

Ligatures in Peirce's Existential Graphs

Frithjof Dau

November 30, 2006

1 Introduction

Peirce's system of Existential Graphs (EGs) is divided into three parts *Alpha*, *Beta* and *Gamma*. Beta builds upon Alpha, and Gamma builds upon Beta. The step from Alpha to Beta corresponds to the step from propositional logic to first order logic (FOL). In this step, a new syntactical element, the *line of identity* (*LoI*), is added to EGs. LoIs are used to denote both the existence of objects and the identity between objects, and they are represented as heavily drawn lines. Consider the EGs of Fig. 1.

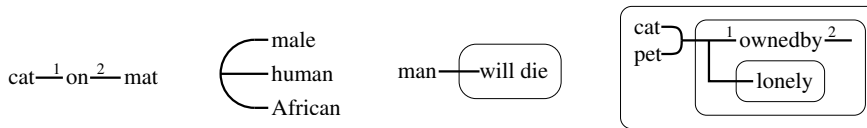


Figure 1: Four Peirce graphs with so-called single-object-ligatures

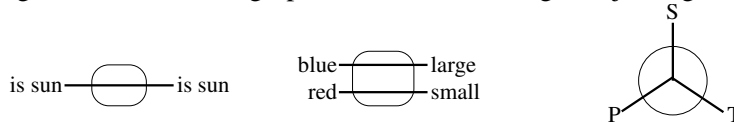


Figure 2: Three Peirce graphs with non-single-object ligatures

All these graphs are drawn on a surface, called the *sheet of assertion* (*SoA*). They are composed of:

1. predicate symbols of arbitrary arity (including 0),
2. LoIs which are sometimes assembled to networks (termed *ligatures*), and

3. ovals, termed *cuts*, which negate the enclosed subgraph.

For predicates of arity ≥ 2 , Peirce used to read the attached LoIs in clockwise order. In the diagrams of Fig. 1, small numbers are added to the attached LoIs instead.

The meaning of these graphs is ‘a cat is on a mat’, ‘there exists a male, human African’, ‘there exists a man who will not die’, and ‘it is not true that there is a pet cat such that it is not true that it is not lonely and owned by some somebody’, i.e., ‘every pet cat is owned by someone and is not lonely’. In all these graphs, LoIs resp. ligatures, even if they cross cuts, are used to denote a single object.

Nonetheless, other examples show that this interpretation of ligatures is simple in every case. Consider the EGs of Fig. 2. They are read ‘there are at least two suns’, ‘there are (not necessarily distinct) objects which are blue, red, large and small, respectively, and the blue and large or the red and small object are distinct’, and ‘there are objects o_1, o_2, o_3 with the properties S, P , and T resp, and these objects are not all identical (i.e., $o_1 = o_2 = o_3$ does not hold)’. A ligature does not denote only a single object in any of these graphs.

For novices it is often not clear how complex ligatures are read. Peirce writes in CP 4.459 (we adopt the usual convention to refer to the collected papers Hartshorne and Burks (1935)) that ‘*A sep¹ which is vacant, except for a line of identity traversing it, expresses with its contents the non-identity of the extremities of that line.*’ This quotation seems to be a convention which explains the reading of the first graph of Fig. 2. But it is in fact, as Peirce writes in CP 4.459, an ‘*Interpretational Corollary*’ from his deeper understanding of heavily drawn lines. From this understanding, the reading of the remaining two graphs can be obtained as well. Providing a means to read these graphs is one of the goals of this chapter.

In Dau (2006b), Peirce’s writings are extensively explored, and from this exploration, a formal, mathematical elaboration of Peirce’s graphs is obtained. Particularly, the reading and handling of ligatures is discussed. Providing the mathematical definitions and results of Dau (2006b) is far beyond the scope of this chapter. Instead, the main results of Dau (2006b) for ligatures are given in a more informal, but nonetheless precise, manner.

The organization of this chapter is as follows. In the next section, the distinction between graphs and their diagrammatic representations is briefly discussed. In Sec. 3, some important technical terms are introduced. In Sec. 4, the handling of ligatures in Peirce’s transformation rules for graphs is discussed. In Sec. 5 it is investigated how ligatures in a graph can be rearranged. Finally in Sec. 6, a specific kind of ligatures, so called *single-object-ligatures*,

is introduced, and it is discussed how they can be used to ease the reading of graphs with complex ligatures.

2 Graphs and Graph Replicas

The ongoing scrutiny is exemplified with several diagrams of EGs. For this reason, it has to be stressed that EGs should not be mistaken with their graphical representations. Peirce draws a clear distinction between EGs and their diagrams. Using the terminology of Peirce’s semiotics, a graph is a *symbol*, and any concrete representation of a symbol is a *replica of that symbol*. As Peirce says: ‘*Symbols act through replicas.*’

Particularly, an EG can be represented in many different ways (i.e., one symbol can have many different replicas). Peirce explicitly said that arbitrary features of the diagrams may vary, as long as they represent the same EG. At the beginning of Peirce (1935) he says: ‘*Convention No. Zero. Any feature of these diagrams that is not expressly or by previous conventions of languages required by the conventions to have a given character may be varied at will. This “convention” is numbered zero, because it is understood in all agreements.*’ (a similar explication can be found in 4.507). For LoIs, he says even more explicitly in Peirce (1935) that ‘*its shape and length are matters of indifference*’, and finally, in 4.500 we find that ‘*Lines of Identity are replicas of the linear graph of identity*’.

To provide a simple example, consider the diagrams of Fig. 3. Although looking quite different, the first two diagrams only differ in the shape and arrangement of the lines of identity, thus they are different replicas of the *same graph*. On the other hand, the third diagram differs in two respects from the first and second one: The order of the predicates *R* and *Q*, as they are attached to the ligature, has changed, and the rightmost branching point is now placed on the SoA instead of the cut. Thus the third diagram is a replica of a *different graph*.

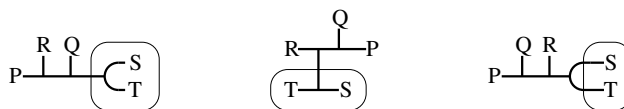


Figure 3: two different replicas of the same graph and a further graph

Obviously, the distinction between graphs and graph-replicas is very sim-

ilar to the distinction between types (graphs) and tokens (graph replicas), as it is known from philosophy. The type-token-issue for a mathematical elaboration of diagrammatic logic is more deeply discussed in Dau (2004); Howse et al. (2002).² Nonetheless, we have to come back to this issue when we informally define subgraphs of EGs.

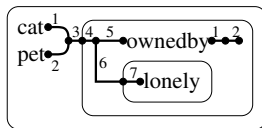
3 Lines of Identity, Ligatures, and Subgraphs

In this section, some important technical terms are clarified. We start with the LoI, which Peirce describes 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). It is important to note that LoIs do not have any branching points, nor they are allowed to *cross* cuts. But it is allowed that they *touch* other elements at their extremities, i.e.:

1. LoIs may be connected at their endpoints. Peirce allows only two or three LoIs to be connected. If three LoIs are connected, the point where they meet is a BRANCHING POINT.
2. LoIs may end on a cut. Particularly, it is allowed that LoIs are connected directly on a cut. Due to this possibility, we can have heavily drawn lines (composed of several LoIs) which cross a cut.

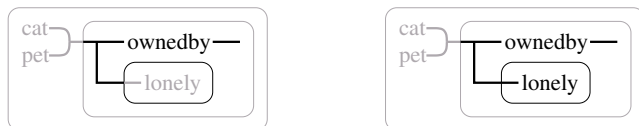
LoIs may be assembled to connected networks termed LIGATURES. Peirce writes in 4.407: ‘*A collection composed of any line of identity together with all others that are connected with it directly or through still others is termed a ligature. Thus, ligatures often cross cuts, [...]*’, and later on in 4.416, he writes ‘*The totality of all the lines of identity that join one another is termed a ligature.*’ Particularly, each LoI is a ligature, but not vice versa.

This will be exemplified with the last graph of Fig. 1. This graph has two maximal ligatures. The left one is composed of (at least) seven LoIs. In the diagram below, these LoIs are numbered, and all endpoints of LoIs are indicated as bold spots. The right heavy line is a single LoI, but even single LoIs can be understood to be composed of smaller LoIs as well. This is indicated by breaking up this line into two LoIs.



Peirce terms the *totality* of all the LOIs that join one another a ligature, thus for him a ligature is a *maximal* connected network of LOIs. To clarify matters, in this chapter *each* connected network of LOIs is called ligature.

Peirce often used the terms ‘area (of a cut)’ and ‘being enclosed (by a cut)’. These terms are not synonymous and shall be clarified now. First of all, a CONTEXT is either the SoA, or any cut. The AREA of a context c contains all elements of an EG which are drawn inside c , but which are not more deeply nested. The context c itself does not belong to its area. An element is said to be ENCLOSED BY c if it is drawn inside c , even if it is more deeply nested. An element in the area of c is more specifically said to be DIRECTLY ENCLOSED BY c , and c is said to be THE CONTEXT OF THAT ELEMENT. The next two diagrams exemplify these terms. Let us denote the outermost cut with c_1 , the middle cut with c_2 and the innermost cut with c_3 . The left diagram indicates all elements located in the area of c_2 by drawing them black (the remaining elements are grayed out). Analogously, the right diagram indicates all elements enclosed by c_2 .



For the calculus we need to distinguish between evenly and oddly enclosed elements. An element is said to be EVENLY ENCLOSED (ODDLY ENCLOSED) if and only if it is enclosed by an even (odd) number of cuts. Evenly (oddly) enclosed contexts are sometimes called NEGATIVE (POSITIVE). Note that an element is evenly (oddly) enclosed if and only if the context of that element is a positive (negative) context. The left diagram below indicates the evenly enclosed elements and the right diagram the oddly enclosed elements of our sample graph.

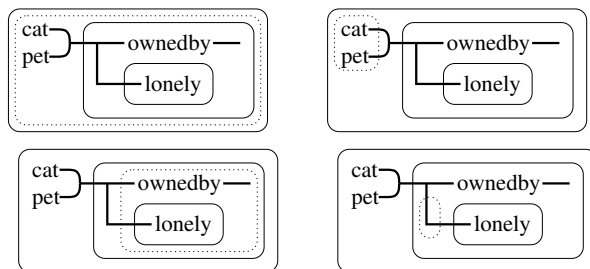


Please note that for each ligature l , there exists a uniquely given context c enclosing l , such that at least some portion of l is directly enclosed by c . We will say that c IS THE CONTEXT OF l resp. l IS STARTS IN THE AREA OF c .

Except the double cut rule, all rules of the calculus rely on the notation of a SUBGRAPH. For the diagrams of EGs, we can provide an intuitive definition of subgraphs, which is inspired by Zeman’s definition in Zeman (1964).

Definition 3.1 (Subgraph-Lines and Subgraphs) *A subgraph-line in the diagram of an EG is a closed, doublepoint-free and smooth curve, which does not touch or cross cuts, nor predicates, and which does not touch heavily drawn lines (but it is allowed that heavily drawn lines are crossed). To distinguish subgraph-lines from cuts, they are drawn in a dotted manner. Given a graph G , any graph which is the part of G enclosed by a subgraph-line in some diagram of the graph is a SUBGRAPH of G .*

Below, four examples of subgraphs are depicted. Please note that each subgraph is located in an uniquely given context: In first two diagrams, this context is c_1 , in the last two diagrams, it is c_2 .



Note that subgraph-lines are defined on the token-level, whereas ‘subgraph’ is defined on the type-level. More precisely, a graph G_s is a subgraph of a given graph G , if *there exists* a diagram of G where G_s is given by a subgraph-line *in that diagram*.³ That is: Sometimes, we have to redraw a diagram (i.e., provide a different token for the same type) in order to find a specific subgraph. For example, consider the left graph below. The graph consisting of the two inner cuts and their enclosures is a subgraph, but in the left diagram, it cannot be indicated by a subgraph-line. Only if we redraw the diagram, for example as in the right diagram, we can find a subgraph-line for this subgraph.



4 Ligatures in the Calculus

In this section, it shall be investigated how ligatures are handled in Peirce’s rules for EGs. In different writings on EGs, Peirce provided different versions

of transformation rules, which differ in details. Roughly speaking, the rules are as follows:

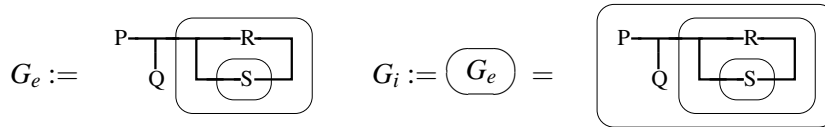
1. **Rule of Erasure and Insertion:** In positive contexts, any subgraph may be erased, and in negative contexts, any subgraph may be inserted.
2. **Rule of Iteration and Deiteration:** If a subgraph of a graph is given, a copy of this subgraph may be inserted into the same or a more deeply nested context, and the copy may be connected with its original. This transformation may be carried out in both directions.
3. **Double Cut Rule (Erasure and Insertion of Double Cuts):** Two Cuts one within another, with nothing between them, except ligatures which pass entirely through both cuts, may be inserted into or erased from any context.
4. **Erasure and Insertion of Lines of Identity:** An isolated LoI may be inserted into or erased from arbitrary contexts.

The following elaboration of the handling of ligatures in Peirce's rules tries to cover Peirce's general understanding of his rules.

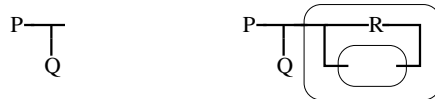
4.1 Erasure and Insertion

The erasure- and insertion-rule is described by Peirce in 4.505 as follows: *'In even seps, any graph-replica can be erased; in odd seps any graph-replica can be inserted.'* In a note to this rule, he explains further that in *'the erasure of a graph by this rule, all its ligatures must be cut.'*

These rules shall be illustrated with the graphs G_e for erasure, G_i for insertion:



With Peirce's explanation of the rule, we see that evenly enclosed subgraphs can be erased, i.e., if a diagram of an EG be given with a subgraph-line s in a positive context, then s and all what is drawn inside s can be erased. This is how the following two graphs are obtained from G_e :

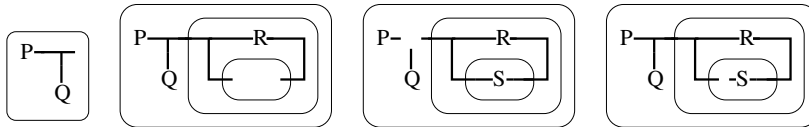


In the first graph, we have erased the whole subgraph of G_e which consists of the outermost cut and all what it is drawn inside. In the second example, the subgraph $-S-$ in the innermost cut has been erased.

Due to the definition of a subgraph, even portions of a ligature are subgraphs. In fact, in another note for the erasure- and insertion rule in 4.505, Peirce says that *'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 following two examples are obtained from G_e by erasing a part of the ligature on the SoA resp. in the innermost cut.



The insertion-rule can be understood best as the inverse direction of the erasure rule, applied to negative contexts: If a graph G can be obtained from G' by erasing an oddly enclosed subgraph, than G' can be derived from G with the insertion-rule. Thus we can derive G_i from each of the following four graphs.



Please note that when a subgraph is inserted into a context c of a graph G , it is allowed that outermost ends of heavily drawn lines of the subgraph are connected to ends of heavily drawn lines of G which are located in c .

4.2 Iteration and Deiteration

If a graph with a subgraph G_s in a context c is given, the iteration-rule allows to write a copy of G_s into c , or into any more deeply nested context which does not belong to G_s . Consider the two graphs below.



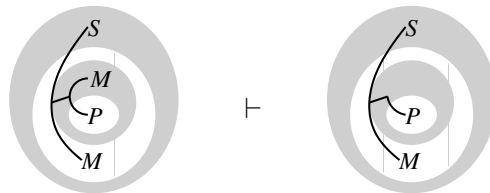
G_1 can be derived from G_{it} by iterating the subgraph $\text{P} \text{---} \text{---} \text{Q}$. This kind of iteration corresponds to the Alpha part of EGs and is easy to understand. But for Beta, the handling of ligatures is crucial. The iterated subgraph may be joined with the existing heavily drawn line. The question is: how?

In 4.506, Peirce writes that the iteration rule ‘includes the right to draw a new branch to each ligature of the original replica inwards to the new replica.’ Consider the following two graphs, which have the same meaning:

$$G_2 := \text{P} \begin{array}{|c} \text{---} \text{Q} \\ \text{---} \text{P} \end{array} \quad G_3 := \text{P} \begin{array}{|c} \text{---} \text{Q} \\ \text{---} \text{P} \end{array}$$

As we have in G_2 a branch from the old replica of $\text{P} \text{---}$ which goes inwardly to the copy of the iterated subgraph, one might think that G_2 results from a correct application of the iteration-rule to G_{it} , while G_3 does not. But, if the calculus is complete, it must be possible to show that G_2 and G_3 can be transformed into each other with its rules. 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 G_3 appears to be the right result of an application of the iteration-rule.

In 4.386 Peirce provides an illuminating example how the alpha-rule of iteration is amended to beta. He writes: ‘Thus, $[A \text{---} (\text{---} B)]$ can be transformed to $[A \text{---} (A \text{---} B)]$.’ Peirce uses in this place a notation with brackets. It is crucial to note that the LoI in the copy of $A \text{---}$ is connected *inside* the cut with the already existing ligature. A similar example can be found in Peirce and Sowa (2000), where Peirce deiterates a copy of $M \text{---}$ (here Peirce uses shadings for representing cuts):

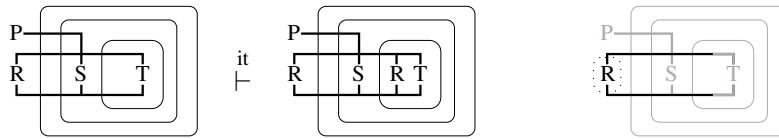


Again it is crucial to note that the copy of $M \text{---}$ was connected in the *third innermost* cut with the ligature. For Gamma, Peirce provides 1906 in 4.566 a similar example.

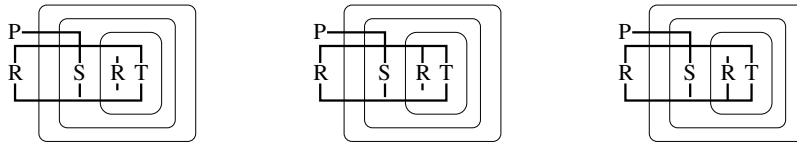
From these examples, we can conclude the following understanding of handling ligatures in the iteration-rule: Assume that a subgraph G_s is iterated from a context c into a context d . Furthermore, assume that G_s is connected to a heavily drawn line which goes inwardly from c to d . Then the copy of G_s may be connected *in* d with this line.

In order to provide a (slightly) more sophisticated example for the handling of ligatures, consider the following graphs. The second graph is derived

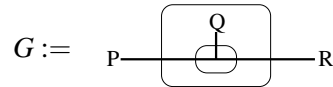
from the first graph by iterating the subgraph $\begin{array}{c} | \\ \text{R} \\ | \end{array}$ from the SoA into the innermost cut d . In the rightmost diagram, we have indicated the iterated subgraph with a subgraph-line, and the heavily drawn lines going inwardly from SoA to d . Please note that it is allowed that this line has already existing further branches (leading to P , S and T).



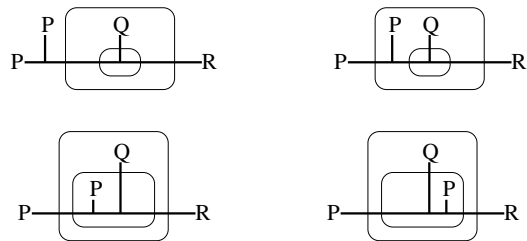
We are allowed to connect the iterated subgraph with the heavily drawn lines going inwardly from SoA to d , but we don't have to. Thus the following three graphs are results of an iteration of the subgraph $\begin{array}{c} | \\ \text{R} \\ | \end{array}$, too.



The phrase “which goes inwardly from c to d ” is crucial for the correct application of the iteration-rule. Consider the following graph:

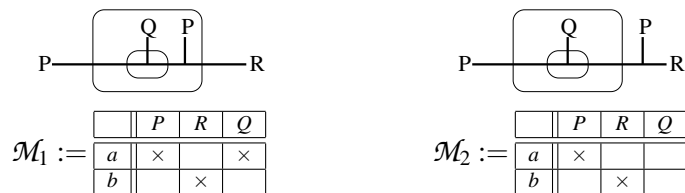


The next four graphs are results of a correct application of the iteration-rule to G by iterating the subgraph $\begin{array}{c} \text{P} \text{---} \end{array}$.



Now consider the following two graphs, where a copy of $\begin{array}{c} \text{P} \text{---} \end{array}$ is connected to a heavily drawn line which does not “go inwardly from c to d ” (the

line crosses some cuts more than once), and two models with elements a, b .



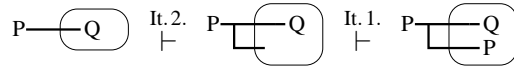
The left graph holds in \mathcal{M}_1 , but G does not, so they have different meanings. Analogously, G holds in \mathcal{M}_2 , but the right graph does not, so these two graphs have different meanings, too. Iteration does not change the meaning of a graph (as it can be reversed by the rule of deiteration), thus these graphs cannot be obtained by iterating $P \dashv\dashv$.

We have seen that G_3 is the result of an correct application of the iteration-rule to G_{it} . We have now to show how G_2 can be derived from G_{it} . In 4.506, Peirce continues his explanation of the iteration-rule: ‘*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.*’ With our handling of ligatures in the iteration-rule, it is not clear why loose ends of ligatures can be be extended inwardly through cuts. For this reason, this extension of ligatures should be considered not as a consequence of the iteration-rule, but as separate clause which cannot be left out. The iteration-rule can thus be summarized as follows:

1. Let a diagram of an EG be given with a subgraph G_s (indicated by a subgraph-line) in a context c and let d be a context which is identical to c or enclosed by c , and which does not belong to G_s . Then a copy of G_s may be drawn into the area of d . In this transformation, the following is allowed: If we have a heavily drawn line which proceeds inwardly (particularly, it crosses no cut-line more than once) to d , and if this line crosses the subgraph-line at a point p , then the copy of p (which is a part of the copy of G_s) may be connected in d to the heavily drawn line.
2. It is allowed to add new branches to a ligature, or to extend any LoI inwardly through cuts.

The deiteration-rule is just the inverse direction of the iteration-rule, that is, it can be best described as follows: If G_s is a subgraph of G which could have been inserted by rule of iteration, then it may be erased.

We have already seen that G_3 can be derived from G_{it} with the iteration-rule. Now we can derive G_2 from G_{it} with the iteration-rule as well.

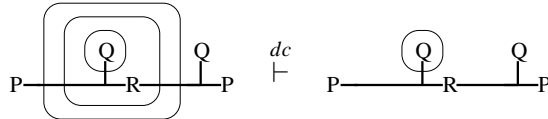


Finally, as mentioned in the second clause of the iteration-rule, it is not allowed to extend ligatures outwardly, as the following example shows (this is an example which shows that the rule NR3 provided by Shin in Shin (2002) is not sound).

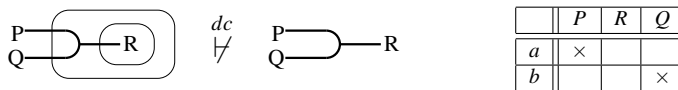


4.3 Double Cuts

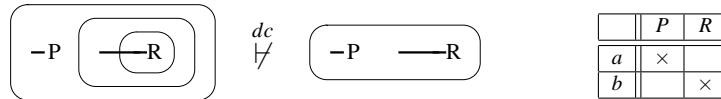
For the Alpha part of EGs, the double cut rule allows to add or remove two cuts if nothing is between them (that is, the area of the first cut contains nothing but the second cut). For Beta, this rule is extended to cover ligatures. Their handling is described by Peirce in 4.567: ‘Two Cuts one within 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:



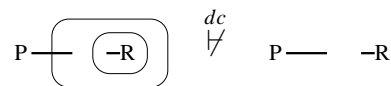
“To pass through both cuts” has to be understood very rigidly: It is crucial that each ligature passes *directly and entirely* through both cuts. In a note for another definition of the double cut rule in 4.508, Peirce describes two cases which do not fulfill this condition as follows: ‘A ligature passing twice through the outer sep without passing through the inner one, or passing from within the inner one into the intermediate space and stopping there’ is not allowed in the double cut rule. The first case can even be strengthened: A ligature which crosses the outer cut twice, even if it crosses the inner cut as well, may cause problems. The next example is an invalid application of the double-cut rule, as the graphs are not semantically equivalent (on the right, you find a model in which the left graph holds, but the right one does not hold):



Similarly for the second case, if a ligature starts in the area of the outer cut, we may obtain an invalid conclusion. This is shown by the following example (again the graphs are not semantically equivalent, which is shown by a model in which the left graph holds, but the right one does not hold):



Strictly speaking, even in the next example, where we now have semantically equivalent graphs, is an invalid application of the double cut rule. In order to derive the second graph from the first, we *first have to retract the left line out of the cut with the deiteration rule*, and then double cut rule can be applied.



4.4 Erasure and Insertion of a Line of Identity

Surprisingly, usually this rule is not explicitly given, neither in Peirce's manuscripts, nor in secondary literature. On the other hand, in the usual semantics of FOL, only *non-empty* models are considered. For this reason, it must be possible to derive the EG which expresses that there is an object, i.e., the EG which contains only a single LoI.

Peirce considered only nonempty universes of discourse as well. 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.*' A LoI is in Peirce's understanding composed of dots. In 4.474, Peirce 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.' As an isolated LoI denotes, like a dot, the existence of some object, we see that the rule of erasing or inserting isolated LoIs in arbitrary contexts is in accordance with Peirce.

5 Rearranging Ligatures

The correct understanding of the iteration-rule allows us to re-arrange ligatures in many ways. In the following, this will be exemplified.

The main idea in all following examples is to iterate or deiterate a portion of a ligature in a context. If a portion of a ligature is iterated, the rule of iteration allows to connect endpoints of the iterated copy with arbitrary points of the ligature. In Fig. 4, each graph is derived from its predecessor with single application of the iteration-rule.

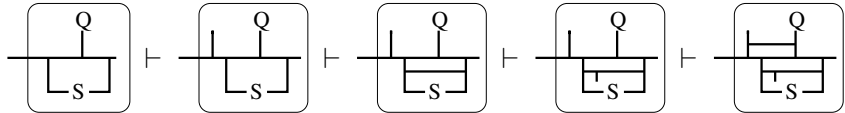
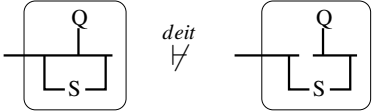
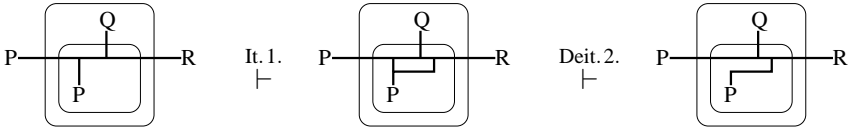


Figure 4: Iterating a part of a ligature within a context

So we see that within a context, a ligature may be extended in arbitrary ways. Vice versa, with the deiteration-rule, we can remove arbitrary parts of a ligature within a context (we can derive each graph in the example from its successor with the deiteration-rule). This is always possible, as long as the remaining ligature keeps connected. But of course, it is not possible to tear a ligature apart with the deiteration-rule. The next example is not valid application of the deiteration-rule, and in fact, these graphs have different meanings.



With the possibility to extend and retract ligatures within a context, it is particularly possible to “move branches along a ligature within a context”. In order to understand this phrase, consider the third and fourth graph on page 10 we have obtained from G with the iteration rule. 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, as the following proofs shows.



The proof uses only iteration and deiteration and may therefore be carried out in both directions. The following theorem (‘theorem’ should not be understood in a mathematical sense) summarizes the discussion so far.

theorem 5.1 *Within a context, we can rearrange any part of a ligature in arbitrary ways, as long as it keeps connected.*

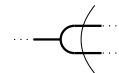
So far, we have considered ligatures being wholly placed within the area of a specific context. Moreover, we have the possibility to rearrange in some sense ligatures when they cross cuts. In order to see an example, we come back to the graphs G_2 and G_3 of the last section. These graphs differ only in the place of the branching point, which has essentially moved from the SoA inside the cut. Such a transformation is always possible, as the following theorem shows.

theorem 5.2 *In Peirce's graphs, we have:*

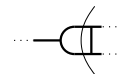
A device like  can be replaced by  and vice versa.

(in this representation, we have sketched a segment of a cut line, and we agree that the whole device is part of a graph, placed in an arbitrary context). With the rules iteration/deiteration and the possibility to move branches along ligatures, we can now prove theorem 5.2.

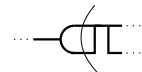
Let a graph be given where the leftmost device of the theorem occurs.



First, we iterate a part of the ligature of the outer cut into the inner cut, and connect it:



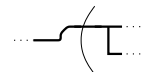
Then, we move the lower branch in the inner cut:



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 the right device, which is the device we wanted to derive (drawn slightly different).

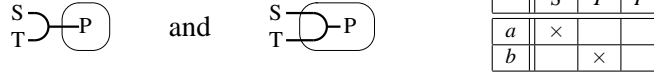


q.e.d.

But it has to be emphasized that we cannot move branching points in arbitrary ways along cuts: It is *not* possible to replace

A device like  by  (or vice versa).

In fact, the following graphs have different meanings, as only the right graph holds in the given model.



6 Ligatures in the Reading of Graphs

In the readings of the graphs of Fig. 1, we assigned to each ligature a single object. To put it more formally: We can provide translations of the EGs to formulas of FOL where we assign to each ligature *one* variable. In fact, we can translate the EGs of Fig. 1 to FOL as follows:

$$\begin{aligned} & \exists x. \exists y. (cat(x) \wedge on(x, y) \wedge mat(y)) \\ & \exists x. (male(x) \wedge human(x) \wedge African(x)) \\ & \exists x. (man(x) \wedge \neg willdie(x)) \\ & \neg \exists x. (cat(x) \wedge pet(x) \wedge \neg (\exists y : ownedby(x, y) \wedge \neg lonely(x))) \end{aligned}$$

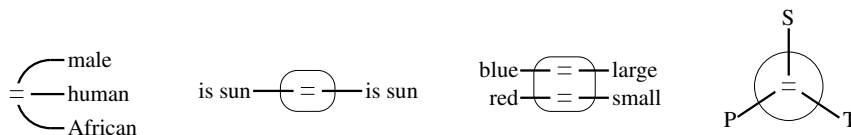
On the other hand, for the graphs of Fig. 2, we have to assign more than one variable to the ligatures. The translations to FOL are:

$$\begin{aligned} & \exists x. \exists y. (issun(x) \wedge issun(y) \wedge x \neq y) \\ & \exists x. \exists y. \exists u. \exists v. (blue(x) \wedge large(y) \wedge red(u) \wedge small(v) \wedge \neg (x = y \wedge u = v)) \\ & \exists x. \exists y. \exists z. (S(x) \wedge P(y) \wedge T(z) \wedge \neg (x = y = z)) \end{aligned}$$

In all graphs of Fig. 2, a part of the ligature traverses a cut (i.e., there is a cut c and a heavily drawn line l which is part of the ligature such that both endpoints of l touch c and the remainder of l is enclosed by c). Such a device denotes non-identity of the endpoints of l , thus a ligature containing such a device l usually denotes different objects. But if such a device does not occur, it has been shown in Dau (2006b) that the ligature denotes a single object. A ligature l such that no part of l traverses any cut will be called SINGLE-OBJECT-LIGATURE (SO-LIGATURE). Note that so-ligatures may contain cycles (see the graphs in Fig. 4) or may cross a cut more than once (see the graphs G_2 and G_3 of section 4.2).

The clue to read arbitrary EGs is to break up non-so-ligatures into several so-ligatures by adding additional equality relations. Peirce writes in Peirce and Sowa (2000) that the second graph of Fig. 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.’ That is, we can replace the branching point by a relation $=_3$, termed TERIDENTITY, expressing that the objects denoted by the attached LOIs are all identical. The corresponding graph is the leftmost graph below (in the diagrams, the indices on the equality-signs are omitted). Even simpler, when two LOIs meet in a point, we can replace this point by the usual binary identity relation $=_2$. So, if an EG with a non-so-ligature is given, we can replace some branching points by $=_3$ and some non-branching points by $=_2$ until the non-so-ligature is split up into several so-ligatures. The graphs we obtain this way from the graphs of Fig. 2 are the second, third and fourth graphs shown below. Now we have two, four and three so-ligatures, respectively, thus the above translations of these graphs to FOL need two, four and three variables.



Replacing branching points by $=_3$ has already been carried out by Zeman in Zeman (1964) in his translation of EGs to formulas. But Zeman replaces *each* branching point by an identity relation, and moreover, he *always* splits heavily drawn lines if they cross a cut more than once (by adding $=_2$). Thus in nearly all cases, Zeman translation uses far more variables than necessary, and the resulting formulas are hard to read.

This has been thoroughly discussed by Shin in Shin (2002). She correctly points out that Zeman’s reading algorithm for existential graphs is comprehensive and yields correct results (in contrast to Robert’s reading in Roberts (1973), as she argues), but as Shin writes, ‘Zeman’s reading fails to capture Peirce’s motivation for the use of lines [of identity]’, and it usually yields a ‘translation that looks more complicated than the original graph’. A main reason for her criticism is the ‘mismatch between the number of lines in a graph and the number of the variables in the translation’. Shin tries to overcome this problem in her reading algorithm, but unfortunately, sometimes she assigns *too few* variables to ligatures, i.e., her reading algorithm is flawed (see Dau (2006a)). With the observation that so-ligatures denote single objects, it is possible to improve Zeman’s reading of EGs, as well as to fix Shin’s reading of EGs. The latter is described in Dau (2006a). In the following, a translation of graphs to formulas in the style of Zeman is described.

We translate each EG G to a formula $\Phi(G)$. In the formulas, we use the existential quantifier ‘ \exists ’ and the junctors ‘ \wedge ’ and ‘ \neg ’. Moreover, to ease

matters, let us agree that we have a further symbol \top standing for ‘true’ (this is helpful for translating empty cuts).

Let G be an EG. Each ligature l which is not attached to any predicate can be, by successively applying the deiteration rule, retracted to an isolated LoI (which is directly enclosed by the context c in which l started), and this isolated LoI can be removed with the ‘Erasure and Insertion of a LoI’ rule. This transformation can be carried out in both directions, i.e., removing a ligature which is not attached to any predicate does not change the meaning of the graph. In the following, we can therefore assume that each ligature is attached to at least one predicate (without this restriction, the following algorithm could yield mal-formed formulas).

Now the following algorithm describes how $\Phi(G)$ is obtained.

1. Transform all ligatures into so-ligatures by appropriately adding relation signs ‘=’.
2. Assign a new variable to each so-ligature.
3. Now we assign to each predicate (occurrence) P of G a formula $\Phi(G, P)$ as follows: Let P be attached to ligatures l_1, \dots, l_n , respectively, and let $\alpha_1, \dots, \alpha_n$ be the variables we assigned to l_1, \dots, l_n . Then let $\Phi(G, P) := P(\alpha_1, \dots, \alpha_n)$.
4. Now we assign inductively to each context c in G a formula $\Phi(G, c)$. If the area of c is empty, we simply set $\Phi(G, c) := \top$. Otherwise let

$$\begin{aligned} \Phi(G, c) \quad := \quad & \exists \alpha_1 \dots \exists \alpha_k : (\Phi(G, P_1) \wedge \Phi(G, P_2) \wedge \dots \wedge \Phi(G, P_l) \\ & \wedge \neg \Phi(G, c_1) \wedge \neg \Phi(G, c_2) \wedge \dots \wedge \neg \Phi(G, c_m)) \end{aligned}$$

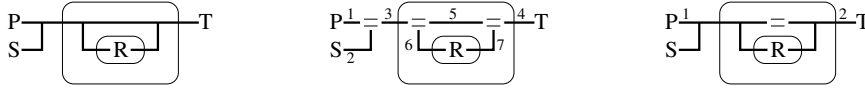
where

- $\alpha_1, \dots, \alpha_k$ are the variables which are assigned to the ligatures which start in the area of c ,
- P_1, \dots, P_l are the occurrences of predicates which are directly enclosed by c , and
- c_1, \dots, c_m are the cuts which are directly enclosed by c .

5. Finally we set $\Phi(G) := \Phi(G, SoA)$.

The formulas at the beginning of this section are examples of this readings (in these formulas, we used the common infix notation for identity, i.e. we write $x = y$ instead of $=(x, y)$ and $x = y = z$ instead of $=_3(x, y, z)$).

Below, a sample graph for Zeman’s reading and the herein presented algorithm is provided. In the middle, the ligatures of the graph are split due to Zeman’s algorithm, on the right, this is done according to our algorithm. The corresponding translations f_Z of Zeman and f_D of Dau show the difference.



$$\begin{aligned}
 f_Z &= \exists x_1, x_2, x_3, x_4 : [P(x_1) \wedge S(x_2) \wedge x_1 = x_2 \wedge x_2 = x_3 \wedge T(x_4) \wedge \\
 &\quad \neg \exists x_5, x_6, x_7 : (x_3 = x_6 \wedge x_6 = x_5 \wedge x_5 = x_7 \wedge x_7 = x_4 \wedge \neg R(x_6, x_7))] \\
 f_D &= \exists x_1, x_2 : [P(x_1) \wedge S(x_1) \wedge T(x_2) \wedge \neg (x_1 = x_2 \wedge \neg R(x_1, x_2))]
 \end{aligned}$$

Notes

- 1 ‘Sep’ is another word Peirce used for cut.
- 2 Both papers argue for a distinction between structures (types) and their mathematical representations (tokens), but in Dau (2004) it is argued that only the types need mathematical definitions, whereas Howse et al. (2002) propagates that this holds both for types and tokens.
- 3 In Dau (2006b), first subgraphs are defined on the type level, and a discussion on graphs and their diagrams yields that each subgraph can be represented by a subgraph-line in *some* diagram, and vice versa, each subgraph-line in a given diagram gives raise to a subgraph.

References

- Dau, F. (2002). An embedding of existential graphs into concept graphs with negations. In Priss, U., Corbett, D., and Angelova, G., editors, *ICCS*, volume 2393 of *LNAI*, pages 326–340, Borovets, Bulgaria, July, 15–19. Springer, Berlin – Heidelberg – New York.
- Dau, F. (2004). Types and tokens for logic with diagrams: A mathematical approach. In Wolff, K. E., Pfeiffer, H. D., and Delugach, H. S., editors, *Conceptual Structures at Work: 12th International Conference on Conceptual Structures*, volume 3127 of *Lecture Notes in Computer Science*, pages 62–93. Springer, Berlin – Heidelberg – New York.
- Dau, F. (2006a). Fixing shin’s reading algorithm for peirce’s existential graphs. In Barker-Plummer, D., Cox, R., and Swoboda, N., editors, *Diagrams*, volume 4045 of *LNAI*, pages 88–92. Springer, Berlin – Heidelberg – New York.
- Dau, F. (2006b). Mathematical logic with diagrams, based on the existential graphs of peirce. Habilitation thesis. To be published. Available at: <http://www.dr-dau.net>.
- Hartshorne, W. and Burks, editors (1931–1935). *Collected Papers of Charles Sanders Peirce*, Cambridge, Massachusetts. Harvard University Press.

Howse, J., Molina, F., Shin, S.-J., and Taylor, J. (2002). On diagram tokens and types. In Hegarty, M., Meyer, B., and Narayanan, N. H., editors, *Diagrams*, volume 2317 of *Lecture Notes in Computer Science*, pages 146–160. Springer, Berlin – Heidelberg – New York.

Pape, H. (1983). *Charles S. Peirce: Phänomen und Logik der Zeichen*. Suhrkamp Verlag Wissenschaft, Frankfurt am Main, Germany. German translation of Peirce's Syllabus of Certain Topics of Logic.

Peirce, C. S. (1931–1935). *MS 478: Existential Graphs*. Harvard University Press. Partly published in of Hartshorne and Burks (1935) (4.394-417). Complete german translation in Pape (1983).

Peirce, C. S. and Sowa, J. F. (1908, 2000). Existential Graphs: MS 514 by Charles Sanders Peirce with commentary by John Sowa. Available at: <http://www.jfsowa.com/peirce/ms514.htm>.

Roberts, D. D. (1973). *The Existential Graphs of Charles S. Peirce*. Mouton, The Hague, Paris.

Shin, S.-J. (2002). *The Iconic Logic of Peirce's Graphs*. Bradford Book, Massachusetts.

Zeman, J. J. (1964). *The Graphical Logic of C. S. Peirce*. PhD thesis, University of Chicago. Available at: <http://www.clas.ufl.edu/users/jzeman/>.

Frithjof Dau (b. 1968) is currently lecturer at the University of Wollongong, Australia. He finished his PhD-thesis 'The Logic System of Concept Graphs with Negations (And its Relationship to Predicate Logic)', which has been published by Springer in the series 'Lecture Notes in Artificial Intelligence', vol. LNCS 2892, in november 2003, and he plans to finish his habilitation thesis on Peirce's existential graphs in 2007. His main research interest is formalizing mathematical logic by means of diagrams, i.e., he is interested in formal languages which allow to visually represent and process information by means of mathematical lattices and graphs.