Multiple readings in peirce's alpha graphs. In Michael Anderson, Bernd Meyer, and Patrick Olivier, editors, *Diagrammatic Representation and Reasoning*. Springer, Berlin – Heidelberg – New York, 2002.

[Sho67] J. R. Shoenfield. *Mathematical Logic*. Addison–Wesley, 1967.

[Sow] John F. Sowa. Semantic foundations of contexts. This paper is a revised merger of [Sow95] and [Sow97b]. Available at: `http://www.jfsowa.com/ontology/contexts.htm`.

[Sow84] John F. Sowa. *Conceptual structures: information processing in mind and machine*. Addison-Wesley, Reading, Mass., 1984.

[Sow92] John F. Sowa. Conceptual graphs summary. In T. E. Nagle, J. A. Nagle, L. L. Gerholz, and P. W. Eklund, editors, *Conceptual Structures: current research and practice*, pages 3–51. Ellis Horwood, 1992.

[Sow95] John F. Sowa. Syntax, semantics, and pragmatics of contexts. In Ellis et al. [ELRS95], pages 1–15. See also [Sow].

[Sow97a] John F. Sowa. Logic: Graphical and algebraic. manuscript, Croton-on-Hudson, 1997.

[Sow97b] John F. Sowa. Peircean foundation for a theory of context. In Lukose [Luk97], pages 41–64. See also [Sow].

[Sow00] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole, Pacific Grove, CA, 2000.

[SPØ02] Henrik Schärfe, Ulrik Petersen, and Peter Øhrstrøm. On teaching conceptual graphs. In Priss et al. [PCA02], pages 285–298.

[Ste96] John Stewart. Theorem proving using existential graphs. Master's thesis, University of California at Santa Cruz, 1996. advisor: Robert Levinson.

[Stu00] Gerd Stumme, editor. *Working with Conceptual Structures. Contributions to ICCS 2000*. Shaker Verlag, Aachen, 2000.

[SW00] G. Stumme and R. Wille, editors. *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*. Springer, Berlin – Heidelberg – New York, 2000.

[Tar41] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, 1941.

[TC99] Wiliam Tepfenhart and W. Cyre, editors. *Conceptual Structures: Standards and Practices*, volume 1640 of *LNAI*. Springer, Berlin – Heidelberg – New York, 1999.

[TDS94] William M. Tepfenhart, Judith P. Dick, and John F. Sowa, editors. *Conceptual Structures: Current Practices*, volume 835 of *LNAI*, College Park, USA, August 1994. Springer, Berlin – Heidelberg – New York.

[TW94] H. P. Tuschik and H. Wolter. *Mathematische Logik – kurzgefasst*. BI Wissenschaftsverlag, Mannheim–Leipzig–Wien–Zürich, 1994.

[vD96] D. van Dalen. *Logic and Structure*. Springer, Berlin – Heidelberg – New York, 1996.

[VP98] Adam Vile and Simon Polovina. Possibilites in peirce's existential graphs for logic education. available from the authors, 1998.

[Wer95] M. Wermelinger. Conceptual graphs and first-order logic. In Ellis et al. [ELRS95], pages 323–337.

[Wil94] Rudolf Wille. Plädoyer für eine philosophische grundlegung der begrifflichen wissensverarbeitung. In R. Wille and M. Zickwolff, editors, *Begriffliche Wissensverarbeitung: Grundfragen und Aufgaben*, pages 11–25. B.I.–Wissenschaftsverlag, Mannheim, 1994.

[Wil96] Rudolf Wille. Restructuring mathematical logic: An approach based on peirce's pragmatism. In A. Ursini and P. Agliano, editors, *Logic and Algebra*, pages 267–281. Marcel Dekker, New York, 1996.

[Wil97] Rudolf Wille. Conceptual graphs and formal concept analysis. In Lukose [Luk97], pages 290–303.

[Wil99] Rudolf Wille. Conceptual landscapes of knowledge: a pragmatic paradigm for knowledge processing. In W. Gaul and H Locarek-Junge, editors, *Classification in the Information Age*, pages 344–356. Springer, Berlin – Heidelberg – New York, 1999.

[Wil00] Rudolf Wille. Contextual logic summary. In Stumme [Stu00], pages 265–276.

[Wil01] Rudolf Wille. Lecture notes on contextual logic of relations. Fb4-preprint, Darmstadt University of Technology, 2001.

[Zem64] Jay J Zeman. *The Graphical Logic of C. S. Peirce*. PhD thesis, University of Chicago, 1964. Available at: `http://www.clas.ufl.edu/users/jzeman/`.

[Zem86] Jay J Zeman. Peirce's philosophy of logic. *Transactions of the Charles S. Peirce Society*, 22:1–22, 1986.

[Zem95] Jay J Zeman. Existential graphs and thirdness. *Semiotica*, 105:311–19, 1995.

[Zem97] Jay J Zeman. Peirce and philo. In Houser et al. [HRE97]. Available at: `http://web.clas.ufl.edu/users/jzeman/peirce_and_philo.htm`.