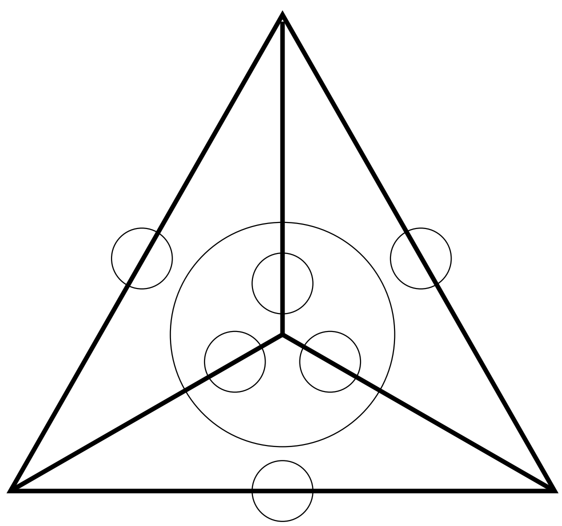
Mathematical Logic with Diagrams

Based on the Existential Graphs of Peirce

Frithjof Dau, TU Darmstadt, Germany



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

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 [**?**]

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 [**?**]

This treatise advocates this view that rigor formal logic can 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 rigour of contemporary mathematics. There are several reasons for choosing Peirce's existential graphs for the purpose of this treatise. These reasons shall be elucidated in the following.

Peirce had been a philosopher and mathematician who devoted his life to the investigation of reasoning and the growth of knowledge. He war 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,

[1] For example, he invented, independently from Frege, together with his student O. H. Mitchell a notation for existential and universal quantification. (eigentlich vier Jahre nach Frege, aber als erste in ‘the effective sense (H. Putnam 1982 wwww)).

and Beta corresponds to first-order predicate logic.[2] The part Gamma is more complicated: It contains features of higher order and modal logic, and 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 had been 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 existential graphs 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 existential graphs has first to be be reworked and reformulated as a precise theory of mathematical logic, before the correspondence to the symbolic logic we use nowadays is mathematically formulated and proven.

Several authors like Zeman, Roberts, Sowa, Burch or Shin have explored the system of existential graphs. 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 existential graphs 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. Existential graphs will be developed as a formal, but diagrammatic, mathematical logic, including a well-defined syntax, an extensional semantics, a sound and complete calculus, and 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 existential graphs, 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 Chap. **??**, 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

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

of logic. Moreover, it gives us a first hint how diagrams can be mathematically formalized. This will be more thoroughly discussed in Chap. ??. In this chapter, the use of representations in mathematical logic is discussed, and two different, possible approaches for a formalization of diagrams are investigated 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 graphs. 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 existential graphs (this is similar to Robert's approach in [?]. See also Chap. ??). Due to this aim, in Chap. ?? it is discussed which place Peirce's systems of existential graphs in his whole philosophy has (this chapter relies on Peirce's semiotics which is described in Chap. ??, but it is a separate chapter in the sense as the remaining treatise hardly refers to it), and Peirce's philosphical intention in the design of the syntax and the transformation rules of existential graphs is discussed. Then, in Chap. 3, Peirce's deeper understanding on the form and meaning of his graphs is investigated, and in Chap. 6, the same is done for Peirce's transformation rules. These four chapters offer a so-to-speak 'historical reconstruction' of Peirce's graphs.

Chaps. ?? and 3 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.

In Chaps. ??–??, the Alpha-part of EGs is mathematically elaborated. The syntax of these graphs is presented in Chap. ??, the semantics and calculus is presented in Chap. ??. In Chap. ??, 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 existential graphs, and Peirce separated existential graphs 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 Chap. **??**, 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 Chap. 3, the diagrams of Peirce's Beta graphs are first investigated to a large degree, before their syntax and semantics are formalized in Chaps. 4 and 5. 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 Chap. 6, before their formalization is provided in Chap. 7. The soundness of these rules can be shown similar to the soundness of their counterparts in Alpha. This is done in Chap. **??**. Similar to Alpha, I will provide translations between Beta graphs and formulas of first order logic. In Chap. **??**, the style of first order logic ($\mathcal{FO}$) which is used for this purpose is presented. In Chap. **??**, 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 Chap. **??**, 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 Chap. **??**, the results of the the preceeding chapters are transferred to the diagrammatic representations of EGs. Thus, this chapter concludes the program of formalizing Peirce's Beta graphs.

The aim and the structure of this treatise should be clear now. In the remainder of this section, some unusual features this treatise contains are 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. **??** or Def. **??**.

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 [**?**]. 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 passages 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 [**?**], book 4, passage 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 [**?**]: ‘*The misnamed Collected Papers [ldots] 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 Chap. **??**, 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 Chap. **??**), 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 [**?**]) 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 [**?**]: '*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 [**?**].

[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 [**?**]), when symbolic logic already had taken the vast precedence in formal logic. For this reason, although Peirce was convinced that the existential graphs are a much better approach to formal logic than any symbolic form of logic, existential graphs 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' instead of 'cut', too) which are represented by fine-drawn, closed, doublepoint-free curves.[4] 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 *judgment*). Thus,

it rains

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

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

[3] In [**?**], 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] Double-point free means that the line must not cross itself.

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

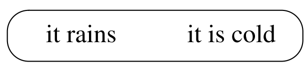
it is cold

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’.

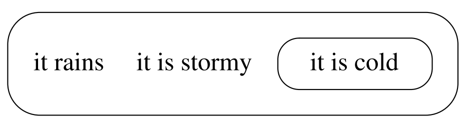
Encircling a graph by a cut is to negate it. For example, the graph

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

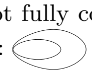
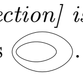


has the meaning ‘it is not true that it rains and that it is cold’, i.e. ‘it does not rain or it is not cold’ (the part ‘it is not true’ of this statement refers to the whole remainder of the statement, that is, the whole proposition ‘it rains and it is cold’ is denied.)

The space within a cut is called its *close* or *area*. Cuts may not overlap, intersect, or touch[5], 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 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

[5] 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 .

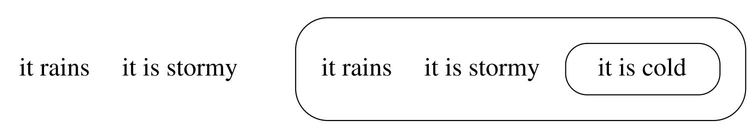
has three distinct areas: The area of the sheet of assertion contains the outer cut, the area of the outer cut contains the the propositions 'it rains' and 'it is stormy' and the inner cut, and the inner cut contains the proposition 'it is cold'. An area is *oddly enclosed* if it is enclosed by an odd 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 prosposition '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 first 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 Chap. **??**). 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:

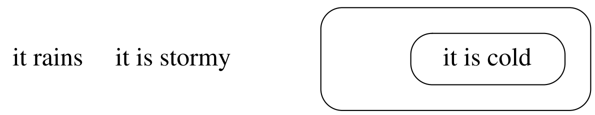
- Erasure: Any evenly enclosed subgraph[6] may be erased.
- Insertion: Any graph may be scribed on any oddly enclosed area.
- Iteration: If a subgraph $\mathfrak{G}$ occurs on the sheet of assertion or in a cut, then a copy of the graph may be scribed on the same or any nested area which does not belong 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:



[6] The technical term 'subgraph' will be precisely elaborated in Chap. **??**.

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:

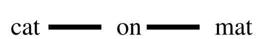
it is cold

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'.

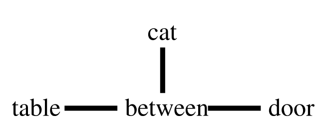
2.2 Beta

The step from the part Alpha of EGs to the part Beta 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:

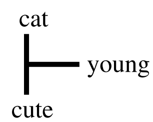


It contains two lines of identity, hence it denotes two (not necessarily different) objects. The first line of identity is attached to the unary predicate 'cat', hence the first object denotes a cat. Analogously the second line of identity denotes a mat. Both lines are attached to the dyadic predicate '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,



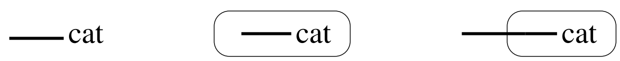
means ‘there is a cat between a table and a door’.

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



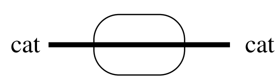
we have a ligature composed of three lines of identity, which meet in a so-called *branching point*. Still this ligature denotes one object: The meaning of the graph is ‘there is an object which is a cat, young and cute’, or ‘ there is a cute and young cat’ for short.

Ligatures may cross cuts (it will become clear in Chap. 3 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 Chap. 3). 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 part Beta 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).

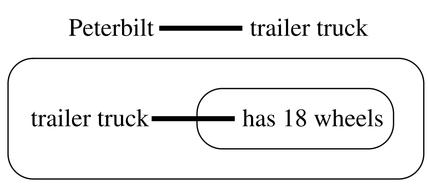
Essentially, the rules for Beta are extensions of the five rules for Alpha such that the Beta-rules encompass the properties of the lines of identity. The rules are now formulated as follows:

- 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: If a subgraph $\mathfrak{G}$ on the sheet of assertion or in a cut, then 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 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 inside of the inner cut.

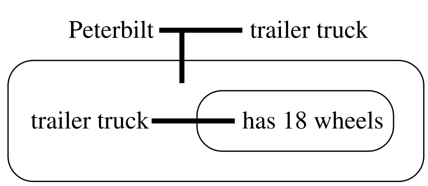
The precise understanding of these rules will be unfolded in Chap. 6. In this chapter, they will be illustrated with an example which is taken from [?]. This example is a proof of the following syllogism of type Darii:

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

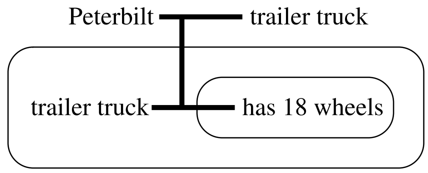
We start with an existential graph which encodes our premises:



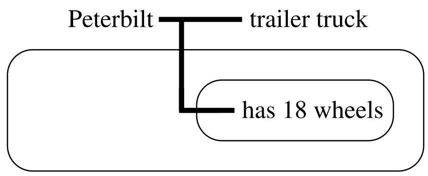
We use the rule of iteration to extend the outer line of identity into the cut:



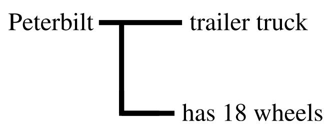
As the area of this cut is oddly enclosed, the insertion-rule allows us to join the loose end of the line of identity we have just iterated with the other line of identity:



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', hence with 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. 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):

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*;*

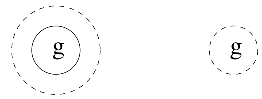
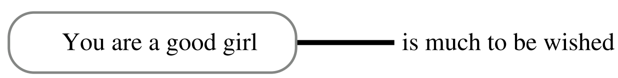


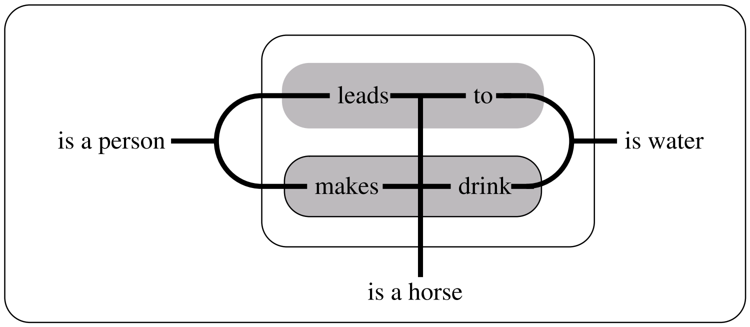
Fig. 2.1. Figs. 182 and 186 in 4.516

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 [**?**]). 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 [**?**], 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 context. For example, 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).



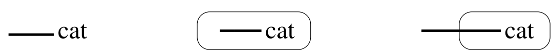
In this example we have two red (gray) ovals. One is purely red; it says that the enclosed graph is possible. The other one is a cut which is red inside, hence it says that the enclosed graph is *im*possible. As the three lines of identity start in the area of a scroll, they can be understood as *universally* 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. 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 [**?**]).

3

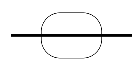
Getting Closer to Syntax and Semantics of Beta

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

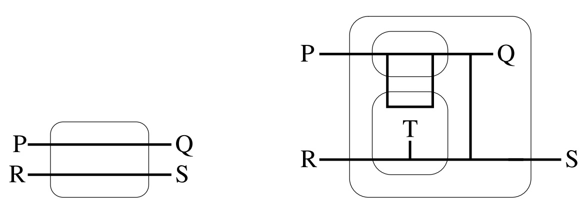


In all cases we have a heavy line (It will become clear soon why I do not write ‘line of identity’) which can be understood to denote one object. The meaning of the graphs are ‘there is a cat’, ‘it is not true that there is a cat’, and ‘there is something which is not a cat’, respectively.

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.

3.1 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:

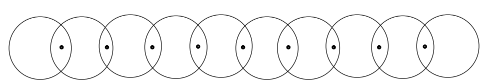
> A heavy line shall be considered as a continuum of contiguous dots; and since contiguous dots denote a single individual, such a line without any point of branching will signify the identity of the individuals denoted by its extremities.

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

$$\mathfrak{E}_1 := \;\rule{1.5em}{1pt}$$

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 ([**?**], 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' ([**?**]), 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



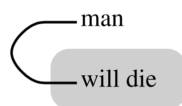
The identity spots are drawn as circles. The overlapping of these circles represents that the identity spots are coincident. Each point 'denotes an indesignate

[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' ([**?**]). We will come back later in some places to this relation to unfold some of its specific properties.

[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.

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.

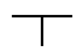
In the magnification, we have chosen a number of *ten* identity spots, but of course, this number is arbitrary. For a number of 10, the most explicit meaning of $\mathfrak{E}_1$ is: There are individuals $o_1, o_2, o_3, \ldots, o_{10}$, o_1 is (identical with) o_2, o_2 is (identical with) o_3, ..., and 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 ([**?**]): '*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. 3.3, 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. Furthermore,

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

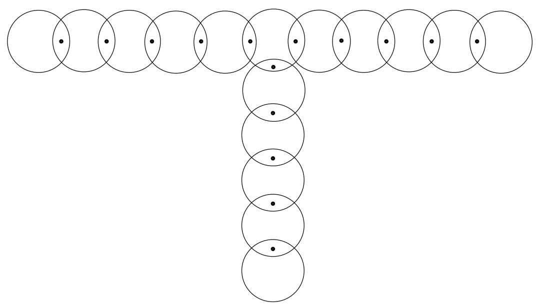
$\mathfrak{E}_2 :=$ 

In order to understand a branching of heavy lines, Peirce provides in 4.445 and in [**?**] similar examples, which are depicted in Fig. 3.1. In 4.445, he explains

Fig. 3.1. Fig. 79 of 4.445 and an example the of the tutorial [**?**]

the left graph of Fig. 3.1 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 [**?**] of 1909 he writes that the right graph of Fig. 3.1 '*is a graph instance composed of instances of three indivisible graphs which assert 'there is a male', '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]

[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 relation, 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

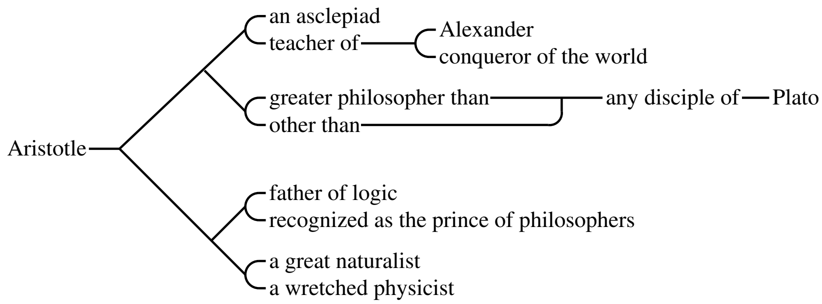


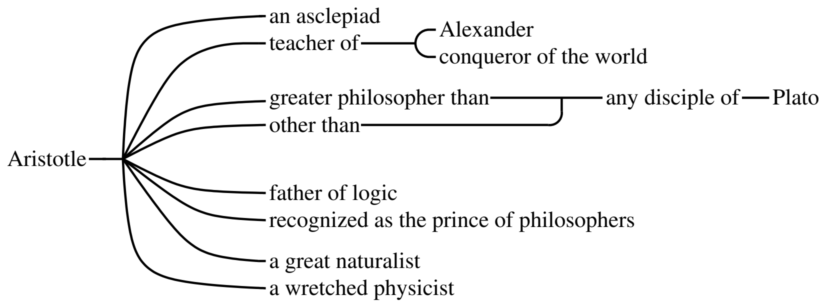
Fig. 3.2. Fig. 80 of 4.445

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 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 19 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 [**?**] have branching points with more than three branches. existential graphs only

An 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 diffently: Allowing only branching points three branches (which is a syntactical restriction) does not lead to a loss of expressiveness.

Concerning branching points, this treatise follows a different path than Peirce. 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 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. 3.1 could be transformed into the following graph, having such a branching point.

(namely 'Aristotle', 'Alexander' and 'Plato'). Secondly, he implicitly brings in an *universal* quantification by the use of a relation '*any* disciple of'.



Comment: 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 is a famous theorem of Peirce. To put the latter slighty more precisely: In a given algebra of relations, where all relations are monadic or binary, the teridentity (which is syntactically reflected by branching points with three branches) cannot be constructed from these relations. This is a *semantical* theorem which is by no means trivial. For a mathematical elaboration of Peirce's reduction thesis, see [**?**] and [**?**].

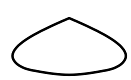
Peirce sometimes uses the term 'line of identity' for a linked structure of heavy lines. In his Cambridge lectures of 1898 ([**?**]) we find the phrase: '*Now as long as there is but one such line of identity, whether it branches or not [...].*', and even in the collected papers, 4.446, we find '*Convention 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.

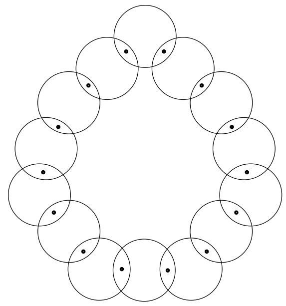
A single LoI is understood to be a ligature as well, but a ligature, as soon as it has branches or when it crosses a cut, is not a LoI. As Peirce writes in 4.499: '*Let us, then, call a series of lines of identity abutting upon one another at seps, a ligature; and we may extend the meaning of the word 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 [**?**]: '*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 [**?**] 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 [**?**] he describes a graph similar to the right graph of Fig.3.3 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. $\mathfrak{E}_2$:

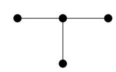
 for $\mathfrak{E}_1$, and  for $\mathfrak{E}_2$.

Fig. 3.3. 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:

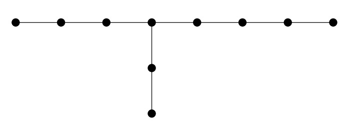
 for $\mathfrak{E}_1$, and  for $\mathfrak{E}_2$.

Fig. 3.4. 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 $\mathfrak{E}_3$.

3.2 Predicates

In Fig. 3.1, 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. 3.1:

Fig. 3.5. Two subgraphs of Fig. 3.1

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.

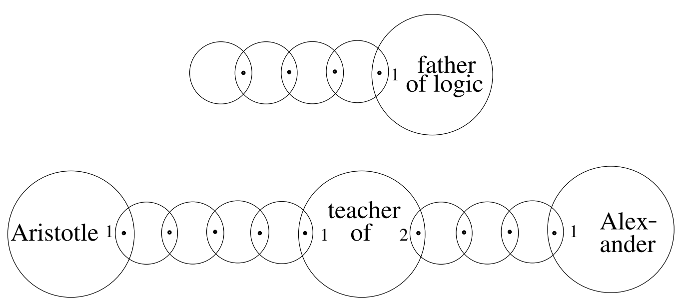
Peirce wanted to develop a 'logic of relatives' (i.e., relations). In fact, in his calculus for EGs, Peirce did not provide any rules for the treatment of object

[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. 3.4 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 (a) Aristotle, the second object is (a) 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 [**?**]: '*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. 3.2 can be magnified as follows:



Of course, the order of the hooks is crucial. For example, considering the right graph of Fig. 3.2, it makes a difference whether Aristotle was the teacher of

[6] As the right graph of Fig. 3.2 show, it is desirable to have object names. In Chapter. **??**, 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 occurence 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. 3.2 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:

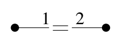
The edge is represented by writing the name of its label and drawing lines from this name to the dot which represent the incident vertex.

[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/Mugnier for CGs (see [?, ?]), or by Pollandt in [?] and by Hereth Correia and Pöschel in [?] for relation graphs. I consider an encoding with edges instead of vertices more practical, but somehow, this decision depends on matter of (mathematical) taste.

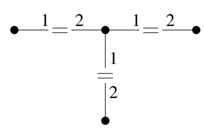
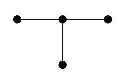
The right graph of Fig. 3.2 is encoded as follows:

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.

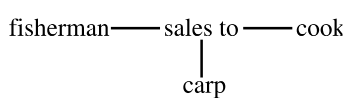
As said above: Identity is a special relation, and the former 'identity edges' are formalized as as directed edges which are labeled with a name for identity, like '='. That is, we have the following formalizations of $\mathfrak{E}_1$ and $\mathfrak{E}_2$:

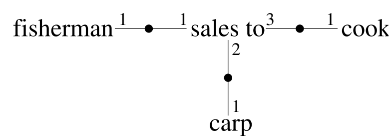
 instead of

and

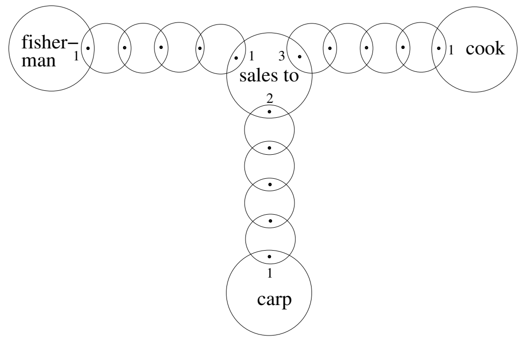
 instead of  .

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:

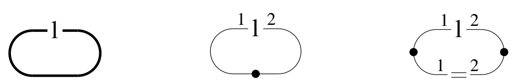




This graph can be magnified as follows:



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.



3.3 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]

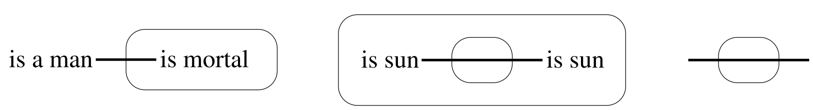


Fig. 3.6. Fig. 67 and 68 of 4.407, and $\mathfrak{E}_{twothings}$

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.

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

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 [**?**] as well) as follows:

> 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 Peice 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. 3.3.

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,*

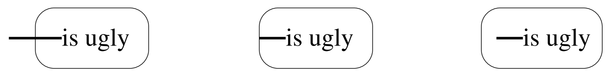
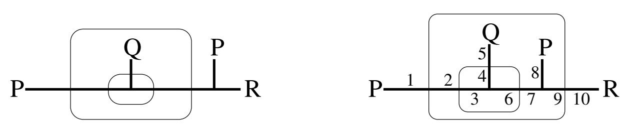


Fig. 3.7. Fig. 154, 155 and 156 of 4.502

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.

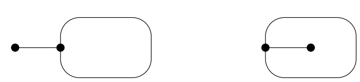
How shall the heavy line of Peirce's Fig. 154 be understood? In 4.416, Peirce says: '*Two lines of identity, one outside a cut and the other on the area of the same cut, may have each an extremity at the same point on the cut.*' This explains how the heavy line of Peirce's Fig. 154 can syntactically 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.

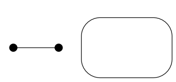
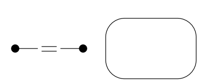
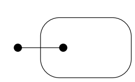
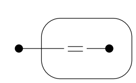
A LoI expresses that the objects denoted by its two extremities are identical. The LoIs l_1 and l_2 have a point in common, namely the point on the cut. Let us denote the object denoted by the left endpoint of l_1 by o_1, the object denoted by the common endpoint of l_1 and l_2 on the cut by o_2, and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

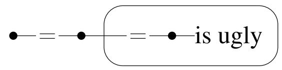
Peirce's conventions are very helpful for our formalization of existential graphs. We have to add cuts to our formalization, but there is no need to formalize graphs where it is possible to express that a vertex is *on* a cut: It is sufficient to consider structures where the vertex is *inside* or *outside*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



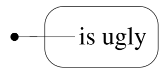
Due to Conventions 8 and 9, we will consider the following two graphs instead:

 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:

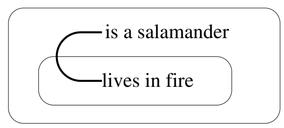


It will turn out in the next chapter that this graph can be transformed to



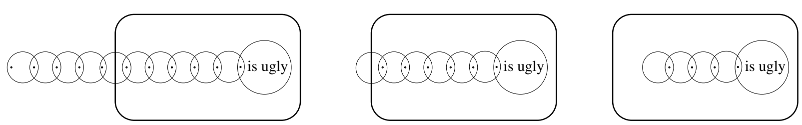
For this graph, it is easy to see that its meaning is 'there is something which is not ugly'.

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.



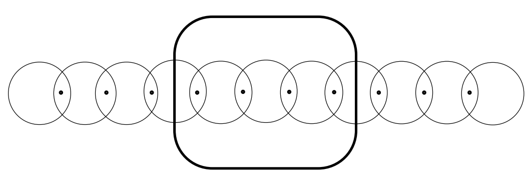
Particularly, he obtains: '*In order, therefore, to legitimate our interpretation of Fig. 83, we must agree that a line of identity crossing a sep simply asserts the identity of the individual denoted by its outer part and the individual denoted by its inner part.*' Then, he comes to Conventions 8. and 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. 3.3:

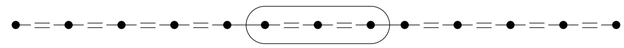


Again, we see that the magnifications correspond to the ongoing formalization of Peirce's graphs.

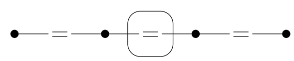
Consider now the graph $\mathfrak{E}_{twothings}$. The meaning of $\mathfrak{E}_{twothings}$ is 'there are at least two things'. We now have the ability to analyze why this is the correct interpretation of $\mathfrak{E}_{twothings}$. A possible magnification of the graph is



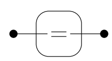
Possible formalizations of $\mathfrak{E}_{twothings}$ can be obtained from the possible magnifications of $\mathfrak{E}_{twothings}$ or from Peirce's conventions. The formalization obtained from the given magnification is depicted below. Thus, the following is a possible formalization of the graph:



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:

 .

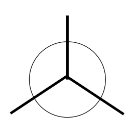
Again, we will see in the next chapter that these formalizations can be simplified. The following graph contains only two vertices, which are placed on the sheet of assertion, and one identity-edge, which is placed in the cut and which connects the vertices.



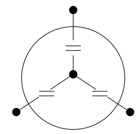
As the identity-edge is placed inside the cut, the identity of the objects denoted by the vertices is denied. I.e., this graph has in fact the meaning 'there are two objects o_1 and o_2 such that it is not true that o_1 and o_2 are identical', that is, there are at least two things. This is probably 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:

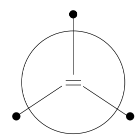


Due to our discussion, a possible formalization of this graph is



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 = for teridentity (identity of three objects), the graph could even simpler be formalized as follows:

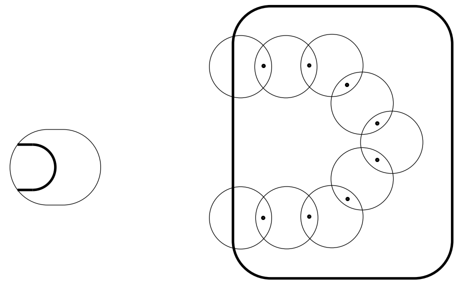


This is even better readable. In the formal definition of EGs, we will use only the usual, dyadic identity, but in Chapter **??**, we will come back to this idea in order to obtain an improved reading ‘algorithm’ for EGs.

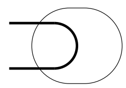
3.4 Border cases: LoI touching or crossing on a Cut

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 Chap. 6, where the rules of Peirce will be discussed, on page 71 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:

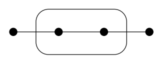


If we consider the points which terminate on the cut to lie outside the cut, we obtain the following diagram



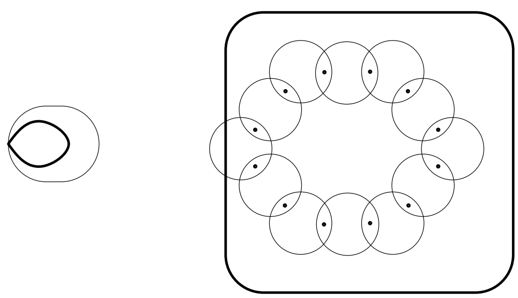
,

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,

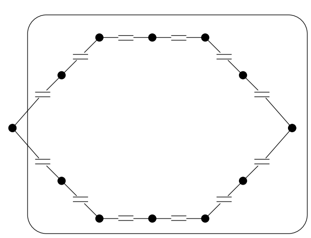


is a possible translation of this graph, which again yields that this graph is equivalent to $\mathfrak{E}_{twothings}$.

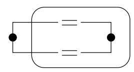
A similar example is the following graph and its magnification:



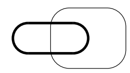
Contrary to the last example, we have *one* identity spot instead of two (which is an endpoint of a LoI). The magnification makes clear that



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

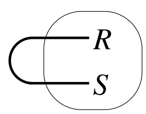
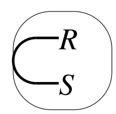
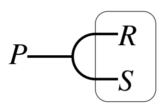
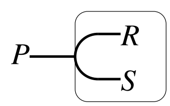


From the possible magnifications, thus the possible formalizations, we conclude that the following Peircean graph is an appropriate substitute for the starting graph:

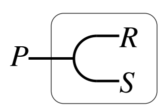


(This is the graph of Fig. 18, page 54 in the book of Roberts. Its meaning is 'there is a thing which is not identical with itself', i.e., this EG is contradictory.)

Analogous considerations show that it is sufficient

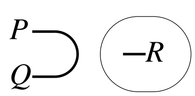
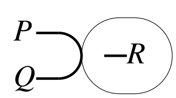
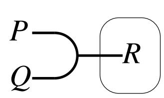
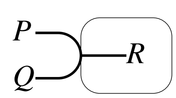
- to consider  instead of  or
- to consider  instead of  .

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

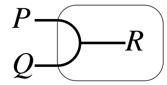


This will be elaborated further in Chap. 6 and Chap. **??**.

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  .

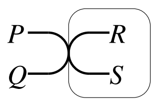
The meaning of the last two graphs is 'there is an object which has properties P and Q, but not R'. It should be noted that the last two graphs entail, but are *not* semantically equivalent to the graph below.

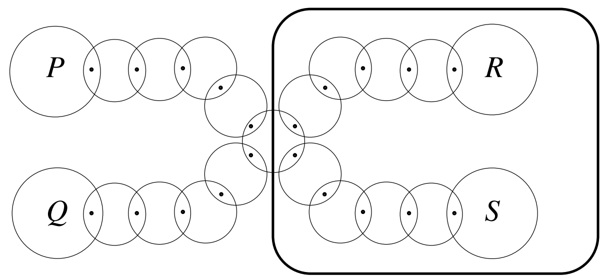


.

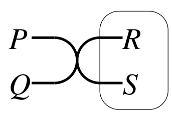
(The meaning of this graph is: 'There is an object o_1 with property P and an object o_2 with property Q, such that either o_1 and o_2 are not identical, or o_1 and o_2 are identical, but the property R does not hold for the (identical objects) o_1 and o_2').

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

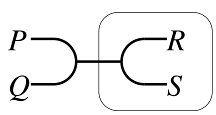




Again the magnification helps to see that the following graph is the appropriate, non-degenerated substitute:

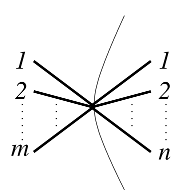


The following graph is semantically equivalent, too:

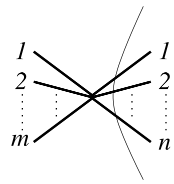


The meaning of these graphs is 'there is an object which has property P and Q, but is has not both properties R and S'.

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.

4

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. 3.1). For this reason, the definition of formal existential graphs is done in two steps: First, formal existential graph *candidates* are defined. An existential graph candidate (EGC) 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 EGC which can be understood as formalization of this EG. The *class* of all these EGCs will be the formalization of the EG.

The underlying structure of EGCs are so-called called RELATIONAL GRAPHS. An EGC is a relational graphs whose edges are additionally labeled with predicate names. In the part 'Extending the System' (*see* Cap. **??** ff.), the expressivity of EGCs is extended by adding object names or so-called query vertices (which can compared to *free* variables in $\mathcal{FO}$), and these extensions are obtained from EGCs by generalizing the labeling. Similarly, the class of *concept graphs with cuts*, which is investigated in [**?**], or the class of *query graphs with cuts* (see [**?**]), is based on relational graph with cuts. For this reason, we first define in Sect. 4.1 relational graphs and investigate their structure. In Sect. 4.2, the labeling of the edges with relation names is added to these graphs: The resulting graphs are existential graph candidates. `Labels auch fuer Knoten!!` Then, in Sect. 4.3, some further syntactical notations for EGCs like *subgraph* etc. are introduced. Finally, in Sect. 4.4, we define formal existential graphs as sets of EGCs which can be mutually transformed into each other by a set of very simple rules.

4.1 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 4.1 (Relational Graphs with Cuts).

A structure $(V, E, \nu, \top, Cut, area)$ is called a RELATIONAL GRAPH WITH CUTS *if*

- *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 $\nu(e) = (v_1, \ldots, v_k)$ 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 $e = (v_1, \ldots, v_k)$ instead of $\nu(e) = (v_1, \ldots, v_k)$. 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 $V_e := \{v \in V \mid \exists i.\nu(e)\big|_i = v\}$. 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: $c = cut(x)$.

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 introduce an order on the elements of an relational graph.

[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 4.2 (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

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

and set $x \leq y :\Longleftrightarrow \exists n \in \mathbb{N}_0 . \beta(x) \in area^n(\beta(y))$ for $x, y \in V \cup E \cup Cut \cup \{\top\}$.
We define $x < y :\Longleftrightarrow x \leq y \wedge y \not\leq x$ and $x \lneq y :\Longleftrightarrow x \leq y \wedge y \neq x$.
For a context $c \in Cut \cup \{\top\}$, we set furthermore

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

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.*

We have that x is enclosed by a cut c if and only if $x \lneq c$, and we obtain the following corollary:

Lemma 4.3 (Order Ideals Generated by Contexts).
Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$ be a relational graph with cuts and $c \in Cut \cup \{\top\}$. 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 k is a vertex or an edge, we have

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

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

If we have two cut-lines in the graphical representations of an EG, 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 EGCs as well, which is the proposition of the next lemma.

Lemma 4.4 (Relations between Order Ideals).

For a relational graph with cuts $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$ *and two contexts* $c_1 \neq c_2$*, exactly one of the following conditions holds:*

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

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. 4.1 and to Lem. 4.3. 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. 4.3 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) for *area* of Def. 4.1.
- $m = 1, n > 1$: Let $c_2{}' \in area^{n-1}(c_2)$ such that $x \in area(c_2{}')$. From a) for *area* of Def. 4.1 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 $m + n$. □

Now we get the following corollary:

Corollary 4.5 (≤ 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. 4.1, 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.

Definition 4.6 (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. This is captured by the following definition.

Definition 4.7 (Dominating Nodes).

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

4.2 Existential Graph Candidates

Existential graph candidates (EGCs) 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 EGCs. 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 EGCs on the basis of relational graphs.

Definition 4.8 (Alphabet).

An ALPHABET *is a pair $(\mathcal{R}, ar : \mathcal{R} \to \mathbb{N}_0)$. 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 4.9 (Existential Graph Candidates).

An EXISTENTIAL GRAPH CANDIDATE (EGC) OVER THE ALPHABET *$\mathcal{A}$ 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,*
- *$\kappa : E \to \mathcal{R}$ is a mapping such that $|e| = ar(\kappa(e))$ for each $e \in E$.*

[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 '$=$'.

For the set E of edges, let $E^{id} := \{e \in E \mid \kappa(e) = \doteq\}$ 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 $cut(e) = cut(e|_1)$ or $cut(e) = cut(e|_2)$, then e is called* STRICT IDENTITY-EDGE.

The vertices, edges and cuts of an EGC will be called the ELEMENTS *of the EGC. The system of all EGCs over $\mathcal{A}$ will be denoted by $\mathcal{EGC}^{\mathcal{A}}$.*

In the following, we will introduce mathematical definitions for the terms *ligature* and *hook*. 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 EGC $(V, E, \nu, \top, Cut, area, \kappa)$, our aim is to describe ligatures as subgraphs of $(V, E^{id}, \nu|_{E^{id}})$, 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 egdes 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 $\nu' = \nu|_{E'}$. 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 alread said, for a given EGC $(V, E, \nu, \top, Cut, area, \kappa)$, our aim it to introduce ligatures as subgraphs of $(V, E^{id}, \nu|_{E^{id}})$. Stricly 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 ommit the mapping ν, but we agree that ν is implicitely given. That is, if we speak about a subgraph (W, F) of (V, E^{id}), it is meant that we mean the subgraph $(W, F, \nu|_F)$ of $(V, E^{id}, \nu|_{E^{id}})$. Now it is easy to define ligatures as follows:

Definition 4.10 (Ligature).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be an EGC. 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 EGC 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 4.11 (Hook).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be an EGC. 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* $e|_i = v$*, then we say that* THE VERTEX v IS ATTACHED TO THE HOOK (e, i)*.*

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *is an EGC,* v *be an 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* $cut(v') \geq cut(e)$*. 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 $\mathfrak{G}'$ *is obtained from* $\mathfrak{G}$ *by* REPLACING v BY v' ON THE HOOK (e, i)*.*

In order to work with EGCs, their mathematical representations are too clumsy and too difficult to handle. Hence one may prefer graphical representations OF EGCs. We have to explain how EGCs are drawn. This shall be done now.

The different elements of an EGC 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 EGC form a tree (see Lemma 4.5). 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 $cut(v)$. 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 $cut(e)$. 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 $v_i > c \geq cut(e|_2)$, 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.

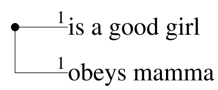
There may be graphs such that its edge-lines cannot be drawn without their crossing one another (i.e., they are not planar). For this reason, the intersection of edge-lines is allowed. But the intersection of edge-lines should be avoided, if it possible.

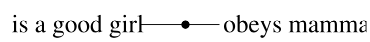
To illustrate these agreements, I provide some examples. These examples are EGCs over an alphbet without constants, thus the mapping ρ is omitted. The first three examples are adopted from Peirce's Cambridge Lectures, Lecture three.

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

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

Below you find two representations for this EGC. 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 EGC 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 EGC can be represented as follows:

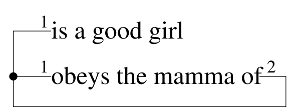
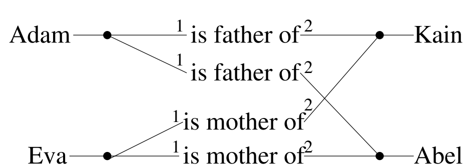


Fig. 4.1. One representation for $\mathfrak{G}_2$

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

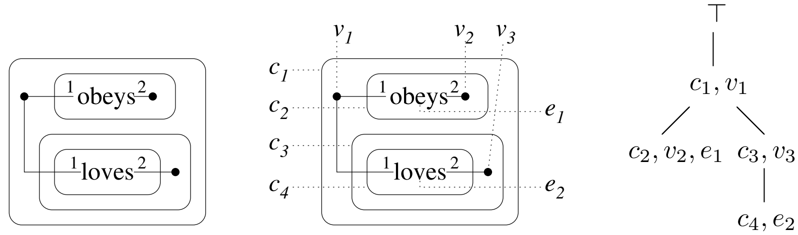


It is well known from graph-theory that not every graph has a planar representation. For this graph, a planar representation would be possible, but the given representation with crossing edge-lines shows nicely the symmetry of the EGC. Crossing edge-lines cause no troubles in the reading of the diagram,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

The next example is a more complex EGC with cuts.

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

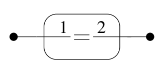
Below, the left diagram is a possible representation of $\mathfrak{G}_3$. In the right diagram on the right, I have sketched furthermore assignments of the elements (the vertices, edges, and cuts) of the EGC to the graphical elements of the diagram. Finally, on the right, the order $\leq$ for $\mathfrak{G}_3$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



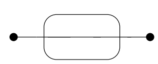
The next example shall explain the two different conventions for identity edges. We had already the first examples in the last chapter on page 29. Consider the following EGC:

$$\mathfrak{G}_4 := (\{v_1, v_2\},\ \{e\},\ \{(e, (v_1, v_2))\},\ \top,\ \{c\}, \{(\top, \{v_1, v_2\}), (c, \{e\})\}, \{(e, \doteq)\})$$

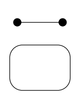
It is a formalization of the the well-known graph claiming there exists at least two things. Due to our normal convention for drawing edges, it is graphically represented as follows:



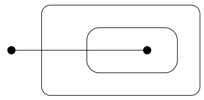
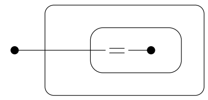
The second convention says that the sign '$\doteq$' may be replaced by a line. This yields:



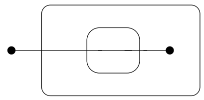
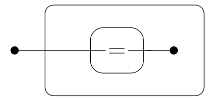
Now the whole identity-edge is represented by a simple line. But as the sign '$\doteq$' we had replaced was placed inside the cut-line, it is important that the new, simple line goes through the inner area of the cut-line as well. The following diagram, in which we have a simple line connecting the vertex-spots 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 

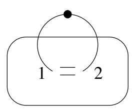
.

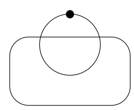
None of the identity-edges can be placed in a different cut, because we consider only graphs with dominating nodes.

This holds even for identity-edges e with $e|_1 = e|_2$. Their edge-lines appear to be closed. For example, consider

$$\mathfrak{G}_5 := (\{v\},\ \{e\},\ \{(e,(v,v))\},\ \top,\ \{c\},\ \{(\top,\{v\}),(c,\{e\})\},\ \{(e,\dot{=})\})$$

Due to the conventions, this graph may be represented as follows:



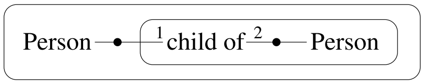


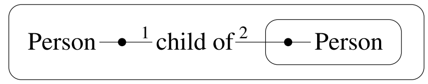
Although the edge-line cannot be mistaken with the cut-line, the right diagram lacks readability, thus one should prefer the left diagram instead.

From the examples it should be clear that each EGC can be represented as a diagram. Similar to Alpha, it shall shortly be investigated which kind of diagrams occur as diagrams of an EGC. 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 EGC in which identity egdes 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 $1, \dots, n$, 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 EGCs are relation-graphs with domination nodes. For example, the left diagram is a diagram of an EGC, while the right graph is not:

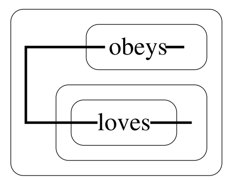




We already used the condition of dominating nodes to provide the second convention for drawing identity edges. So, even if identity edges are drawn due the second condition, from each properly drawn diagram (that is, it has to satisfy the above given conditions) we can reconstruct up to isomorphism (which will we be defined canonically in Def. 4.13) and the orientation of those identity-edges which are drawn as simple lines) the underlying EGC. Thus, from now on, we can use the diagrammatic representation of EGCs in mathematical proofs.

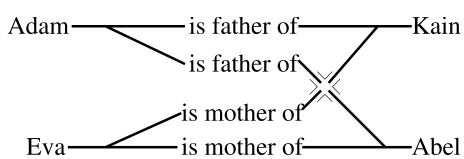
Next, the correspondence between the diagrams of Peirce's Beta graphs and the diagrams of EGCs shall be investigated.

If an diagram of an EGC 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 EGCs 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



[3] This is a general problem of mathematical graphs which is extensively investigated in graph theory.

4.3 Further Notations for Existential Graph Candidates

Similar to Alpha, we have to define *subgraphs* of EGCs. 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]

Definition 4.12 (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* $c' \in Cut'$ *,*
- $cut(x) \in Cut' \cup \{\top'\}$ *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 EGC. 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 labelling, i.e. if* $\kappa' = \kappa\big|_{E'}$ *holds.*

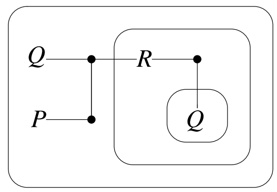
In both cases (relational graph with cuts and EGCs), we write $\mathfrak{G}' \subseteq \mathfrak{G}$ *and* $area^{-1}(\mathfrak{G}') = \top'$ *resp.* $cut(\mathfrak{G}') = \top'$*.*

Similar to Alpha (see Lem. **??**), 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 $Cut' \cup \{\top'\}$).

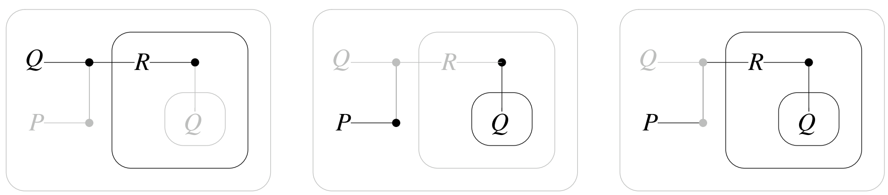
[4] In this definition, subgraphs are first defined for relational graph with cuts, then for EGCs. The reason is that we will slightly extend the syntax for EGCs in Def. **??** by adding a second labelling function. Strictly speaking, the following definition of subgraphs cannot be applied to these graphs, but if we define subgraphs first for relational graph with cuts, it should be clear how this definition should be modified for the extended EGCs of Def. **??**.

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 a context c if and only iff $\top' \leq c$. Moreover, we can apply Def. 4.6 to subgraphs as well. Hence we we distinguish whether a subgraph is evenly or oddly enclosed.

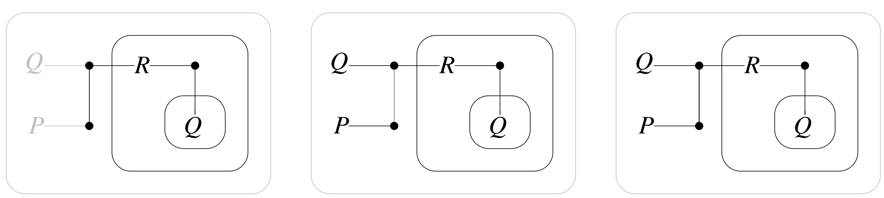
To get an impression of subgraphs of EGC, an example will be helpful. Consider the following graph:



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 Chapt. **??**, it is investigated how subgraphs can be diagrammtically depicted.

Analogously to Alpha, isomorphisms between EGCs 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 EGCs.

Definition 4.13 (Isomorphism).

Let $\mathfrak{G} := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$, $i = 1, 2$, *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 EGCs. 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 labelling, 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 4.14 (Partial Isomorphism).

For $i = 1, 2$, *let* $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ *be two relational graph with cuts, and let* $c_i \in Cut_i \cup \{\top_i\}$ *contexts. Let*

1. $E_i' := \{e \in E_i \mid e \not\leq c_i\}$,
2. $V_i' := \{v \in V_i \mid v \not\leq c_i\}$, *and*
3. $Cut_i{}' := \{d \in Cut_i \cup \{\top_i\} \mid d \not< c_i\}$

for $i = 1, 2$. *Let* $\mathfrak{G}_i'$ *be the restriction of* $\mathfrak{G}_i$ *to these sets, i.e.,* $\mathfrak{G}_i' := (V_i', E_i', \nu|_{E_i'}, \top_i, Cut_i', area_i', \kappa_i')$. *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* ISOMORPHISM FROM $\mathfrak{G}_1$ TO $\mathfrak{G}_2$ EXCEPT FOR THE CUTS $c_1 \in Cut_1 \cup \{\top_1\}$ AND $c_2 \in Cut_2 \cup \{\top_2\}$.

Analogously to the last definition, partial isomorphismns between EGCs are partial isomorphismns between the underlying relational graph with cuts which respect κ.

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 $e_i \in area(c_i) \cap E_i$, $i = 1, 2$).

Finally, we have to define juxtapositions for EGCs.

Definition 4.15 (Juxtaposition).

Let $n \in \mathbb{N}_0$ and $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i)$ be a relational graph with cuts for $i = 1, \ldots, n$. 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 RGC 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 $i = 1, \ldots, n$.*

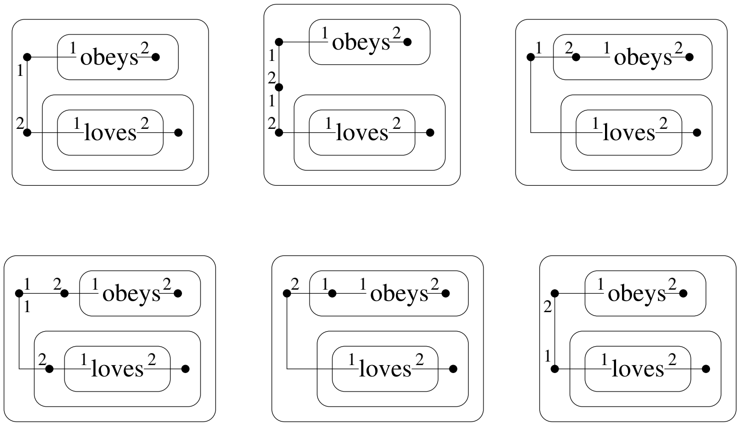
In the graphical notation, the juxtaposition of graphs $\mathfrak{G}_i$ (relational graphs with cuts or EGCs) is simply noted by writing the graphs next to each other, i.e. we write:

$$\mathfrak{G}_1 \ \mathfrak{G}_2 \ \ldots \ \mathfrak{G}_n \qquad .$$

Again, similar to Alpha, the juxtaposition of an empty set of EGCs yields the empty graph, i.e., $(\emptyset, \emptyset, \emptyset, \top, \emptyset, \emptyset, \emptyset)$.

4.4 Formal Existential Graphs

In Sect. 4.2, I have argued that EGCs can be represented by diagrams, and that we can reconstruct an EGC from a diagram up to isomorphism. Moreover, we have discussed how the diagrams of EGCs 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 EGCs may yield the same Peirce diagram as well. For example, the next EGCs yield the same Peircean diagram as $\mathfrak{G}_3$ from Sect. 4.2:



Neither the diagrams of EGCs, 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 EGCs 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 EGC 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 EGC, 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 an vertex which is attached to a hook (e, i). Furthermore let c be a context with $cut(v) \geq c \geq cut(e)$. 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 $x \notin V \dot\cup E \dot\cup Cut \dot\cup \{\top\}$.

Definition 4.16 (Transformation Rules for Ligatures).

The transformation rules for ligatures for EGCs over the alphabet $\mathcal{R}$ are:

- **isomorphism**

  Let $\mathfrak{G}, \mathfrak{G}'$ be EGCs 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 EGC. 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

  $$\nu'(f) = \nu(f) \textit{ for } f \neq e \textit{ , and } \nu'(e) = (v_2, v_1)$$

  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 EGC. Let $v \in V$ be an 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 $cut(v) \geq c \geq cut(e)$. 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.

Now we can formally define when two EGCs are considered to be formalizations of the same Peircean diagram.

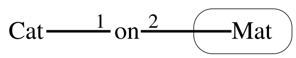
Definition 4.17 (Existential Graphs).

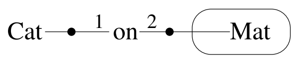
Let $\mathfrak{G}_a, \mathfrak{G}_b \in EGC$ be EGCs over a given alphabet $\mathcal{A}$. We set $\mathfrak{G}_1 \sim \mathfrak{G}_2$, if $\mathfrak{G}_2$ can be obtained from $\mathfrak{G}_1$ with the four rules above (i.e., if there is a finite sequence $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ 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').

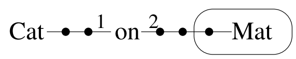
Each class $[\mathfrak{G}]_\sim$ is called an (FORMAL) EXISTENTIAL GRAPH *over $\mathcal{A}$. Formal existential graphs will usually be denoted by the letter $\mathfrak{E}$.*

To provide a simple example, a Peircean Beta graph and two elements (that is: EGCs) of the corresponding formal Beta graph are depicted.





$(\{v_1, v_2, v_3\},$
$\{e_1, e_2\},$
$\{(e_1, (v_1, v_2)), (e_2, (v_1, v_3))\},$
$\top, \{c_1, c_2, c_3, c_4\},$
$\{(\top, \{\}), (c_1, \{v_1, c_2, c_3\})\},$
$\{(e_1, \text{Cat}), (e_2, \text{loves}), (e_3, \text{Mat})\})$



$(\{v_1, v_2, v_3, v_4, v_5\},$
$\{e_1, e_2, e_3, e_4, e_5, e_6\},$
$\{(e_1, (v_1)), (e_2, (v_2, v_3)),$
$(e_3, (v_5)), (e_4, (v_1, v_3))\},$
$(e_5, (v_3, v_4)), (e_6, (v_4, v_5))\},$
$\top, \{c\},$
$\{(\top, \{v_1, v_2, v_3, v_4, e_1, e_2\}), (c, \{v_5, e_3\})\},$
$\{(e_1, \text{Cat}), (e_2, \text{loves}), (e_3, \text{Mat})\})$
$\{(e_3, \doteq), (e_4, \doteq), (e_5, \doteq)\})$

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 EGCs are already mathematical structures), thus it is not used for this purpose.

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 EGCs 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 EGCs. 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 EGCs to definitions for formal EGs. For example, we can say that $\mathfrak{E}_0$ is the juxtaposition of $\mathfrak{E}_1, \ldots, \mathfrak{E}_n$, if we have EGCs $\mathfrak{G}_0, \mathfrak{G}_1, \ldots, \mathfrak{G}_n$ 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 EGCs have the same meaning.

5

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 [**?**]) or by providing a mapping -let us call it Φ– of existential graphs to $\mathcal{FO}$-formulas (e.g. [**?**], [**?**], [**?**], [**?**]). 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 Chap. **??** is provided. To the best of my knowlegde, this is the first approach to equip Peirce's Beta graphs with a direct semantics.

In the first section, it will be shown how EGCs are evaluated in relational structures. It should be expected that equivalent EGCs have the same meaning (i.e., their evaluation in arbitrary structures yield always the same result). This will be shown in the second section.

5.1 Semantics for Existential Graph Candidates

EGCs 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 5.1 (Relational Structures).

A RELATIONAL STRUCTURE OR 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 EGC 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. **??**) – by valuations.

Definition 5.2 (Partial and Total Valuations).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGC 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 EGC is valid in a relational structure over $\mathcal{A}$. The semantics we will provide is the formalization of Peirce's endoporeutic method to evaluate existential graphs. He read and evaluated existential graphs 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 on the graph $\mathfrak{G}$ from Fig. 4.1 on p. 49.

We start the evaluation of $\mathfrak{G}$ on the sheet of assertion $\top$. As only the cut c_1 is directly enclosed by $\top$, $\mathfrak{G}$ is true if the part of $\mathfrak{G}$ which is enclosed by c_1 is false. As c_1 contains the vertex v_1 and the cut c_2, we now have the following: $\mathfrak{G}$ is true if it is not true that there exists an object o_1 such that o_1 is a cat and the proposition which is enclosed by c_2 is false. Now we have to evaluate the area of c_2. Intuitively spoken, the area of c_2 becomes true if there is an object that is a cute animal and identical to o_1. So, during this step of the evaluation, we refer to the object o_1. (That is why the endoporeutic method proceeds *inwardly*: We cannot evaluate the inner cut c_2 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_2.) Hence $\mathfrak{G}$ is true if there is no cat such that there is no other object which is identical to the cat and which is a cute animal. In simpler words: $\mathfrak{G}$ is true if there is no cat which is not a cute animal, i.e., if every cat is a cute animal.

Definition 5.3 (Endoporeutic Evaluation of Graphs).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an existential graph candidate 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 EGCs and if $\mathfrak{G}$ is an EGC 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.

5.2 Semantics for Existential Graphs

In the next section, a calculus for EGCs will be presented. This calculus will have derivation rules, but recall that we already have four transformation rules on EGCs, 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 Chap. **??**.

We start with a simple definition.

Definition 5.4 (Partial Isomorphism Applied to Valuations).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be EGCs 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 $\mathfrak{G}$ such that $dom(ref) \cap \{v \in V \mid cut(v) \leq c\} = \emptyset$. 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 5.5 (Main Thm. for Soundness, Implication Version).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be two EGCs 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. 5.4, $f(ref)$ 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. 5.3, 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

$$\mathcal{M} \models_{endo} \mathfrak{G}_a \stackrel{\text{Def.} \models}{\Longleftrightarrow} \mathcal{M} \models_{endo} \mathfrak{G}_a[\top_a, \emptyset]$$

$$\stackrel{\mathrm{P}(\top_a)}{\Longrightarrow} \mathcal{M} \models_{endo} \mathfrak{G}_b[\top_b, \emptyset]$$
$$\stackrel{\mathrm{Def.}\models}{\Longleftrightarrow} \mathcal{M} \models_{endo} \mathfrak{G}_b$$

which yields the particular property for $\top$. □

This will be the main theorem to prove the soundness of those rules of the (forthcoming) calculus which weaken the 'informational content' of an EGC (e.g. erasure and insertion). The transformation rules of Def. 4.16 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 5.6 (Main Thm. for Soundness, Equivalence Version).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ be two graphs 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 Thm. 5.5. □

In the following, we will show that two EGCs $\mathfrak{G}, \mathfrak{G}'$ 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. 5.1 and 5.3, 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 5.7 (The Transformation Rules for Ligatures are Sound).

If $\mathfrak{G}$ and $\mathfrak{G}'$ are EGCs, $\mathcal{M} := (U, I)$ 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. 4.16, i.e., we have a vertex $v \in V$, an edge $e \in E$ be an edge with $e|_i = w$ for an $i \in \mathbb{N}$, and a context c with $cut(v) \geq c \geq cut(e)$ 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 $\{d \in Cut \cup \{\top\} \,|\, d \not\leq c\}$. 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 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 Thm. 5.5, so we conclude $\mathcal{M} \models_{endo} \mathfrak{G} \iff \mathcal{M} \models_{endo} \mathfrak{G}'$. □

As we now know that all transformation rules respect the relation $\models$, we immediately obtain the following theorem:

Theorem 5.8 ($\sim$ Preserves Evaluations).

Let $\mathfrak{G}_1, \mathfrak{G}_2$ be two EGCs with $\mathfrak{G}_1 \sim \mathfrak{G}_2$, and let $\mathcal{M}$ be a relational structure. Then we have: $\mathcal{M} \models \mathfrak{G}_1 \quad \iff \quad \mathcal{M} \models \mathfrak{G}_2$.

Now we can carry over the semantics for EGCs to semantics for formal EGs:

Definition 5.9 (Semantics for Existential Graphs).

Let $\mathfrak{E}$ be an EG, $\mathfrak{G}$ be an representing EGC for $\mathfrak{E}$ (i.e., $\mathfrak{E} = [\mathfrak{G}]_\sim$) and $\mathcal{M}$ be a relational structure. We set: $\mathcal{M} \models \mathfrak{E} \quad :\iff \quad \mathcal{M} \models \mathfrak{G}$.

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 EGCs and classes of EGCs.

6

Getting Closer to the Calculus for Beta

In Chap. 3, we first have extensively investigated Peirce's understanding of the 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. **??** in the Alpha-part, we will deeply discuss Peirce's rules for existential graphs, before the formal definition for the calculus is provided in Chap. 7.

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 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 se 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 EGCs 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.

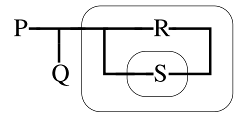
6.1 Erasure and Insertion

We have already discussed in Sec. **??** 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.*'

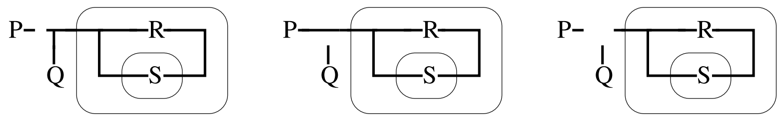
The rule of erasure shall here be exemplified with the following graph:

[1] This has already be mentioned in Sec.**??**.

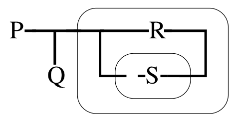
[2] This principle will be adopted in the part 'Extending the System', where the expressiveness of EGCs is extended and new rules are added to the calculus

$\mathfrak{G}$:= 

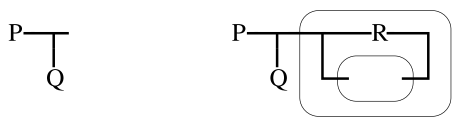
The following three examples are obtained from $\mathfrak{G}$ by erasing a part of the ligature on the sheet of assertion.



Analogously, we can erase parts of the ligature in the innermost cut. For example, we can derive

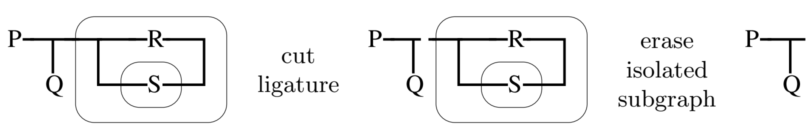


Similar to Alpha, whole subgraphs may be erased from positive contexts. This is how the following two graphs are obtained:

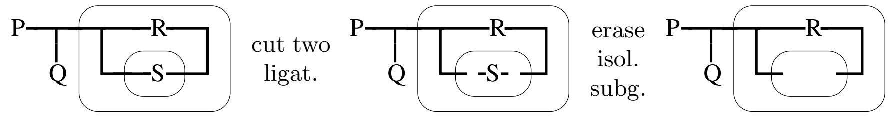


In the first graph, we have erased the whole subgraph of $\mathfrak{G}$ which consists of the outermost cut and all what it is scribed inside. In the second example, the subgraph —S— in the innermost cut has been erased.

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:



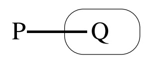
Analogously, the deletion of the subgraph in the second example can be performed as follows:



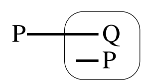
It will turn out that this consideration will be helpful for a mathematization of these rules for EGCs.

6.2 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 my be joined with the existing line of identity. 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:

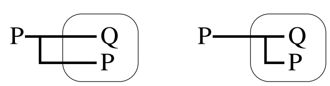
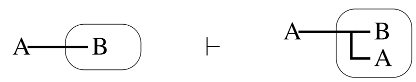


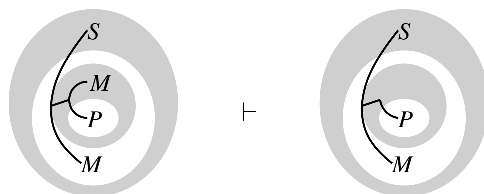
Fig. 6.1. two different existential graphs with the same meaning

As we have in the left graph of Fig. 6.2 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 right 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. 6.2 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. 6.2 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 [**?**], where Peirce uses the rule of deiteration to remove a copy of M— (like in one example we referred to in Chap. 3, he uses shadings for representing cuts):

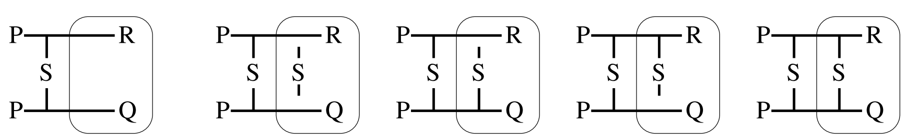


Again it is crucial to note that the copy of M— was connected in the *innermost* cut with the ligature. For Gamma, Peirce provides 1906 in 4.566 another example with the same handling of ligatures when a subgraph is iterated.

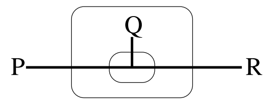
From these examples, I conclude the following understanding of handling ligatures in the iteration-rule:

> **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 five graphs of Fig. 6.2 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 $\mathfrak{G}$.

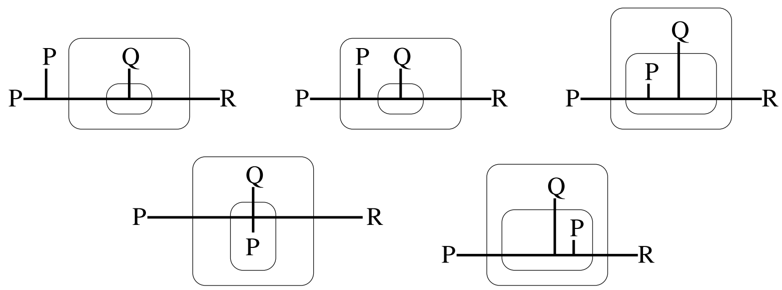


Fig. 6.2. Five results from a correct application of the iteration-rule to $\mathfrak{G}$

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 $\mathfrak{G}$.

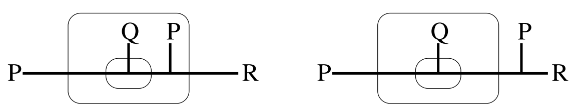


Fig. 6.3. two different results from a incorrect application of the iteration-rule

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	×		×
b		×	

$\mathcal{M}_2 :=$

	P	R	Q
a	×		
b		×	

It can easily be seen that $\mathfrak{G}$ does not hold in M_1, but but the left graph of Fig. 6.3 holds in M_1, thus they have different meanings. Analogously, $\mathfrak{G}$ holds in M_2, but the right graph of Fig. 6.3 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. 6.2 cannot be allowed to be derived by the rule of iteration from $\mathfrak{G}$.

With our understanding, the right graph of of Fig. 6.2 is the result of an correct application of the iteration-rule. We have now in turn to show how the left graph of Fig. 6.2 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 inwards 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.

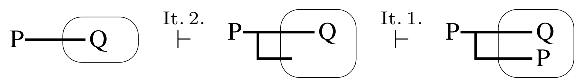
2. A branch with a loose end may be added to a ligature, and this loose end may be extended inwardly through cuts.[3]

[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.

Deiteration:

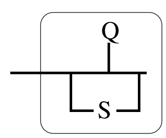
The rule of iteration may be reversed. That is: If $\mathfrak{G}'$ can be derived from $\mathfrak{G}$ with the rule of iteration, then $\mathfrak{G}$ can be derived from $\mathfrak{G}'$ with the rule of deiteration.

With this rule, clause 1., the right graph Fig. 6.2 can be derived from $\mathfrak{G}$. Using both clauses one after the other, it is possible to derive the left graph of Fig. 6.2 from $\mathfrak{G}$ as well, as the following derivation shows:



Moreover, our understanding of the iteration-rule allows to re-arrange ligatures in cuts in nearly arbitrary ways. In the following, this will be exemplified with some examples. In Chap. **??**, 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. 6.4 can be derived by a single application of the iteration-rule:

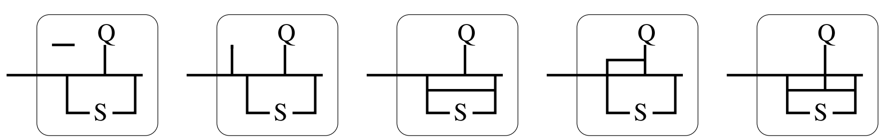
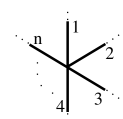


Fig. 6.4. Iterating a Ligature within a Context

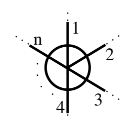
In the first example, a portion of the ligature is iterated without connecting the copy with the ligature. In the second example, a portion of the ligature is iterated, and one of its two ends is connected to the ligature (this could be an application of part 2. of the iteration-rule as well). In the 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 22, I have already mentioned that it is Peirce did not consider EGs where branching points with more than three branches occur. 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.

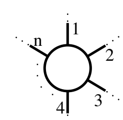
A ligature may contain branching points with more than three branches. A branching point with n branches can be sketched as follows:



With an n-fold application of the last lemma, we can add n lines of identity to this part of the ligature, which yields:

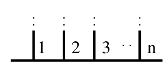


Now, again with the last lemma, we can remove all lines of identity within the circle. Then we obtain:



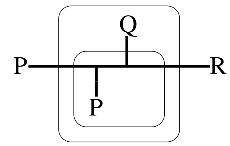
With this procedure, we can transform each identity spot in which more than three LoIs terminate into a 'wheel' on which only teridentity spots occur.

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 [**?**] instead of branching points with n branches):

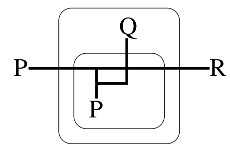


Next, I want to show that "branches of a ligature may be moved along that ligature". In order to understand this phrase, consider the third, fourth and fifth graph of Fig. 6.2. These graphs differ only in the fact that the branch P— of the ligature in the innermost cut is attached at different positions to that ligature. These graphs can be easily transformed into each other. I will exemplify this with the third and forth graph of Fig. 6.2 (the shape and length of some lines of identity are slightly different to the graphs of Fig. 6.2, but "the shape and length [of some lines of identity] are matters of indifference" (4.416)).

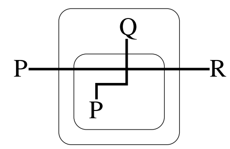
We start with the third graph of Fig. 6.2:



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:

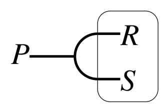


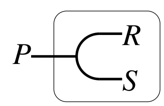
The portion of the ligature between the two branches above P could have been inserted by a similar application of the iteration-rule like in the last step. Thus we are allowed to remove it with the deiteration-rule. This yields the graph on the right, which is the fourth graph of Fig. 6.2:



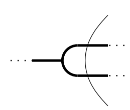
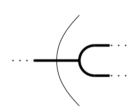
The proof uses only iteration and deiteration and maybe therefore be carried out in both directions.

In Chpt. 3 we have already seen that



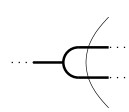
and


have the same meaning (see page 38). 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:

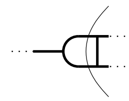
A devise like  can be replaced by  and vice versa

(in this representation, we have sketched a segment of a cut line, and we agree that the whole device is part of an 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.

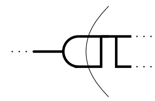
Let a graph be given where the left device occures. I.e., we start with the device on the right, placed in an arbitrary context:



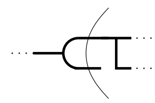
First we iterate a part of the ligature of the outer cut into the inner cut:



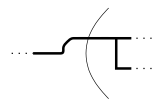
Now we move the branch in the inner cut:



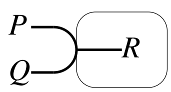
The iteration of the first step is reversed with the deiteration-rule:



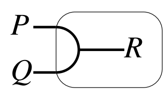
The 'loose' end of the ligature is retracted with the deiteration-rule. We obtain right device, which is the device we wanted to derive (drawn slightly different).



A more general and formally elaborated version of this transformation will be provided for EGCs in Chpt. **??**. Finally, recall that we have already seen that the following graphs have different meanings (see page 3.4):



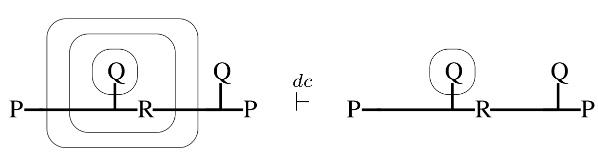
and



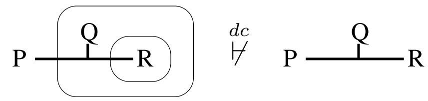
The above given proof cannot be applied to these graphs, as it relies on some applications of the iteration-rule to parts of the ligature in the outer cut.

6.3 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 Chap. 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:

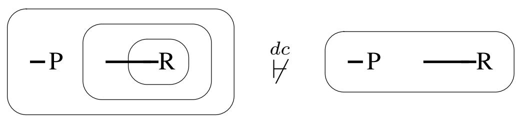


It is obvious that the area of the outer cut is not allowed to contain any relation-names. For example, although we have a ligature which passes through both cuts, the following is an invalid application of the double-cut rule (indicated by the crossing out of the symbol '$\vdash$'):



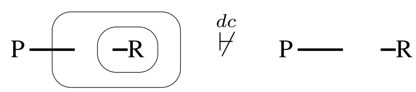
In fact, if we consider the relational structure $\mathcal{M}_1$ with a single element a, such that $P(a)$ and $R(a)$, but not $Q(a)$ holds, the left graph holds in $\mathcal{M}_1$, but the right graph does not.

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:



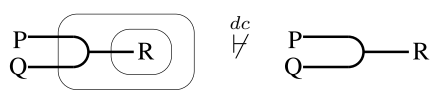
The left graph has the meaning 'if there is an object with property P, then there is an object which has not property R', while the meaning of the right graph is 'it is not true that there exists an object with the property P and an object with the property R', thus this conclusion is not semantically 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 $\mathcal{M}_2$:

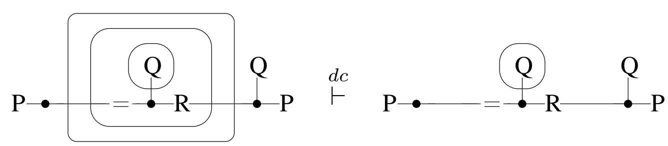
$\mathcal{M}_2 :=$

	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 EGCs if we go from Alpha to Beta. That is, for EGCs, 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 EGC, 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 EGCs. A application of the double cut rule of the corresponding EGCs 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):



6.4 Inserting and Deleting a Heavy Dot

In our semantics, we consider, as usual in mathematics, only *non-empty* relational structures as models (see Def. 5.1). 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 calulus, it is convenient to add this principle as a rule to it.

7

Calculus for Existential Graphs

In this chapter, we will provide the formal definitions for the calculus for EGCs and EGs.

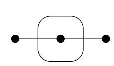
We start with the calculus for EGCs. Before its formal definition is given, we first have to introduce a simple relation on the set of vertices of an EGC. Recall that in the iteration-rule, where a subgraph is iterated from a context c into a context d, we had to consider 'ligatures which go inwardly from c to d'. This idea is formally captured as a relation Θ on the set of vertices.

Definition 7.1 (Θ).

Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ be an EGC. 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. *$cut(v_1) \geq cut(v_2) \geq \ldots \geq cut(v_n)$, and*
3. *for each $i = 1, \ldots, n-1$, there exists an identity link $e_i = \{v_i, v_{i+1}\}$ between v_i and v_{i+1} with $cut(e_i) = cut(v_{i+1})$.*

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. **??**, we will first describe the whole calculus using common spoken language. After this, we present mathematical definitions for the rules.

Definition 7.2 (Calculus for Existential Graphs Candidates).
The calculus for EGCs 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* $c \leq cut(\mathfrak{G}_0)$ *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* $cut(v) = cut(\mathfrak{G}_0)$ *is a vertex, and if* $w \in V_0$ *with* $cut(w) = c$ *is a vertex with* $v\Theta w$*, then an identity link between* v *and* w *may be inserted into* c*.*
  - *If* $v \in V_0$ *is a vertex, and if* $c \leq cut(v)$ *is a cut, then a new vertex* w *and an identity link 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 EGC graph and $c_1, c_2 \in Cut$ such that $area(c_1) = \{c_2\}$. Let $c_0 := cut(c_1)$ (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**

  We first provide general definitions for inserting and erasing vertices, edges and closed subgraphs.

  Let $e \in E$, and let $\mathfrak{G}^{(e)} := (V^{(e)}, E^{(e)}, \nu^{(e)}, \top^{(e)}, Cut^{(e)}, area^{(e)}, \kappa^{(e)})$ be the following graph:

  - $V^{(e)} := V$
  - $E^{(e)} := E \backslash \{e\}$
  - $\nu^{(e)} := \nu|_{E^{(e)}}$
  - $\top^{(e)} := \top$
  - $Cut^{(e)} := Cut$
  - $area^{(e)}(d) := area(d) \backslash \{e\}$ for all $d \in Cut^{(e)} \cup \{\top^{(e)}\}$
  - $\kappa^{(e)} := \kappa|_{E^{(e)}}$

  Let $c := cut(e)$. 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 EGC 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:

  - $V' := V \backslash V_0$
  - $E' := E \backslash E_0$
  - $\nu' := \nu|_{E'}$
  - $\top' := \top$
  - $Cut' := Cut \backslash Cut_0$
  - $area'(d) := \begin{cases} area(d) & d \neq \top_0 \\ area(d) \backslash (V_0 \cup E_0 \cup Cut_0) & d = \top_0 \end{cases}.$
  - $\kappa' := \kappa|_{E'}$

  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 'erasure and insertion, erasing and inserting a vertex' are restrictions of the general definitions above:

  - **erasure and insertion**

    Let $\mathfrak{G}$ be an EGC and let k be an edge or a closed subgraph of $\mathfrak{G}$ with $c := cut(k)$, 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 EGC and let $v \in V$ be a vertex with $E_v = \emptyset$ and $\kappa(v) = \top$. Then $\mathfrak{G}_0 := (\{v\}, \emptyset, \emptyset, cut(v), \emptyset, \emptyset, \emptyset)$ is a closed subgraph only consisting v. Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by erasing v from the context $cut(v)$. 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 EGC with the subgraph $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0)$ and $c \notin Cut_0$ be 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 with $w \Theta v$ 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. Let $F := \{e_{v,w} \mid v \in W_0 \text{ and } w \in W_v\}$.

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

– $V' := V \times \{1\} \;\cup\; V_0 \times \{2\}$

– $E' := E \times \{1\} \;\cup\; E_0 \times \{2\} \;\cup\; F$

– $\nu'(e') := \begin{cases} ((v_1, i), \ldots, (v_n, i)) \text{ for } e' = (e, i) \in E \times \{1\} \;\cup\; E_0 \times \{2\} \text{ and } \nu(e) = (v_1, \ldots, v_n) \\ ((w, 1), (v, 2)) \text{ for } v \in W_0,\ w \in W_v \text{ and } e' = e_{v,w} \end{cases}$

– $\top' := \top$

– $Cut' := Cut \times \{1\} \;\cup\; Cut_0 \times \{2\}$

– $area'$ is defined as follows:

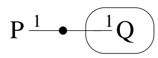
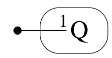
for $(d, i) \in Cut' \cup \{\top'\}$ and $d \neq c$, let $area'((d, i)) := area(d) \times \{i\}$ and

$area'((c, 1)) := area(c) \times \{1\} \cup area_0(\top_0) \times \{2\} \cup F$

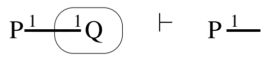
– $\kappa'(e') := \begin{cases} \kappa(e) \text{ for } e' = (e, i) \in E \times \{1\} \;\cup\; E_0 \times \{2\} \\ \doteq \text{ for } e' \in F \end{cases}$

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.

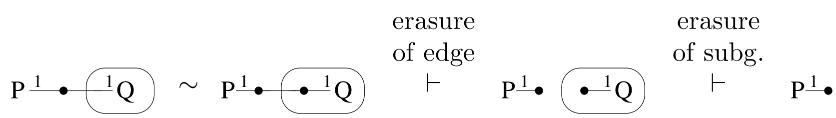
It should be noted that the iteration/deiteration-rule may be applied to arbitrary subgraphs, whilst the erasure/insertion-rule may only be applied to *closed* subgraphs. The reason is that the erasure (or insertion) of a subgraph which is not closed does not yield a well-formed EGC. This causes no troubles, as we already inormally discussed in Sect. 6.1 how non-closed subgraphs can be erated. To see an simple example for EGC, 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 EGCs 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, i.e., we set:

Definition 7.3 (Proof).

Let $\mathfrak{G}_a$, $\mathfrak{G}_b$ be EGCs. 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 $\mathfrak{H} := \{\mathfrak{G}_i \,|\, i \in I\}$ is a (possibly empty) set of EGCs, 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 EGCs can be transferred to a calculus for existential graphs.

Definition 7.4 (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 EGCs (see Def. 7.2, if this holds for some representing EGCs $\mathfrak{G}_a \in \mathfrak{E}_a$ 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]

If we have a proof $(\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_n)$ for two EGCs $\mathfrak{G}_a$, $\mathfrak{G}_b$ 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 EGCs is be used for formal existential graphs will be provided in Chpt. **??**.

[1] Recall that the juxtaposition of existential graphs is carried over from the definition of the juxtaposition for EGCs, as it has been explained directly after Def. 4.17.