Institute for Language, Logic and Information

PEIRCE'S PROPOSITIONAL LOGIC: FROM ALGEBRA TO GRAPHS

Víctor Sánchez Valencia

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language LP-89-08

University of Amsterdam

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

PEIRCE'S PROPOSITIONAL LOGIC: FROM ALGEBRA TO GRAPHS

Víctor Sánchez Valencia
Department of Mathematics and Computer Science
University of Amsterdam

Received November 1989

Research supported by N.W.O.

PEIRCE'S PROPOSITIONAL LOGIC: FROM ALGEBRA TO GRAPHS

Summary

Both historic and systematic aspects of Peirce's logical work are studied in the present report. In the first part we consider the change in Peirce's treatment of logical calculi, starting from a Hilber type approach and culminating in a more representative one, the *system of existential graph* (SEG). The second part is devoted to a presentation of the propositional part of SE, and to the proof that this system has the same deductive strength as modern standard system. Nex, the report turns to the analysis of SEG from the perspective of transformational grammar. Subsequently, we pay attention to the semantic basis of these rules. In this part we see the pervasive notion of *monotonicity* playing a systematic and historical rol. In particular, we argue that Peirce's insertion and deletion operations are conceptually linked to the so called monotonicity rules known from formal semantic. Finally, in a concluding section we show that Peirce's logical work can be used in the construction of a *monotonicity calculus* akin to systems of *natural logic*.

Introduction

As we are often told, Peirce introduces in [Peirce 1885] 'a symbolism adequate for the whole of logic and identical in syntax with the systems now in use' [Kneale and Kneale 1962, p. 431]. Within this symbolism Peirce sets up a propositional system which is seen as the second successful axiomatization of propositional logic. In 1897 Peirce's lifelong interest in logic culminated in the development of the *system of existential graphs (SEG)*. Published interest in SEG dates back to [Gardner 1958], [Prior 1964]; the topic has subsequently been taken up by [Roberts 1973], [Thibaud 1975] and [Sowa 1984].

Whereas the system of 1885 constitutes a standard propositional calculus, the SEG has special properties of its own. In the first place, Peirce defines a *non-linear* propositional language with the same expressive power as usual linear languages. In the second place, the system embodies a few global inference rules which, seen from the perspective of standard systems, allow us to draw consequences from given premises without having to resolve them into smaller parts. Roughly speaking, the effect of Peirce's rules is the following:

- within any context we may introduce c.q. eliminate double negations;
- within specified syntactic positions we are allowed to insert c.q. to delete (occurrences of) formulas; and finally
- within certain syntactic configurations we are allowed to copy c.q. eliminate (occurrences of copied) formulas.

Peirce's richly diverse logical research contributed significantly to the revival of logic that took place at the end of the 19th century. Peirce's place in the history of logic justifies by itself the historical study of his logical theories and the assessment of the contemporary value these theories have.

In the present report we hope to show the importance of the study of SEG from two distinct perspectives.

- In the first place we assess the importance of SEG for the the history of modern logic. As we will point out, SEG constitutes Peirce's second successful treatment of classical logic. Furthermore, Peirce himself seems to have described SEG as his *chef d'oeuvre* , thus estimating SEG even more important than his 1885 contribution. These facts alone make SEG worth studying from a historical point of view. However, SEG itself constitutes the result of a historical development and we intend to point out some aspects of this process. In particular, we will see that there is a (literally) Fregean motivation behind the construction of the formal language of SEG. We will also see that, almost reminding Jevon's program of mechanical inference, Peirce sought to reduce inference to simple substitution procedures.

- In the second place we will point out the relevance of Peirce's global inference rules for the construction of *natural logics* (i.e. systems of inference, based directly on grammatical form). We noticed above that SEG embodies global inference rules different from the local rules of standard systems. This different approach to inference is worth studying on its own account. Furthermore, this view on inference is a feature SEG shares with systems of natural logic. These systems rely essentially on the notion of *monotonicity*. We intend to show that Peirce's deletion and insertion rules are based on this notion as well. Moreover, we will stress the fact that Peirce's copying rule is based on the semantical notion of conservativity - a principle not exploited yet in the construction of natural logics.

There remains, however, an aspect of SEG that we will not consider in the present report, although this aspect renders SEG an interesting system for modern readers. [Sowa 1984], a book devoted to cognitive science and artificial intelligence, claims that SEG is more adequate than standard systems for the representation of knowledge. In particular, Sowa takes SEG as the logical base for the construction of a theory of *conceptual graphs*, from the perspective of artificial intelligence research. Readers interested in this aspect of SEG are referred to Sowa's own work.

1. The 1885 system

1.1 The language and the icons

In the 1885 paper Peirce tries to develop a system "adequate for all the problems of deductive logic"[CP 3. 364][1]. In order to deal with these problems Peirce builds up an implicational language containing the following basic expressions:

- propositional letters: t, t', . . . ;
- implication symbol: $\rightarrow$;
- parentheses;

and a formation rule

- If x,y are propositions, so is $(x \rightarrow y)$.

In this language Peirce formulates a few primitive expressions which he calls *icons:*

(1) $x \rightarrow x;$
(2) $(x \rightarrow (y \rightarrow z)) \rightarrow (y \rightarrow (x \rightarrow z));$
(3) $(x \rightarrow y) \rightarrow ((y \rightarrow z) \rightarrow (x \rightarrow z));$
(5) $((x \rightarrow y) \rightarrow x) \rightarrow x.$

These primitive icons are supplemented with a derived one:

(6) $x \rightarrow (y \rightarrow x)$.

(Observe that the icons 2,5 and 6 form a sufficient base for the axiomatization of the pure implicational calculus).The fourth icon, intended to cope with negation, is introduced in a less conspicuous way;

> We must now again enlarge the notation so as to introduce negation.Let b be such that we can write b $\rightarrow$ x, whatever x may be. Then b is false.[CP. 3. 381

There is no unanimity among commentators with respect to the exact meaning of the fourth icon. There are at least three different interpretations of this passage. The most influential one is due to Prior[2] who supposes that Peirce is offering at the same time the definition $\neg x = x \to f$ and the icon $f \to x$. [Prior 1958, 135], while others just read either the definition or the icon.

It seems that Prior's view is the most plausible with regard to the spirit of Peirce's treatment of negation, although it is the least faithful to the letter of the cited passage. For instance, Peirce likes to stress that the principles of contradiction and of excluded middle are to be treated separately (which by the way he identifies with $\varphi \to \neg\neg\varphi$ and $\neg\neg\varphi \to \varphi$ respectively).[3]

By using Modus Ponens (MP) and the definition of negation we can derive $\varphi \to \neg\neg\varphi$ within the 1885 system:

> $(p \to f) \to (p \to f)$ is an icon. Hence by (2) and MP, $p \to ((p \to f) \to f)$. And by definition $p \to \neg\neg p$.

The so-called *Peirce's law* is introduced in order to obtain a negation-free equivalent of the principle of excluded middle:

> A fifth icon is required for the principle of excluded middle and other propositions connected with it. One of the simple formulae of this kind is $((x \to y) \to x) \to x$. This is hardly axiomatical. [CP. 3. 384]

But this icon by itself is not sufficient for the derivation of $\neg\neg p \to p$. As Prior points out, the definition of negation and Peirce's law are not sufficient: an additional axiom is required. [Prior 1962 p. 51]:

1.	$((f \to p) \to f) \to f$	Peirce's law
2.	$\neg\neg p \to f$	1 and definition of negation
3	$f \to ((f \to p) \to p)$	From icons 1,2 and MP
4.	$\neg\neg p \to ((f \to p) \to p)$	From 2,3 icon 3 and MP
5.	$(f \to p) \to (\neg\neg p \to p)$	From 4 , icon 2 and MP

By adding the ex falso rule to Peirce's system we get the desired formula.

1.2 Rules of proof

One intriguing aspect of Peirce's paper is the absence of *explicitly* formulated rules of proof.[4] Peirce states that if x is a true proposition we may write $y \to x$. This can be interpreted as stating

> R1: from $\vdash x$ follows $\vdash y \to x$.

But of course this rule cannot be sufficient. For instance, the proof Peirce gives of (6) reveals that other unstated rules are being used:

> "To say that $x \to x$ is generally true is to say that it is so in every state of things, say in that in which y is true; so that we may write $y \to (x \to x)$, and then by transposition of antecedents $x \to (y \to x)$ or from x we may infer $y \to x$. [CP. 3. 378]

Prior and Berry assert that Peirce's system presupposes both substitution and Modus Ponens. But the expressions used by Peirce have the status of schematic letters:

> The letters of applied algebra are usually tokens, but the x,y,z, etc of a general formula, such as$(x + y)z = + xz$ are blanks to be filled up with tokens, they are indices of tokens. [CP. 3. 364]

We think we are allowed to dispense with substitution: the letters Peirce uses are blanks to be filled with flesh and blood proposition letters.

Given $y \to (x \to x)$, we know that from (2) and Modus Ponens the desired icon follows. It is justifiable to add MP to Peirce's system. As the following passage shows, Peirce himself would have accepted the correction:

> the moment that a person who has made the incomplete inference "A therefore B" is led to reflect upon or criticize this procedure into the slightest degree, he will recognize the leading principle; If A then B as a premise and thus reform his inference as follows:
>
> If A then B
> But A
> Hence, B.
>
> This form of inference is called Modus Ponens. [CP. 4. 365]

This seems to settle the matter: we can act as if Peirce has forgotten to formulate explicitly one of his inference rules. Now we can reconstruct Peirce's proof of (6) in this way:

1.	$x \to x$	(1)
2.	$y \to (x \to x)$	From 1 by R1
3.	$x \to (y \to x)$	From 2 and the adequate choice of (2) by M.P.

Thus, by using (1), (2), R1 and MP Peirce derives (6). The latter, (3) and (5) form an adequate basis for the axiomatization of the implicational calculus. If we add to these principles the principle *ex falso sequitur quodlibet*, then Peirce has offered a basis for the axiomatization of classical propositional logic.[5]

1.3 Rules of inference

However, there is another possible reconstruction of Peirce's proof. We could say that Peirce does not use (2) as a step but invokes it in the same way that we invoke Modus Ponens. The most dramatic explanation would be the suggestion that Peirce is systematically confused between a *logical consequence* and a *material implication* interpretation of his implication symbol. Or, alternatively, that he intended his symbol to be read ambiguously.[6]These kinds of considerations may arise in connection with the reading Peirce gives of (6) : $y \to x$ follows from x. But there is a more sanguine (c.q. conservative) course of action. [Peirce 1885] is a revised version of [Peirce 1881]. In this earlier paper we read that

> the two inferences
>
> x
> y and x
> $\therefore$ z $\therefore y \to z$
>
> are of the same validity.[CP. 3. 182]

So Peirce states $x,y \vdash z \Leftrightarrow x \vdash y \to z$, a kind of postulated Deduction Theorem. [7] By applying this *theorem* to the primitive icons we obtain new inference rules which are then used in derivations.

1.4 The Future

Having built his system, Peirce disposes of it by pointing out that "the general formulae given above are not convenient in practice" and then proceeds to sketch several more flexible approaches. One of these consists of the acceptance of $x \to y$ as necessarily true if it is not possible to find a counter-example for it, i.e a valuation making x true and y false [CP. 3.387- 391]

Peirce further demolished the i885 system by allowing (multiplication), disjunction(addition) and negation, instead of material implication, as primitive operations. By adding the equations $x \wedge \neg x = \mathbf{f}$; $x \vee \neg x = \mathbf{t}$; $x \wedge \mathbf{t} = x$ and $x \vee \mathbf{f} = x$ to the classical algebraic properties of the Boolean operators, we could say that a formula x is necessarily true if it resolves into **t**.

To treat problems of deductive logic, Peirce introduces another device:

> To any proposition we have the right to add any expression at pleasure;also to strike out any factor of any term. The expressions for different propositions separately now may be multiplied together.[CP. 3. 391]

These last two rules are the basis of Peirce's future theory of deduction. Eventually he would assert that

> "There must be operations of transformation. In order that these operations should be as analytically represented as possible, each elementary operation should be either an insertion or an omission. Operations of commutation, like xy ∴ yx may be dispensed with by not recognizing any order or arrangement as significant. Associative transformations, like (xy)z ∴ x(yz) . . . will be dispensed in the same way." [CP. 4. 374]

The system in which this ideal was achieved is the theme of the next fewpages: *the alpha graphs*. In this system the adding and striking out operations did indeed become insertion and omission rules.

1.5 Qualitative Logic and monotonicity

[Roberts 1973] and [Thibaud 1976] describe how the alpha graphs system grew from Peirce's interest in diagrammatic representations of logical expressions. In this section we make a distinct connection between the later graphs and Peirce's early work. This connection has to do with Peirce's interest in inference rules. Here we will see the first appearance of a syntactic version of monotonicity rules expressed in a formal language.

The missing link between the graphs and the 1885 system is the so-called *Qualitative Logic*, a calculus probably set up in the early nineties. The language of this system contains a special symbol for material implication(*illation*) formed from the disjunction symbol joined to a negation line (*streamer*):

$$\overline{\;\;+}$$

> Thus 'if a, then b', would be written
>
> $\overline{a\!+}\, b$
>
> [Peirce 1976 IV p. 108]

After the reader has gone through the trouble to learn yet another notation, Peirce shows how this new symbol can be represented in the standard Boolean frame:

> we first separate the streamer of the sign of illation and in place of
>
> $\overline{a\!+}\, b$
>
> write ¬a ∨ b.[Peirce 1976 IV p. 114]

So let us make use of the old symbols. The system also contains two constants, one for *verum* and one for *falsum* and admits the blank as a legal expression. We will use **t** and **f** for verum and falsum respectively; the blank is taken as an equivalent of **f**.

Peirce's next step is the construction of a formal calculus.[8] One of the rules expresses the icon missing in the 1885 paper. It also gives an independent definition of negation by using **f**:[9]

> For **f** which is the symbol of falsity, we have the general formula **f** ∨ a whatever a may be. The falsity of a, usually written ¬a is really equivalent to ¬a ∨ **f**. [1976 IV, p. 109]

Here we have a passage which inclines the balance in favour of Prior's interpretation of the 1885 system.

There is another theme from that earlier paper finding its way intoo this system. Peirce opens his treatment of the calculus by saying that the inference rules must be instrumental:[10]

> to prove the two propositions *If ¬a ∨ b, then if a then b*; and *If from a follows b, then ¬a ∨ b*.[1976 IV, p. 107]

And there are also links with the latter systems:

> we require that the rules should enable us to dispense with all reasoningin our proofs except the mere substitution of particular expressions in general formulae. [1976 IV, p. 108]

But, after all, Modus Ponens is taken as a primitive inference rule. Then Peirce proceeds to generalize Modus Ponens in the first of three extra rules he formulates:

> The general rule of substitution is that if $\neg a \vee b$, then b may be substituted for a under an even number of negations, while under an odd number a may be substituted for b.[1976 IV, p. 108]

The link with his final approach is clearer in one of the extra rules which Peirce derives :

> **f** may be substituted for any term under an odd number of negations, and **t** for any term under an even number.[1976 IV, p. 110]

These passages are an unambiguous formulation of the so-called *(syntactic) monotonicity* rules. In modern format, these rules can be captured in the following way:

- $\varphi \rightarrow \psi, F(\varphi) \vdash F(\psi)$ where $F(\varphi)$ is any formula containing the subformula φ positively and $F(\psi)$ results from it by replacing this occurrence of φ by ψ.

- $\varphi \rightarrow \psi, F(\psi) \vdash F(\varphi)$ where $F(\psi)$ is any formula containing the subformula ψ negatively and $F(\varphi)$ results from it by replacing this occurrence of ψ by φ.

Peirce's above cited passage constitutes the first formal expression of these rules we have been able to find. With the benefit of hindsight (which the present gives about the past), we see that Peirce did not have a long way to go to arrive at his later deletion/insertion rules. At the end of 1.4 we quoted two passages from Peirce's work. In the first one, from the standard period, we mentioned rules allowing the elimination of conjuncts and the introduction of disjunctions. In the second one, from the graph period, we quoted the ideal of admitting only insertion and deletion. From the transition period we saw a passage which states that the rules should be reduced to operations admitting substitutions.These passages show a certain continuity in Peirce's thoughts about inference rules. In particular we can see that the later rules grew from the monotonicity rules.The connection between Peirce's insertion/deletion rules and the monotone rules is not only systematic but historical as well.

Digression

It is worth knowing that the syntactic monotonicity rules have a role in the further development of logic. These are for example *Dictum de Omni* in [Sommers 1982, pp.184],the *Semisubstitutivity of Conditional Rules* in [Zeman 1967 p. 484], and *Theorem 24* in[Kleene 1967,p.124]. Even some years earlier, the rules were mentioned in [Kleene 1952,p. 154.]. In this last book we are referred to [Curry's 1939,pp. 290-91] for another version of the rules. Curry, in turn, refers us to [Herbrand 1930 § 3.2][11] and [Maclane 1934].[12] Finally, the first post-Fregean reference to the rules which we have found is [Behmann 1922,pp. 172-174].

In general, these versions of the monotonicity rules diverge in two important ways. There is the weak version in which the premises of the rules are provable formulas (Behmann,Curry, Herbrand, Kleene 1967, Zeman). There is also the strong version in which the premises are assumptions (Kleene 1952, Maclane, Sommers). Furthermore, there is a version in which the substitution affects all the occurrences of φ in $F(\varphi)$ (Herbrand, Maclane) and another in which the substitution affects only a specified occurence of φ (Behmann, Curry, Kleene, 1952, 1967, Sommers, Zeman).

Semantic monotonicity appears in [Lyndon 1959] as a property of first-order sentences A generalization beyond formal languages is provided by [van Benthem 1986].

1.6 Remarks

In the above pages we have pointed out some features of Peirce's logic by using a modern terminology. This kind of anachronism is very tempting. Successful theories, like Borges' successful writers, create their own past. This self-made past may or may not coincide with the real past. Likewise, Peirce's paper has been seen as belonging to a tradition leading to the standardization of logic in the hands of Hilbert's School. Nevertheless the question arises whether or not we are entitled to attribute to him the full-fledged notion of the concepts involved in this choice of terminology. For instance, we have noted that

- he sets up an axiomatic system yielding pure implicational logic,
- he used schematic letters in the formulation of his axioms,
- he distinguished several principles governing the behaviour of negation,
- he has some awareness of the deduction theorem or he has formulated introduction and elimination rules for the implication symbol.

And of course the most sensible thing to do is to question these remarks In diffuse form and scattered through the pages and the years all these ideas are present in Peirce's writings. We can see these ideas in them because we know where to look and what to expect. It is the future in the hands of, say, Bernays and Gentzen which makes of this part of the logical past a source of anticipations.

2. The Alpha Graphs

2.1 Preliminaries

Peirce sets up the SEG with the intention of giving "a satisfactory logical analysis of the reasoning in mathematics". [CP.4. 424]. Part of this project consists of the construction of an adequate artificial language in which proofs and inference principles can be represented. The following remarks due to Peirce communicates the same spirit permeating Frege's project:

> What is requisite is to take really typical mathematical demonstrations, and state each of them in full, with perfect accuracy, so as not to skip any step, and then to state the principle of each step so as perfectly to define it, yet making this principle as general as possible... If we attempt to make the statement in ordinary language, success is practically impossible... At all times, the burden of language is felt severely, and leaves the mind with no energy for its main work. It is necessary to devise a system of expression for the purpose which shall be competent to express any proposition whatever without being embarrassed by its complexity, which shall be absolutely free from ambiguity, perfectly regular in its syntax, free from all disturbing suggestions, and come as nearer to a clear skeleton diagram of that element of the fact which is pertinent to the reasoning as possible. [NM. 3. 406]

The alpha graphs form the propositional basis of Peirce's system. The propositional language uses the blank area on which we write as a symbol for a specific object, namely *verum*. As symbol for negation it employs an oval around the proposition to be negated and it uses juxtaposition to symbolize conjunction. We will see shortly that as a notational variant of propositional logic the alpha graphs are not perspicuous.[13] Their utility lies somewhere else: within this language Peirce formulates a few global inference rules which, supplemented with the blank (or t as symbol for verum) as axiom, yield full propositional logic.

At this point a word of caution is in order. We are *not* asserting that Peirce himself axiomatized propositional logic (on the basis of the blank and his global rules). [14]We *do* assert that it can be shown that the alpha graphs contain the basis required for such an axiomatization. As a matter of fact, it seems more probable that in the construction of the SEG, Peirce is primarily concerned with the relation of *formal deducibility* between a set of assumptions and an end formula.[15]

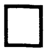
For convenience's sake, in this second part, we will not always follow Peirce's terminology or notation. The reader interested in the original formulation of the graphs is referred to [Peirce 1933 IV] or to the richer presentation found in [Roberts 1973].

2.1 The language

We use the following language:

- Proposition-letters: $P_0, P_1, \ldots$
- A sheet of assertion :

taken as a graph representing verum in the sense that each blank part of it stands for a true expression.

- A box intended to denote the negation of any expression it encloses: 

In general we omit the explicit mention of the sheet of assertion considering the areas on which we write as our temporary sheet, and we will refer to the blank by the expression **t**.

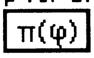
Graphs:

- Each propositional letter P is a graph.
- **t** is a graph.
- If φ is a graph, so is 

- If φ_1 and φ_2 are graphs so is $\varphi_1 \varphi_2$.

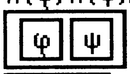
The "translation" from the usual languages into Peirce graphs is straightforward:

$\pi(p)$ = p for all propositional letters.

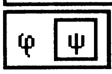
$\pi(\neg\varphi)$ = 

.

$\pi(\varphi \wedge \psi)$ = $\pi(\varphi)\pi(\psi)$.

$\pi(\varphi \vee \psi)$ = 

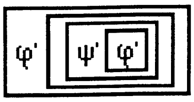
.

$\pi(\varphi \rightarrow \psi)$ = 

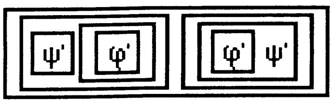
.

As a tool for immediate use we list translations of some well-known formulas, using φ' as an abbreviation of $\pi(\varphi)$:

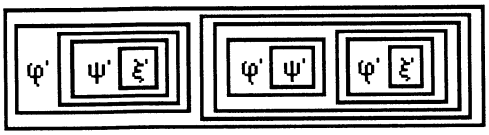
1. $\pi(\varphi \rightarrow (\psi \rightarrow \varphi))$=



2. $\pi(\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$ =



3. $\pi(\varphi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)) =$



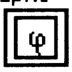
2.3 The calculus

Axiom

1. The blank is an axiom

Rules of inference

1. Within an even number of boxes any graph φ may be deleted. (Deletion Rule DR)[16]
2. Within an odd number of boxes any graph φ may be inserted. (Insertion Rule IR)[17]
3. The graphs

   φ and 

   are mutually interchangeable. (Double Negation Rule DN)[18]
4. The graphs $\varphi(\ldots\psi\ldots)$; $\varphi(\ldots\varphi\psi/\psi\ldots)$ are mutually interchangeable. (Copying Rule CR)[19]
5. $$\frac{\varphi \qquad\qquad \psi}{\varphi\psi}$$
   (Conjunction Rule CNR)[20]
6. The graphs $\varphi\psi$ and $\psi\varphi$ are mutually interchangeable. (Generalized Commutativity Rule GCR)[20]

• *Examples*

The following inferences are examples in standard notation of uses of DR,IR and CR respectively:

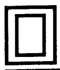
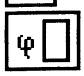
(1) $\varphi_1 \wedge \varphi_0 \vdash \varphi_i$ (i= 1,0).
(2) $\neg\varphi \vdash \neg(\varphi \wedge \psi)$.
(3) $\varphi \wedge \neg\psi \vdash \varphi \wedge \neg(\varphi \wedge \psi)$; $\varphi \wedge \neg(\varphi \wedge \psi) \vdash \varphi \wedge \neg\psi$

Next, we define the notions of *proof* in SEG , *theorem* of SEG and *deduction* in SEG in the following form:

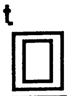
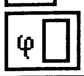
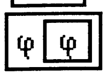
- A *proof* in SEG is a sequence $\varphi_1, \ldots \varphi_n$ of graphs such that, for each φ_i, either φ_i is t or φ_i is a direct consequence of some of the precedings graphs by virtue of one of the above rules or φ_i has the form $\varphi_j \varphi_k$ for some $j,k < i$.
- A *theorem* of SEG is a graph φ such that there is a proof with $\varphi_n = \varphi$.
- A *deduction* of φ from $\{\psi_1, \ldots, \psi_m\}$ in SEG [Notation: $\psi_1, \ldots, \psi_m \vdash \varphi$] is a sequence $\xi_1, \ldots, \xi_n$ of graphs such that $\varphi = \xi_n$ and for each ξ_i, either ξ_i is t or ξ_i is in $\{\psi_1, \ldots, \psi_m\}$ or ξ_i is a direct consequence of some of the precedings graphs by virtue of one of the above rules or ξ_i has the form $\xi_j \xi_k$ for some $j,k < i$.

- *Examples*

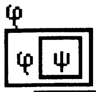
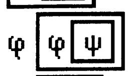
- $\vdash \pi(\varphi \rightarrow t)$

1. t — axiom
2.  — DN
3.  — IR

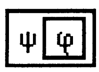
- $\vdash \pi(\varphi \rightarrow \varphi)$

1. t — axiom
2.  — DN
3.  — IR
4.  — CR (applied with ψ = blank)

- $\varphi, \pi(\varphi \rightarrow \psi) \vdash \psi$

1. φ — Assumption
2.  — Assumption
3.  — CNR
4.  — CR
5.  — DR
6. ψ — DN

Notice that this example proves that the alpha graphs are closed under Modus Ponens. We can also prove a kind of "implication introduction" in SEG:

$\psi \vdash \varphi \Rightarrow \vdash$ 

Proof

Let $\xi_1, \ldots \xi_n$ be a deduction of φ from ψ. We show that $\vdash \boxed{\psi \boxed{\xi_i}}$ for $1 \leqslant i \leqslant n$. First of all, ξ_1 must be φ or t. Our two first examples take care of these cases.

Assume now that $\vdash \boxed{\psi \boxed{\xi_j}}$ for all $j < i$. Either ξ_i is φ, or ξ_i is t, or ξ_i follows by R1-R4 from some ξ_k, or ξ_i has the form $\xi_k \xi_m$ for some ξ_k, ξ_m with $k,m < j$. In the first two cases, $\vdash \boxed{\psi \boxed{\xi_i}}$, as in the case i =1.

In the third case, by inductive hypothesis, $\vdash \boxed{\psi \boxed{\xi_k}}$. Hence, by applying the rule that transforms ξ_k into ξ_j once again, $\vdash \boxed{\psi \boxed{\xi_i}}$ [21].

In the last case, we have by hypothesis:

1. $\vdash \boxed{\psi \boxed{\xi_k}}$
2. $\vdash \boxed{\psi \boxed{\xi_m}}$ — Then,
3. $\vdash \boxed{\psi \boxed{\xi_m}} \quad \boxed{\psi \boxed{\xi_k}}$ — R5.
4. $\vdash \boxed{\psi \boxed{\xi_m}} \quad \boxed{\psi \boxed{\xi_k \boxed{\psi \boxed{\xi_m}}}}$ — CR
5. $\vdash \boxed{\psi \boxed{\xi_k \boxed{\psi \boxed{\xi_m}}}}$ — DR
6. $\vdash \boxed{\psi \boxed{\xi_k \boxed{\boxed{\xi_m}}}}$ — CR
7. $\vdash \boxed{\psi \boxed{\xi_k \xi_m}}$ — DN.

This completes the proof. The case i = n constitutes the desired result.

2.4 Deductive strength of the alpha graphs

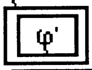
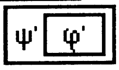
In this subsection we derive some theorems which are needed to prove the following assertion : The alpha graphs have the same deductive strength as standard propositional logic.[22] It suffices to prove

- $\varphi' \vdash (\psi \rightarrow \varphi)'$
- $(\neg \psi \rightarrow \neg \varphi)' \vdash (\varphi \rightarrow \psi)'$
- $(\varphi \rightarrow (\psi \rightarrow \xi))' \vdash ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)))'$

since we have already shown that the system is closed under Modus Ponens and that implication introduction is valid for SEG.

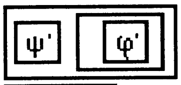
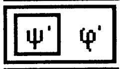
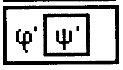
• $\varphi' \vdash (\psi \rightarrow \varphi)'$

Proof

1. φ' Assumption
2.  DN
3.  IR

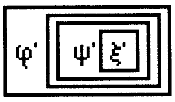
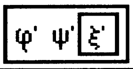
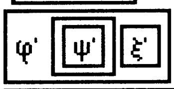
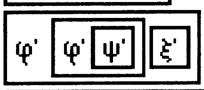
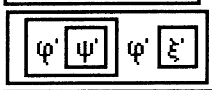
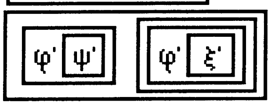
T3 $(\neg\psi \rightarrow \neg\varphi)' \vdash (\varphi \rightarrow \psi)'$

Proof

1.  Assumption
2.  DN
3.  GCR

T4 $(\varphi \rightarrow (\psi \rightarrow \xi))' \vdash ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi))'$

Proof

1.  Assumption
2.  DN
3.  DN
4.  CR
5.  GCR
6.  DN

3. The Alpha Rules as Transformations

In the preceding paragraph we have formulated Peirce's representation system for propositional inferences. A natural question in this regard concerns the complexity of the alpha representation of inferences, vis-á-vis the complexity of inferences as codified in standard systems. Partially, this is a question of linguistic nature. To investigate this matter more closely, we formulate the alpha rules in the formalism of transformational grammars, and compare inferences in this new setting with inferences in Hilbert-type systems. In this section we claim that in Peirce's system inference is to be seen as a (literally) "transformational process" in the linguistic sense. In order to show this, we will develop the necessary technical means (the reader may skip details at first reading).

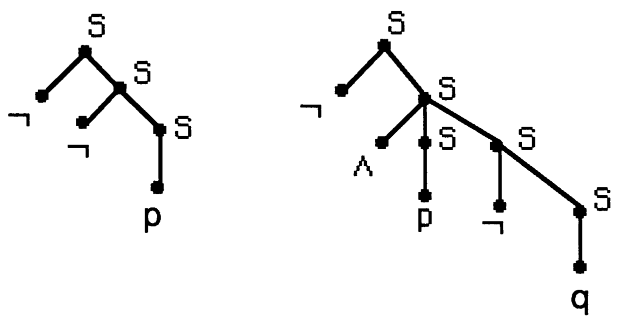
Let us start by reminding the reader of the most relevant aspects of (standard) transformational grammars. A grammar of this type consists of two parts. In the first place, a context-free or context-sensitive part yielding phrase-markers usually represented as parsing -trees (PT). In the second place, transformational rules which are mappings of PT's into PT's. Each such rule contains two parts : a structural analysis(SA) and a structural change(SC) which tells us what the resulting PT is. The structural analysis is used to determine whether or not the rule applies to a given input. If the rule will apply to a given input α, then the SA tells us which segments of α will be deleted, expanded or rearranged.

3.1 The base

As base of the system we use a context-free grammar generating the propositional formulation in Polish notation

1. $S \Rightarrow \wedge SS$
2. $S \Rightarrow \neg S$
3. $S \Rightarrow$ atomic formula (t, p, p', p'', . . .)

With each of the expressions generated by this grammar we associate a PT.
Below we give the PT of the formulas $\neg\neg p$, $\neg \wedge p \neg q$:



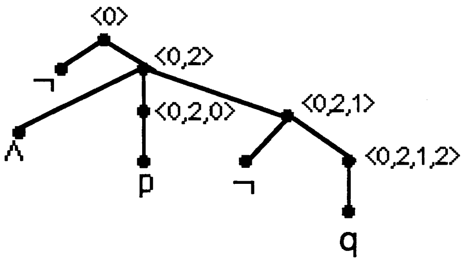
3.2 Coding and polarity

Observe that in this format, given a formula φ with PT α, the sub-formulas of φ are all the formulas generated by the sub-trees with tope node labeled S. Taking the PT of φ as stepping-stone, we can derive a code for all the sub-formulas of φ.

- Given a formula φ and its PT α, we code the nodes labeled S as follows:

(1) The top node is given the code $\langle 0 \rangle$
(2) If a node labeled S has code x and branches into the nodes labeled $\neg$ and S, then this new S is given the code $\langle x,2 \rangle$.
(3) If a node labeled S has code x and branches into the nodes labeled $\wedge$, S and S, then these new S-nodes get the codes $\langle x,0 \rangle$, $\langle x,1 \rangle$ respectively.

Example



- A node labeled S in a coded PT is called positive if the code of S contains an even number of 2's. Otherwise it will be called negative.
- A sub-formula ψ of a formula φ is called positive (negative) if the node labeled S which yields ψ is positive (negative).

Example

According to the above coded PT, the positive sub-formulas of ¬∧p¬q are the formula itself and q, whereas the negative sub-formulas are ∧p¬q, p, ¬q.

3.3 Deletion and insertion in the Polish notation

It is evident that in this perspective, the deletion rule can not be applied without a further proviso. If we delete a positive sub-formula the resulting string may not correspond to a well formed formula. For instance, by deleting the positive q in ∧pq, we obtain ∧p. The reason is, of course, that in the Polish notation the blank is not a legal expression. To incorporate Peirce's deletion rule into our standard languages, we need to decompose this rule into two parts. In the first place, we allow every positive sub-formula to be replaced by the proposition **t**. In the second place, we make explicit the algebraic properties of verum: ∧p**t** is interchangeable with p. Hence, given ∧pq we derive ∧**t**q in the first place and then q. Thus deletion becomes a combination of algebra and the rule: **t** may be substituted for any positive formula.By the same token, insertion can not be applied without further proviso. Thus, we can't pass from ¬p into ¬pq, since this string does not belong to the Polish language. Once again, we decompose this global rule into an algebraic and a substitutional part. The algebraic part remains the same as before. The substitutional part, consists of the mirror image of the deletion rule: any formula may replace a given negative occurrence of **t**. Thus, we derive ¬∧p**t** by algebra and substitute q for **t**, obtaining ¬∧pq.

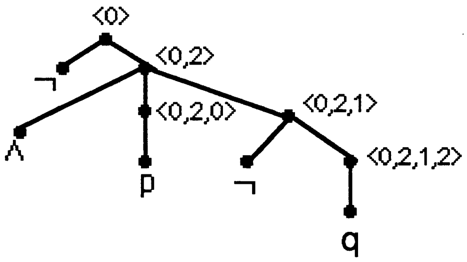
3.4 Dominance and c-commanding

We say that each node β in a PT α directly dominates itself and if β branches into the nodes $\beta_1, \ldots, \beta_n$ then β directly dominates each β_i. Furthermore, if β directly dominates γ then β dominates γ and if β dominates γ and γ dominates ξ, then β dominates ξ as well.

Now we define the node relation *c-commanding* and c-domain[23]:

> (i)A node β c-commands a node γ iff β does not dominate γ and the first branching node that directly dominates β, dominates γ.(ii)The c-domain of a given S-node β consists of all the S-nodes γ such that β c-commands γ.

example

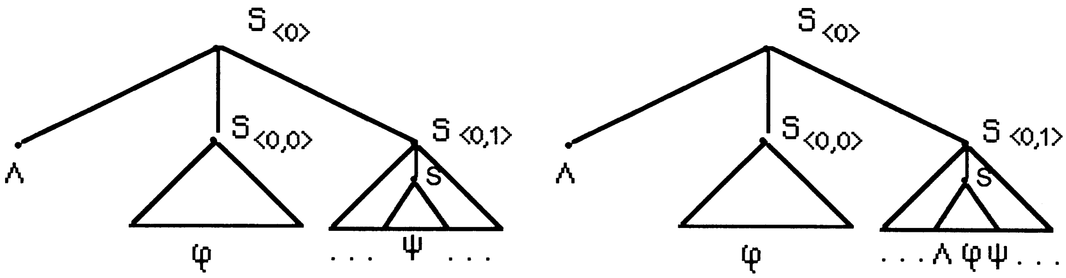


The node <0,2,0> c-command the nodes <0,2,1>, <0,2,1,2> since <0,2>, the first branching node dominating <0,2,0>, dominates the last two S-nodes. On the other hand, <0,2> does not c-command any of the lowest S-nodes since it dominates them. And <0,2,1,2> does not c-command <0,2,0> since the first branching node that dominates this node fail to dominate <0,2,0>.The c-domain of the nodes <0>, <0,2>, <0,2,1,2> is the empty set.The c-domain of <0,2,0> is {<0,2,1>, <0,2,1,2>} and the c-domain of <0,2,1> is {<0,2,0>}

3.5 c-commanding and the copying rule

In this language the copying rule can be formulated as follows:

$\wedge\varphi(\ldots\psi\ldots) \Leftrightarrow \wedge\varphi(\ldots\wedge\varphi\psi\ldots)$. The abbreviated PT of these formulas are



Notice now that in both trees the nodes dominated by the node <0,1> belong to the c-domain of the node generating φ; in particular the node generating ψ and $\wedge\varphi\psi$ belongs to this domain. Thus we can view this rule as the permission to copy or eliminate a given formula inside the c-domain of its top S- node.

3.6 Transformation rules

The SD of a transformation is a sequence of terms $T_1,\ldots,T_n$. Each T_1 is a node symbol or a variable (X_i) taking values onthe set of strings of node symbols. To see whether a given transformation applies to a PT α, we examine whether the yield of α can be factorized into a sequence of strings $\alpha_1,\ldots,\alpha_n$ such that for terms in T_i which are node symbols, the corresponding substring α_i can be traced back to a node with the given label. For terms that are variables, any string (including *e*, the empty string) can satisfy the analysis.

Example

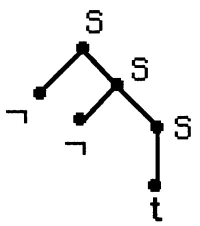
• $X_1 \wedge SSX_2$ is a structural description since it is a sequence of variables and node symbols. Given now the PT of the formula $\neg \wedge p \neg t$, we can see that this marker satisfies the analysis since it can be factorized into the strings $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ where $\neg = \alpha_1$, $\wedge = \alpha_2$, $p = \alpha_3$, $\neg t = \alpha_4$, $e = \alpha_5$. Furthermore, $\wedge$ can be traced back to the node labeled $\wedge$, and p and $\neg t$ can both be traced back to nodes labeled S.

Occasionally it is necessary to state further restrictions a PT must meet in addition to the restrictions specified in the SD. In particular, we will find necessary to restrict some transformations to those nodes which have a positive or negative code. For instance, we state the SD $X_1SX_2SX_3$ and add the condition: the third member must have a positive(negative) code. Furthermore, we will find it necessary to restrict some transformations to those strings in which a node c-commands some other nodes. For instance, we state the SD $X_1X_2X_3$ and add the condition: X_2 must c-command X_3.

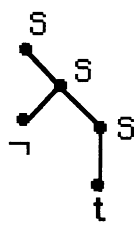
Before describing the SC part of the transformation rules, let us list some of the operations which can transform one PT into another.

(1)We can delete a node α of a PT. In this case we remove everything dominated by α and everything that directly dominates it.

• For instance given the PT

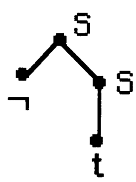


we can delete the highest ¬- node obtaining in this way

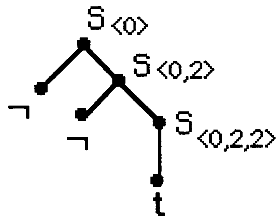


Now the product of the deletion is not a PT. But we assume that if a node α does not branch, then the node below it must have a different label. Each PT not of this form is to be identified with the PT obtained by identifying such pairs.

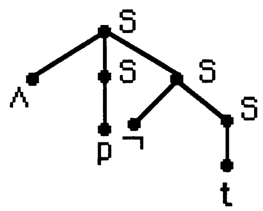
Thus, by this standard convention the above tree is identified with the PT:



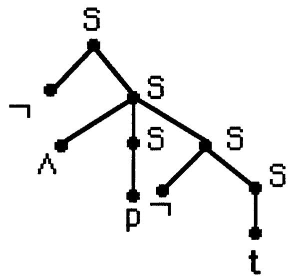
(2) We can replace a part of a PT with a PT. Thus given the following coded PT:



we can replace the node <0,2> and everything dominated by it with the PT



obtaining in this way



The SC part of the transformational rules consists of a sequence of natural numbers counting the members of the SD followed by the symbol "⇛". Under the numbers we describe the transformations as follows:

(i) If there is no change, the number is repeated.
(ii) If there is a deletion, we put the symbol e under the number corresponding to the term that must be deleted.
(iii) If there is a replacement, we put the bracketed yield of the PT corresponding to the new term under the number of the term we want to replace.

Examples

SC: 1 2 3 4 ⇛
1 e e 4

This description indicates that in the PT with a SD of length 4, the second and third member must be deleted.

SC: 1 2 3 ⇒
1 [∧φ2] 3

This description indicates that in the PT with with a SD of length 3, the second member must be replaced by a PT that yields ∧φ2 where 2 is the P-marker corresponding to the second member of the original tree.

3.7 Peirce rules

1. Deletion

SD: X_1 S X_2 ; Condition: 3 is positive.
SC: 1 2 3 ⇒
1 t 3

2. Insertion

SD: X_1 t X_2 ; Condition: 3 is negative.
SC: 1 2 3 ⇒
1 [φ] 3

3. Double Negation

SD: X_1 S X_2
SC: 1 2 3 ⇒
1 [¬¬2] 3

SD: X_1 ¬ ¬ S X_2
SC: 1 2 3 4 5 ⇒
1 e e 4 5

4.1 Copying I

SD: X_1 S X_2 S X_3 ; Condition: 2 c-commands 4
SC: 1 2 3 4 5 ⇒
1 2 3 [∧24] 5

4.2 Copying II

SD: X_1 S X_2 ∧ S S X_3; Condition: 2 c-commands 5 and 2 = 5.

SC: 1 2 3 4 5 6 7 ⇒
1 2 3 e e 6 7

5. **t**-rules

SD: X_1 S X_2
SC: 1 2 3 ⇒
1 [∧t2] 3

SD: X_1 ∧ **t** S X_2
SC: 1 2 3 4 5 ⇒
1 e e 4 5

3.7 Derivations with Peirce's rules and derivations in standard systems

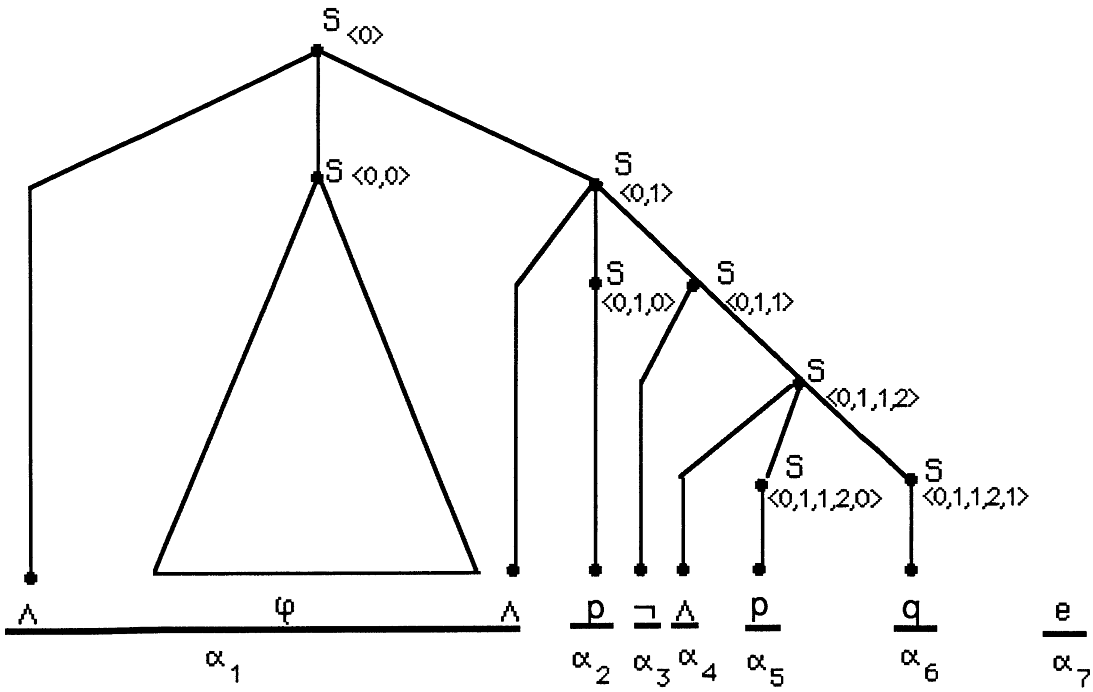
Now we shalll compare inferences in this new setting with inferences in Hilbert type systems.We will say that the formula φ follows from $\varphi_1, \ldots, \varphi_n$ if there is a sequence $P_1, \ldots, P_k$ of PT such that each P_i is the PT of one of the φ_i's or follows from P_j $(j< i)$ in virtue of one of the transformation rules or is the PT of the form $\wedge P_j P_m$ ($j,m< i$) and P_k is the PT of φ.

As we pointed out earlier on, Peirce's rules allow us to draw consequences from given premises without having to resolve them into smaller parts. Sometimes there are no mayor differences between an alpha inference and a standard one. But a typical example of an interesting case in which the systems behave differently, is the following:

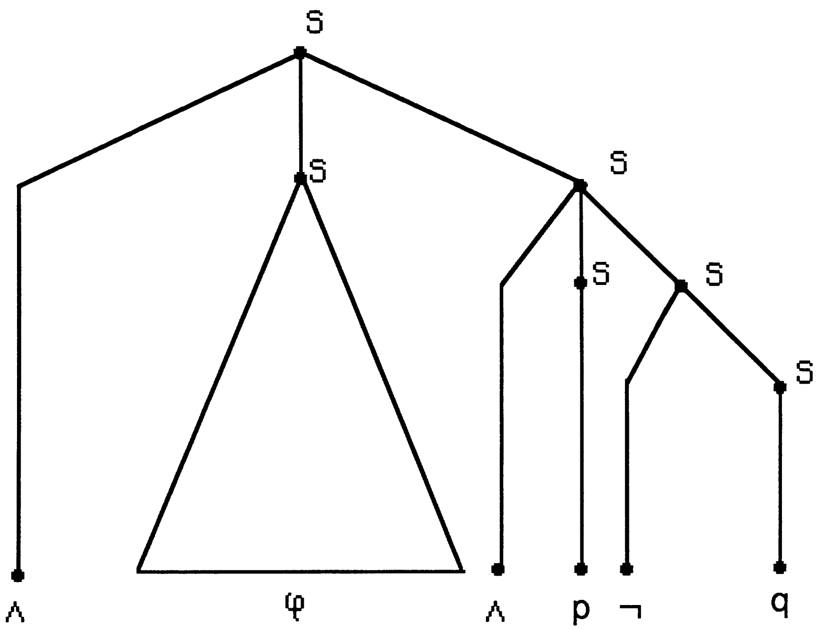
$$\wedge\varphi\wedge p\neg\wedge pq \vdash \wedge\varphi\wedge p\neg q.$$

In our system the proof take this form:

- ∧φ∧p¬∧pq can be factorized in the substrings $\alpha_1 = \wedge\varphi\wedge$, $\alpha_2 = p$, $\alpha_3 = \neg$, $\alpha_4 = \wedge$, $\alpha_5 = p$, $\alpha_6 = q$ and $\alpha_7 = e$. Furthermore, in the PT of this formula, the S node to which α_2 can be traced back, c-commands the nodes to which α_5 and α_6 can be traced back. This means than our formula satisfies the **SA** of 4.2. Hence, by the SC of this rule, we derive a PT that yields ∧φ∧p¬q. Thus we have the one step derivation from



to



- On the other hand, a proof in a Hilbert type system will take more inferential steps, since we need to resolve the premise into small parts:

1.	$\rightarrow\wedge\varphi\wedge p\neg\wedge pq\wedge p\neg\wedge pq$	Theorem
2.	$\wedge\varphi\wedge p\neg\wedge pq$	Premise

3.	$\wedge p\neg\wedge pq$	MP 1,2
4.	$\rightarrow\wedge p\neg\wedge pqp$	Theorem
5.	p	MP 3,4
6.	$\rightarrow\wedge p\neg\wedge pq\neg\wedge pq$	Theorem
7.	$\neg\wedge pq$	MP 3,6
8.	$\rightarrow\neg\wedge pq\rightarrow p\neg q$	Theorem
9.	$\rightarrow p\neg q$	MP 7,8
10.	$\neg q$	MP 5,9
11.	$\rightarrow p\rightarrow\neg q\wedge p\neg q$	Theorem
12.	$\rightarrow\neg q\wedge p\neg q$	MP 5,11
13.	$\wedge p\neg q$	MP 10,12
14.	$\rightarrow\wedge\varphi\wedge p\neg\wedge pq\varphi$	Theorem
15.	φ	MP 14,2
16.	$\rightarrow\varphi\rightarrow\wedge p\neg q\ \wedge\varphi\wedge p\neg q$	Theorem
17.	$\rightarrow\wedge p\neg q\ \wedge\varphi\wedge p\neg q$	MP 15,16
18.	$\wedge\varphi\wedge p\neg q$	MP 13,17

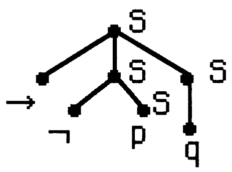
We can see now that the complexity of Peirce's rules pay off in the sense that more syntactic analysis allow us to dispose of piecemeal decomposition of premises into smaller parts. However, Peirce himself says that SEG is "not intended as a calculus, or apparatus by which conclusions can be reached and problems solved with greater facility that by more familiar systems of expressions". [CP. 4. 424].

Roughly speaking, this lack of computational ease is partially due to the relative *complexity* of the global rules. Peirce's global rules do not apply to formulas simpliciter but to formulas cum syntactic analysis.This is not the case in usual formulations of propositional logic. For instance, in ordinary propositional logic, in order to check whether φ,ψ,ξ is a proof of ξ from φ and ψ by MP, it suffices to look with the naked eye at the external form of the formulas, i.e. we test this sequence by the criteria: ψ and ξ are wffs and $\varphi = \psi \rightarrow \xi$. No further analysis of the formulas is required.[24] Nevertheless, the complexity of Peirce's rules remains manageable. The same procedure used to determine whether a string constitutes a formula or not, can be used to code the syntactic information needed as input of the global rules.

Notice by the way that first-order logic has rules demanding a syntactic analysis of given formulas. A typical example is the familiar rule " $\forall x\varphi \rightarrow \psi$ is a axiom if ψ is like φ except for containing y free wherever φ has x free." In this case it is not enough to check whether φ and ψ are wff's. We need to look at the syntactical analysis of the formulas, at their parsing tree or formation sequence, and to check whether the binding patterns satisfies the description. And in this respect this rule resembles the transformational rules of generative grammars and the global rules of the existential graphs.

3.8 Remarks

Let us conclude this part by pointing out one important drawback of the system. The system we have been working is rests essentially based on two connectives. Of course, we can introduce other logical operators using the usual definitions. But then the global rules would need to be handled with care. For instance, q c-commands p in



But $\rightarrow\neg pq$ does not entail $\rightarrow\neg\wedge pqq$. The fact is that in the official tree of the former formula, q corresponds to a node that does not c-command the node corresponding to p.

Similarly, even if q occurs positively in $\vee pq$, we can not delete this formula to obtain p as a result. All we get is $\vee pt$ and this expression, as expected, cannot be simplified into p. The moral of this tiny example is that deletion, in the strict sense, concerns positive *conjuncts* and not *positive formulas* in general. Similarly, the insertion rule allows insertion of *conjuncts* in negative contexts and not of arbitrary formulas.Thus, even if we extend the language, we have to take care that the global rules apply only to formulas in primitive notation.[25] As a matter of fact, Peirce's copying rule reveals an important syntactical property of $\wedge$ that it does not share with its mirror image $\vee$. On the other hand, the substitutional part of the deletion rule is a particularization of Peirce's monotonicity rules we saw before:

> The general rule of substitution . . .that if $\neg a \vee b$, then b may be substituted for a under an even number of negations, while under an odd number a may be substituted for b.[1976 IV, p. 108]

In this particularization t plays the role of b and the formula $\vee\neg at$ is implicitly assumed.

4. Monotonicity, Conservativity and Peirce's Rules

In this section we will be concerned with the logical analysis of the two global logical principles embodied in SEG:

- monotonicity
- conservativity

and we will consider briefly the principle of *convexity*[26],closely related to monotonicity:

These notions have been useful in describing and explaining linguistic phenomena; cf. [Barwise and Cooper 1981], [Zwarts 1986], [Keenan and Falz 1985],[van Benthem 1986], [Westerståhl 1986].

In this part we show that Peirce's propositional rules can be incorporated into a monotonicity calculus (i.e. a calculus containing the so-called *monotonicity rules*). It has been observed that many propositional inferences revolve around monotonicity. A monotonicity calculus gives a direct explanation of interesting propositional inferences. Examples are Modus Ponens: $\varphi \rightarrow \psi, \varphi \vdash \psi$; Modus Tollens $\varphi \rightarrow \psi, \neg\psi \vdash \neg\varphi$; Syllogism: $\varphi \rightarrow \psi, \psi \rightarrow \xi \vdash \varphi \rightarrow \xi$.

Nevertheless a simple monotonicity calculus does not completely match the inferential strength of propositional logic. Although many propositional inferences rest *only* on monotonicity, not all of them do. Examples are inferences which are based on the algebraic properties of the connectives involved and inferences such as $p \rightarrow q \,/\, p \rightarrow (p \wedge q)$. But in many of these cases, there are tautological formulas which bridge the gap that seems to preclude the use of monotonicity. Thus in the last inference $p \rightarrow p \wedge p$ constitutes the desired tautological bridge:

$$\frac{p \rightarrow q \qquad p \rightarrow p \wedge p}{p \rightarrow (p \wedge q)}$$

An interesting feature of Peirce's graphs is that we can see which principles have to be added to the monotonicity calculus in order to obtain the full strength of classical logic. We will argue that a simple monotonicity calculus supplemented with an adequate version of the so-called *principle of conservativity* yields classical propositional logic, modulo two tautologies and double negation rules.

As a by-product of this approach we will be able to show that Peirce's system is sound: his deletion and insertion rules are special cases of monotonicity rules. Hence the soundness of the latter implies the soundness of the former. Furthermore, we will see that the copying rule is a special case of the provably sound principle of conservativity.

4.1 Monotonicity

In the field of formal semantics the notion of monotonicity is used primarily to characterize the stable behaviour of certain *mathematical objects* with respect to replacements in their arguments. Thus, given a non-empty set D and a relation R on the power set of D, we call R *upward monotone* in its right argument if for any $A,B,C \subseteq D$, $R(A,B), B \subseteq C \Rightarrow R(A,C)$. Similarly, we call R *downward monotone* in its right argument if $R(A,B), C \subseteq B \Rightarrow R(A,C)$. This persistency of R with respect to the "increase" or "decrease" of its right argument can of course be present on its left argument. For instance, in formal semantics we will say that the binary quantifier ALL is *upward (monotone)* in its right and *downward (monotone*)in its left argument. [27]

But there are at least two interrelated ways in which we assign monotonicity properties to *syntactical objects* as well: specific lexical items and designated positions within natural language sentences are also called monotone:

- We say, for instance, that the determiner *X* is right/left, upward/downward monotone if its denotation, the quantifier X, is. Thus the determiners *all, each, every* will be described as upward with respect to their VP and downward with respect to their N. [28]

- On the other hand we can also say that expressions occurring in the environment of determiners occur in *inferentially ensitive positions* and we can describe these positions in terms of monotonicity. Thus we say that the N in a sentence of the form *Every N VP* occurs in downward monotone *position* and that the VP therein occurs in upward monotone *position* (more briefly: *upward*). [29]

The characterization of monotone positions is generalized in the following two inference rules present in van Benthem 1986:

- $[\![X]\!] \leqslant [\![Y]\!], \ldots X \ldots \Rightarrow \ldots Y \ldots$
  if X is upward in ... X ...

- $[\![X]\!] \leqslant [\![Y]\!], \ldots Y \ldots \Rightarrow \ldots X \ldots$
  if Y is downward in ... Y ...

In this connexion the question arises : which syntactic conditions on . . .X. . . guaranties the upward(downward) monotonicity of X. For first-order languages the answer to this question is contained in Lyndon 1959. In this paper Lyndon defines quite explicitly the notion of upward monotone (which he calls "increasing") first-order sentences.[30] Essentially, he defines a first-order sentence $\varphi(R)$ as upward monotone in the predicate R^n whenever

- $\forall x_1 \ldots x_n (R^n (x_1 \ldots x_n) \rightarrow S^n (x_1 \ldots x_n)), \varphi(R) \vDash \varphi(S)$,
  where $\varphi(S)$ is the result of replacing the predicate R by S in $\varphi(R)$.

Subsequently, Lyndon proves that there is a syntactic condition on φ and R that implies upward monotonicity of φ with respect to R . Fix a first-order language with only $\exists, \forall, \wedge, \vee, \neg$.We say that a sentence φ is positive in R if every occurrence of R in φ is positive (i.e. occurs under an even number of negation symbols).

This syntactic characterization motivates the following proposition in Lyndon 1959:

- If φ is positive in R, then φ is increasing in R.

Thus, in the context of formal language of first-order logic, positiveness implies upward monotonicity. Similarly, negativeness implies downward monotonicity. However, the *direct* converse does not hold, since for instance $\varphi = p \wedge \neg p$ is upward monotone in p but φ is not positive in p. [31]

However, the counting of negations is a coarse criteria for monotonicity, even in the context of a first-order language with all the connectives present.. For instance in $p \rightarrow q$, the formula p is positive in the sense that it occurs under 0 negations but obviously it is not upward therein. On the other hand, as we pointed out, *man* in *Every man walks* counts as downward but there is no negation dominating this noun. As a matter of fact, the above characterization of positiveness is confined to formal languages with $\neg, \wedge$ and $\vee$ as the only primitives. Thus, if we wish to incorporate $\rightarrow$ into the list of primitive symbols, the definition of polarity would take this form:

- R is positive in R.
- If R is positive (negative) in φ, then R is positive(negative) in $\varphi \wedge \psi$, $\varphi \vee \psi$, $\psi \rightarrow \varphi$, $\forall x \varphi$.
- If R is positive (negative) in φ, then R is negative(positive) in $\neg \varphi$, $\varphi \rightarrow \psi$.

In this format, the name positive or negative loses the direct connotation of occurrence under negations. The idea that emerges is that of expressions occurring under the scope of symbols denoting operators with specific monotone properties. By itself, every expression X is upward However, embedded in a larger syntactical environment, X may retain this monotonicty, or have it changed into downward or lack any form of monotonicity altogether. Examples of the latter are the antecedent X in the conditional *ifXY*, or *women* in the expression *Most women read,* or the formula p in $p \leftrightarrow q$.

4.2 Monotonicity Rules and Alpha Graphs

Consider a propositional language L consisting of proposition letters, the propositional constant t (verum) and the logical constants $\neg, \wedge, \vee, \rightarrow$. Further, assume the standard semantic interpretation. Let Φ be a formula. Assume that Φ has been associated with a standard parsing tree α. We specified the subformulas of Φ by the following coding:

- Φ gets the code <0>.
  If $\neg \varphi$ is a sub-formula of Φ with code x, then φ gets the code <x,2>.
  If $\varphi \wedge \psi$ is a sub-formula of Φ with code x, then φ gets the code <x,0>and ψ the code<x,1>.
  If $\varphi \vee \psi$ is a sub-formula of Φ with code x, then φ gets the code <x,0> and ψ the code<x,1>
  If $\varphi \rightarrow \psi$ is a sub-formula of Φ with code x, then φ gets the code <x,2> and ψ the code <x,0>.

- Suppose that φ is a sub-formula of Φ and that a particular occurrence of φ in Φ has been specified by the code x (notation: φ_x). We call φ_x positive(negative) in Φ if x contains an even number of 2's. Otherwise it will be called negative.

- We denote Φ with a particular specified occurrence of φ in Φ, by the expression $\Phi(\varphi_x)$. The result of replacing this specified subformula φ in Φ by ψ is denoted by $\Phi(\psi_x)$.

Given the above coding, we express Peirce's general substitution rules as follows:

- $\varphi \rightarrow \psi, \Phi(\varphi_x) \vdash \Phi(\psi_x)$ if φ_x is positive in Φ.
- $\varphi \rightarrow \psi, \Phi(\psi_x) \vdash \Phi(\varphi_x)$ if φ_x is negative in Φ.

There are several ways in which we can prove that these rules are sound. The most direct one consists in resorting to monotonicity:

- φ_x is upward in $\Phi(\varphi_x)$:=
  $[\![\varphi]\!] \leqslant [\![\psi]\!], \Rightarrow \Phi(\varphi_x) \leqslant \Phi(\psi_x)$.
- φ_x is downward in $\Phi(\varphi_x)$:=
  $[\![\varphi]\!] \leqslant [\![\psi]\!] \Rightarrow \Phi(\psi_x) \leqslant \Phi(\varphi_x)$.

Consequently we can prove that the syntactic condition of polarity implies monotonicity:

- φ_x is positive/negative in $\Phi(\varphi_x) \Rightarrow \varphi_x$ is upward/downward in $\Phi(\varphi_x)$.

Proof
Induction on the complexity of $\Phi(\varphi_x)$.

Remark

Observe that the definition of upward differs from Lyndon's notion of *increasing*. Increasing is a property of sentences with respect to subformulas, whereas in the above definition upward is a property of sentences and subformula *occurrences*. Thus a given sentence Φ may have an upward subformula φ_x without being increasing in it. For example, $p \rightarrow (p \wedge q)$ is not increasing in p but $p_{<<0,1>,1>}$ is upward therein. On the other hand, if Φ is increasing in φ, then all the specified occurrences of φ are upward.

With these two definitions we can show that the soundness of Peirce's deletion and insertion operations revolve around monotonicity.

Suppose Φ is a formula in which φ_x occurs positively. Then φ_x is upward in Φ. Then, since for all φ, $\models \varphi \rightarrow t$, we have : $\Phi(\varphi_x) \models \Phi(t_x)$. Thus, in positive positions any formula may be deleted, in the sense that it may be replaced by t

Let now Φ be a formula in which t_x occurs negatively. Then t_x is downward in Φ. Hence, since for all φ, $\models \varphi \rightarrow t$ we have: $\Phi(t_x) \models \Phi(\varphi_x)$.Thus, in negative positions any formula may be introduced, in the sense that any formula may replace t occurring downward.

4. 3 Conservativity and the Copying Rule

We pointed out that although many propositional inferences rest directly on monotonicity, not all of them do. For instance $\varphi \rightarrow \psi \vdash \varphi \rightarrow (\varphi \wedge \psi)$ remains unaccounted for: ψ occurring in a positive position is replaced by a formula having more chances of being false than ψ itself In general, this substitution will not give correct results. It is φ, the formula copied into the consequent which keeps things straight. By the same token, monotonicity is not enough to distinguishes $\wedge$ from $\vee$: they denote upward monotone functions in both arguments.

The above copying of formulas can be accounted for by appealing to a version of the principle of *conservativity*, also known from logical semantics. The notion of conservativity was introduced by Keenan (1981) to characterize quantifiers. Given the putative assertion Q(A,B), the principle of conservativity is intended to account for the fact that with some quantifiers only the part of B common to A matters in determining whether Q(A,B) holds or not Hence a binary quantifier Q is called conservative, whenever Q(A,B) iff Q(A,A ∩B) is the case.[32]

Conservativity, like monotonicity, has been introduced into logical semantics with respect to the denotations of a specific linguistic category: determiners. We propose the following extension of the notion of conservativity to the category of connectives: Let Δ be a binary connective C,

- Δ is conservative if $[\![\psi\Delta\Phi(\varphi_x)]\!] = [\![\psi\Delta\Phi((\varphi \wedge \psi)_x)]\!]$

It is easy to see that conjunction and implication are conservative connectives. [33] For instance:

> $[\![\psi \wedge \Phi(\varphi_x)]\!] = 1 \Leftrightarrow [\![\psi]\!] = [\![\Phi(\varphi_x)]\!] = 1$. Now, if $[\![\varphi]\!] = 1$ or $[\![\varphi]\!] = 0$, $[\![\varphi\wedge\psi]\!] = [\![\varphi]\!]$. In both cases it follows that $[\![\psi \wedge \Phi(\varphi_x)]\!] = 1$. Otherwise, if $[\![\psi \wedge \Phi(\varphi_x)]\!] = 0$, then $[\![\psi]\!] = 0$ (and then we are through) or $[\![\Phi(\varphi_x)]\!] = 0$ and $[\![\psi]\!] = 1$. But then as above, whether φ is true or false, $[\![\varphi\wedge\psi]\!] = [\![\varphi]\!]$. Thus $[\![\psi \wedge \Phi(\varphi_x)]\!] = 0$.

Hence, Peirce's copying rules have conservativity of ∧ as their semantical basis; on this semantical property of conjunction rests the soundness of those rules.

Digression

There are weaker principles than monotonicity which have been used in the semantic study of generalized quantifiers. The notion of *Convexity* is a case in point. We say that a quantifier Q is convex in its first argument if QXY,QZY, $X \subseteq V \subseteq Z$ entails QVY. Redundantly, we say that a determiner D is convex in its first argument if DAB, DCB, $[\![A]\!] \subseteq [\![E]\!] \subseteq [\![C]\!]$ entails DEB. For instance, the determiner ONE, which lacks any form of monotonicity, is convex in its two arguments. Thus, from the sentences *One girl sings*, and *one person sings* it follows *one female sings*

This notion of convexity can be generalized to propositional languages. Let us call the connective Δ convex in its first argument if $\varphi\Delta\psi$, $\xi\Delta\psi$, $[\![\varphi]\!] \leqslant [\![\theta]\!] \leqslant [\![\xi]\!] \Rightarrow \theta\Delta\psi$. In a similar way we call the connective Δ convex in its second argument if the corresponding entailment holds. Thus we can easily see that ↔ and Sheffer's operator | are convex in both arguments although they are not monotone therein.

Let us call a first order sentence Φ convex in the unary predicate R if the following holds:

$\Phi(R)$, $\Phi(S)$, $\forall x(Rx \rightarrow Qx)$, $\forall x(Qx \rightarrow Sx) \vdash \Phi(Q)$, where $\Phi(S)$ and $\Phi(Q)$ are like $\Phi(R)$ except by having S, Q at exactly the same places $\Phi(R)$ has R. By using Peirce's rules we can show that $\Phi(R)$ is convex in R if it is equivalent to $\Phi_0(R) \wedge \Phi_1(R)$ where R is positive in $\Phi_0(R)$ and negative in $\Phi_1(R)$. Suppose that $\Phi(R) = \Phi_0(R) \wedge \Phi_1(R)$ and that Φ_0 is positive in R whereas Φ_1 is negative in R. Then, using using Peirce's monotonicity rules (and completeness), we show that this description of Φ constitutes a sufficient condition for its convexity in R.

$$\dfrac{\dfrac{\dfrac{\Phi(R)}{\Phi_0(R) \quad \forall x(Rx \rightarrow Qx)}}{\Phi_0(Q)}\text{Peirce's rule} \qquad \dfrac{\dfrac{\Phi(S)}{\Phi_1(S) \quad \forall x(Qx \rightarrow Sx)}}{\Phi_1(Q)}\text{Peirce's rule}}{\Phi(Q)}$$

4.4 A Monotonicity Calculus

We can ask which principles have to be added to the monotonicity rules in order to obtain a system as strong as classical propositional logic. We conclude this report by answering this question on the basis of the above considerations concerning Peirce's graphs as well as modern semantics.

Consider the propositional language L consisting of proposition-letters, the propositional constant **t** (verum) and the logical constants $\neg, \wedge$. Let $\rightarrow$ be defined as usual.

We define a (strengthened) monotonicity calculus as a system of inference rules consisting of:

(0)	$\vdash \varphi \rightarrow \mathbf{t}$
(1)	$\vdash \varphi_0 \wedge \varphi_1 \rightarrow \varphi_i \; (i = 0,1)$
(2)	if φ_x is positive in $\Phi(\varphi_x)$, then $\varphi \rightarrow \psi,\ \Phi(\varphi_x) \vdash \Phi(\psi_x)$.
(3)	if φ_x is negative in $\Phi(\varphi_x)$, then $\psi \rightarrow \varphi,\ \Phi(\varphi_x) \vdash \Phi(\psi_x)$.
(4)	$\ldots \varphi \,\Delta\, \Phi(\psi_x) \ldots \vdash \ldots \varphi \,\Delta\, \Phi((\varphi \wedge \psi)_x) \ldots$
(5)	$\ldots \varphi \,\Delta\, \Phi((\varphi \wedge \psi)_x) \ldots \vdash \ldots \varphi \,\Delta\, \Phi(\psi_x) \ldots$
	where Δ is $\wedge$ or $\rightarrow$.
(6)	$\ldots \neg\neg\varphi \ldots \vdash \ldots \varphi \ldots$
(7)	$\ldots \varphi \ldots \vdash \ldots \neg\neg\varphi \ldots$

Let us show the strength of this calculus by proving that $\wedge$ is commutative, associative and idempotent.

(8) $\vdash \varphi \rightarrow \varphi$

1.	$\varphi \rightarrow \mathbf{t}$	(0)
2.	$\varphi \rightarrow \varphi \wedge \mathbf{t}$	1 and (4)
3.	$\varphi \wedge \mathbf{t} \rightarrow \varphi$	(1)
4.	$\varphi \rightarrow \varphi$	2,3 and (2)

(9) $\vdash \varphi \rightarrow (\psi \rightarrow \varphi)$

1.	$\varphi \rightarrow \varphi$	(8)
2.	$\varphi \rightarrow \neg\neg\varphi$	1 and (7)
3.	$(\mathbf{t} \wedge \neg\varphi) \rightarrow \neg\varphi$	(1)
4.	$\varphi \rightarrow \neg(\mathbf{t} \wedge \neg\varphi)$	2,3 and (3)
5.	$\psi \rightarrow \mathbf{t}$	(0)
6.	$\varphi \rightarrow \neg(\psi \wedge \neg\varphi)$	4,5 and (3)
7.	$\varphi \rightarrow (\psi \rightarrow \varphi)$	6 and definition

(10) $\vdash \varphi \rightarrow (\psi \rightarrow (\psi \wedge \varphi))$

1.	$\varphi \rightarrow (\psi \rightarrow \varphi)$	(9)
2.	$\varphi \rightarrow (\psi \rightarrow (\psi \wedge \varphi))$	1 and (4)

(11) $\vdash \varphi \leftrightarrow \varphi \wedge \varphi$

1.	$\varphi \wedge \varphi \rightarrow \varphi$	(1)
2.	$\varphi \rightarrow \varphi$	(8)
3.	$\varphi \rightarrow \varphi \wedge \varphi$	(4)
4.	$\varphi \leftrightarrow \varphi \wedge \varphi$	from 1,2, (10) and definition.

(12) $\vdash (\varphi \wedge \psi) \rightarrow (\psi \wedge \varphi)$

1.	$(\varphi \wedge \psi) \rightarrow (\varphi \wedge \psi)$	(8)
2.	$(\varphi \wedge \psi) \rightarrow ((\varphi \wedge \psi) \wedge (\varphi \wedge \psi))$	1 and 4
3.	$(\varphi \wedge \psi) \rightarrow \psi$	(1)

4.	$(\varphi\wedge\psi)\rightarrow(\psi\wedge(\varphi\wedge\psi))$	2,3 and (2)
5.	$(\varphi\wedge\psi)\rightarrow\varphi$	(1)
6.	$(\varphi\wedge\psi)\rightarrow(\psi\wedge\varphi)$	4,5 and (2)

The other direction can be proved similarly. Hence, by using (10) and the definition of $\leftrightarrow$,

(13) $\vdash(\varphi\wedge\psi)\leftrightarrow(\psi\wedge\varphi)$

(14) $\vdash\varphi\wedge(\psi\wedge\xi)\rightarrow(\varphi\wedge\psi)\wedge\xi$

1.	$\varphi\wedge(\psi\wedge\xi)\rightarrow\varphi\wedge(\psi\wedge\xi)$	(8)
2.	$\varphi\wedge(\psi\wedge\xi)\rightarrow(\varphi\wedge(\psi\wedge\xi))\wedge(\varphi\wedge(\psi\wedge\xi))$	1 and (3)
3.	$(\psi\wedge\xi)\rightarrow\psi$	(1)
4.	$\varphi\wedge(\psi\wedge\xi)\rightarrow(\varphi\wedge\psi)\wedge(\varphi\wedge(\psi\wedge\xi))$	2,3 and (2)
5	$\varphi\wedge(\psi\wedge\xi)\rightarrow(\psi\wedge\xi)$	(1)
6.	$\varphi\wedge(\psi\wedge\xi)\rightarrow(\varphi\wedge\psi)\wedge(\psi\wedge\xi)$	4,5 and 2
7	$\psi\wedge\xi\rightarrow\xi$	(1)
8.	$\varphi\wedge(\psi\wedge\xi)\rightarrow(\varphi\wedge\psi)\wedge\xi$	6,7 and (2)

The other direction can be proved in similar way. Hence
(15) $\vdash\varphi\wedge(\psi\wedge\xi)\leftrightarrow(\varphi\wedge\psi)\wedge\xi$.

Finally, perhaps laboring the obvious,we show that the derivations of $(\neg\varphi\rightarrow\neg\psi)\rightarrow(\psi\rightarrow\varphi)$ and $(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow((\varphi\rightarrow\psi)\rightarrow(\varphi\rightarrow\xi))$can be carried out in this new setting

(11) $\vdash(\neg\varphi\rightarrow\neg\psi)\rightarrow(\psi\rightarrow\varphi)$

1.	$(\neg\varphi\rightarrow\neg\psi)\rightarrow(\neg\varphi\rightarrow\neg\psi)$	(8)
2.	$(\neg\varphi\rightarrow\neg\psi)\rightarrow\neg(\neg\varphi\wedge\neg\neg\psi)$	def
3.	$(\neg\varphi\rightarrow\neg\psi)\rightarrow\neg(\neg\varphi\wedge\psi)$	(6)
4.	$(\neg\varphi\rightarrow\neg\psi)\rightarrow(\psi\rightarrow\varphi)$	def

(12) $\vdash(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow((\varphi\rightarrow\psi)\rightarrow(\varphi\rightarrow\xi))$

1.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow(\varphi\rightarrow(\psi\rightarrow\xi))$	(8)
2.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow\neg(\varphi\wedge\neg\neg(\psi\wedge\neg\xi))$	def
3.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow\neg(\varphi\wedge(\psi\wedge\neg\xi))$	2, (6)
4.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow\neg(\varphi\wedge(\neg\neg\psi\wedge\neg\xi))$	3,(7)
5.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow\neg(\varphi\wedge(\neg(\varphi\wedge\neg\psi)\wedge\neg\xi))$	4,(3)
6.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow\neg(\neg(\varphi\wedge\neg\psi)\wedge(\varphi\wedge\neg\xi))$	5,(13),(15)
7.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow\neg(\neg(\varphi\wedge\neg\psi)\wedge\neg\neg(\varphi\wedge\neg\xi))$	6, (7)
8.	$(\varphi\rightarrow(\psi\rightarrow\xi))\rightarrow((\varphi\rightarrow\psi)\rightarrow(\varphi\rightarrow\xi))$	def

The above proofs show that this strengthened monotonicity calculus is equivalent to standard formulations of classical logic. In this regard, something can be said for considering this system as simply another formulation of classical logic. But we can think of at least two reasons in favour of this system. In the first place, its geography of inference principles is interesting, since it seems to harmonize better with current ideas in natural language semantics. And in the second place, this strengthened monotonicity calculus seems less ad hoc than the strengthened version given in [Sommers 1980]. In this latter work monotonicity - there called *distribution-* is supplemented with reductio, elimination and introduction rules for conjunction and algebraic rules.

Concluding remarks

In the last section of this report we have compared the propositional part of SEG with recent generalized quantifiers research. Prima facie this comparison is odd. We would rather expect a consideration of the predicate logic fragment of SEG in the generalized quantifiers perspective. However, the study of the predicate logic fragment of the SEG leads to the same conclusions. For reasons of space, details are omitted here. However, in a later publication we will consider some special themes arising from Peirce's treatment of predicate logic.

---

1 References to [Peirce 1931-38] will follow the standard convention of referring first to the series, then to the volume, and finally to the paragraph. For example [CP 3. 364] means [Peirce 1931-38], volume 3, paragraph 364.

2 The other two commentators are Quine and Berry. Quine sees Peirce as expressing in an ambiguous way the definition:

$\neg x =_{def} \Pi_y (x \rightarrow y)$ or $\neg x =_{def} x \rightarrow \Pi_y y$ [Quine 1935,pp. 291-292].

On the other hand Berry interprets Peirce as giving a definition of negation in terms of implication and falsum:

$$\neg x =_{def} x \rightarrow \mathbf{f}.$$ [Berry 1952 p. 157].

3 "... we have the immediate inference S is true. Hence, the denial of S is false. The statement of the validity of this general inference is termed the principle of contradiction. The converse of this principle, namely that from the denial of the denial of a proposition the truth of that proposition follows, is termed the principle of excluded middle. This principle constitutes a distinct principle concerning negation. " [NM. 4. 373]

4 Peirce's system falls short of constituting a formalization of propositional logic even if we interpret his icons as axioms. In the absence of proof rules no proof gets off the ground. Similarly, no deduction of a conclusion from premisses can take place since, officially, we are in the dark about which steps to this end are legal ones.

5 Later on we will see that the adding of the ex falso principle to the other icons is historically justifiable.

6 This last is the interpretation proposed in [Dipert 1981].

7 Alternatively, the $\Rightarrow$ direction can been seen as an implication introduction rule as in natural deduction systems.The converse direction, together with Icon 1, can be used to derive an implication elimination rule:

• $(x\rightarrow y) \rightarrow (x\rightarrow y) \Leftrightarrow x\rightarrow y \vdash x\rightarrow y \Leftrightarrow x\rightarrow y, x \vdash y$.

8 "So far, we have a language but still no algebra. For an algebra is a language with a code of formal rules for the transformation of expressions, by which we are enabled to draw conclusions without the trouble of attending to the meaning of the language we use." Peirce [1976 IV, p. 107]

9 The third rule contains the application of the truth-table method to test validity. We skip it out, as we do not havae anything interesting to say about it.

10 Notice that now the connexion between $x \vdash y$ and $\vdash x \rightarrow y$ is not postulated but stated as something which needs proof.

11 Herbrand 1971 p. 78.

12 Maclane 1979 pp. 28-30

13 Peirce was well aware of this lack of perspicuity, writing that his "system is not intended to serve as a universal language for mathematicians or other reasoners, like that of Peano." [CP. 3. 424]. There is an apparent contradiction between this remark and the description of the aims of the

SEG quoted above. However, we can consider the SEG as a *first* attempt to achieve the above mentioned goals, an attempt open to further modifications.

14 The first published proof of this assertion is to be found in [Roberts 1973].

15 " Part II will develop formal 'rules' or permissions, by which one graph may be transformed into another without danger of passing from truth to falsity and without recurring to any interpretation of the graphs; such transformations being of the nature of immediate inferences." [Peirce IV, paragraph 423]

16 Within an even finite number (including none) of seps, any graph may be erased; within an odd number any graph may be inserted.

[Peirce 1933, IV paragraph 492]

17 See footnote 17

18 "Anything can have double enclosures added or taken away, provided there be nothing within one enclosure but outside the other." [CP.4. 379]

19 "Any graph may be iterated within the same or additional seps, or if iterated, a replica may be erased, if the erasure leaves another outside the same or additional seps."

[CP. 4 .492].

In the development of the system an expression of the form ... φ ... is to be understood as standing for an arbitrary graph with some particular occurrence of φ. The expression ... ψ/φ ... is to be understood as the result of replacing that particular occurrence of φ by ψ.

20 As a matter of fact, this rule does *not* belong to Peirce's formulation of the system. In his system writing the graph φ at one point and the graph ψ at another means nothing else than $\varphi \wedge \psi$, whenever they are enclosed by the same number of parentheses. Separated introduction of premises is not possible. Hence, CNR could not even be expressed in his system. On the other hand, in Peirce's original system commutativity is an implicit rule.

21 Notice that rules R3 and R4 allow for the substitution of an expression by another in any context.The other two rules ask for information about the polarity of some expressions . But each positive[negative] graph of ξ_j is positive[negative] in $\boxed{\psi \boxed{\xi_j}}$.

22 We show that SEG is essentially equivalent with the system P_2 of [Church 1956].

23 For the linguistic motivation and use of these notions see [Reinhart 1983].

24 Nevertheless, it can be shown that Hilbert type proofs considered as a language containing the propositional language plus a separation marker between proofsteps, is not context free. For instance, in order to recognize whether or not φ,ψ,ξ form a MP-sequent, we need more powerful machines than the usual push-down automata. In fact, a two-way push down automaton would do the job.

25 Of course, we can prove the *usual* inference rules governing these new connectives, but this is beside the point.

26 In the literature this notion is called "continuity".

27 This means that ALL (A,B) and B $\emptyset$ C entails ALL (A,C) (upwards monotonicity) and ALL (A,B) and C$\emptyset$ A entails ALL (C,B) (downwards monotonicity).

28 See for instance [Zwarts 1987] for a linguistic use of this characterization of expressions.

29 Note that this characterization intentionally suggests that given a sentence of the form *Every N_1 VP* and the additional information $[\![N_2]\!] \leqslant [\![N_1]\!]$ we may conclude that Every N_2 VP is the case. However, a determiner is only one of the factors that determine the monotonicity of given positions within a sentence: the monotonicity effects of grammatical construction rules and of other categories of lexical items have to be taken into consideration.

---

30 Lyndon 1959 pp.144-145.

31 Up to equivalence, however, Lyndon proves a converse of the above proposition:

> If φ is increasing in R, then φ is equivalent to a sentence ψ that is positive in R.

It is worth noticing that deciding whether a sentence φ is monotone in R is very difficult. [Gurevich 1984] proves that its is undecidable whether φ is upward monotone in R. On the other hand, deciding whether a sentenceφ is positive in a predicate letter is obviously decidable.

32 We say for example that the quantifier MOST is conservative, since MOST(MEN,LAUGH) iff MOST(MEN, LAUGH $\cap$ MEN).

33 Naturally disjunction and co-implication are not: $[\![\varphi \vee \psi]\!] \neq [\![\varphi \vee (\varphi \wedge \psi)]\!][\![$; $[\![\varphi \leftrightarrow \psi]\!] \neq [\![\varphi \leftrightarrow (\varphi \wedge \psi)]\!]$

REFERENCES

Barwise and Cooper 1981

Generalized Quantifiers and Natural Language", J. Barwise and R. Cooper, **Linguistics and Philosophy** 4, 159-219.

Behmann 1922

"Beiträge zur Algebra der Logik insbesondere zum Entscheidungsproblem", H. Behmann, **Mathematischen Annalen** vol. 86, 162-229.

Berry 1952

"Peirce's contributions to the logic of statements and quantifiers" in **Studies in the Philosophy of Charles Sanders Peirce**, ed. P.P. Wiener and F.H. Young.

Church 1956

Introduction to mathematical logic, vol.I. A. Church, Princenton,N.J. (Princenton Univ. Press).

Curry 1939

" A note on the reduction of Gentzen's calculus LJ", H. Curry, **Bulletin of the American Mathematical Society**, vol. 45, pp. 288-93

Dipert 1981

"Peirce's propositional logic", R. Dipert, **Review of Methaphysics**, 34, 569-595.

Herbrand 1930

" Investigations in Proof Theory", J. Herbrand, reprinted in [Herbrand 1971].

Herbrand 1971

Logical Writings, ed. W. D. Golfarb, Dordrecht-Holland.

Gardner 1958

Logic, Machines and Diagrams, M. Gardner New-York.

Gurevich 1983

"Toward logic tailored for computational complexity" in **Computation and Proof Theory** (M.M. Richter et al., Eds) pp. 225-251. Springer Lecture Notes in Mathematics 1104.

Keenan and Falz 1985

Boolean Semantics for Natural Language, Keenan and Falz Reidel Dordrecht.

Kneale & Kneale, M. 1962

The Development of Logic, Kneale,W. and Kneale, M. Clarendon Press, Oxford.

Kleene 1952

Introduction to Metamathematics, S.C. Kleene, North-Holland.

Kleene 1967

Mathematical Logic, S.C. Kleene, New-York.

Lyndon 1959

"Properties preserved under Homomorphim", R.C. Lyndon in
Pacific Journal of Mathematics, 9, pp.142-154, 1959.

Maclane 1934

Abgekürzte Beweise im Logikkalkul, S. Maclane, reprinted in [Maclane 1979]
Maclane 1970

Maclane 1979

Collected Papers, S. Maclane 1979, Springer, Berlin.

Reinhart 1979

"Syntactic Domains for Semantic Rules", in **Formal Semantics and Pragmatics for Natural Languages**, (eds) Guenthner and Schmidt. Reidel, Dordrecht.

Peirce 1880

"On the Algebra of Logic", C.S. Peirce, reprinted in[Peirce 1933 III].

Peirce 1885

"On the Algebra of Logic: A Contribution to the Philosophy
of Notation", C.S. Peirce, reprinted in [Peirce 1933 III].

Peirce 1933

Collected Papers of Charles Sanders Peirce, Volumes III-IV, ed. C.
Hartshorne & P. Weiss, Cambridge Harvard University Press.

Peirce 1976

The New Elements of Mathematics by Charles. S. Peirce
i-iv, ed. C. Eisele, Mouton,The Hague-Paris.

Prior 1958

"Peirce's axioms for Propositional Calculus", A.N. Prior, **Journal of Symbolic Logic**, Vol. 22. pp. 135-6

Prior 1962

Formal Logic, A. N. Prior. Oxford University Press.
First edition 1955.

Prior 1967

"Peirce", A. N. Prior, **The Encyclopedia of Philosophy**, New York, Vol. 4. pp.

Quine 1935

"Review of [Peirce 1933]", W. O. Quine, **ISIS** Vol XXII, pp. 285-97.

Roberts 1973

The Existential Graphs of Charles S. Peirce, D.R. Roberts, Mouton, The Hague-Paris.

Sowa 1984

Conceptual Structures, J.F. Sowa, Reading, Massachusetts.

Thibaud 1975

La logique de Charles Sanders Peirce, P. Thibaud, Provence.

Westerstahl 1986

"Quantifiers in Formal and Natural Languages", D.Westerstahl
Center for the Study of Language and Information. Report NO. CSLI-86-55.

Zeman1967

"A System of Implicit Quantification", J.J. Zeman, **Journal of Symbolic Logic**, vol. 32 pp. 480-505.

Zwarts 1986

Categoriale Grammatica en Algebraische Semantiek,
F. Zwarts, Groningen (Dissertation)

The ITLI Prepublication Series

1986

86-01		The Institute of Language, Logic and Information
86-02	Peter van Emde Boas	A Semantical Model for Integration and Modularization of Rules
86-03	Johan van Benthem	Categorial Grammar and Lambda Calculus
86-04	Reinhard Muskens	A Relational Formulation of the Theory of Types
86-05	Kenneth A. Bowen, Dick de Jongh	Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06	Johan van Benthem	Logical Syntax

1987

87-01	Jeroen Groenendijk, Martin Stokhof	Type shifting Rules and the Semantics of Interrogatives
87-02	Renate Bartsch	Frame Representations and Discourse Representations
87-03	Jan Willem Klop, Roel de Vrijer	Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04	Johan van Benthem	Polyadic quantifiers
87-05	Víctor Sánchez Valencia	Traditional Logicians and de Morgan's Example
87-06	Eleonore Oversteegen	Temporal Adverbials in the Two Track Theory of Time
87-07	Johan van Benthem	Categorial Grammar and Type Theory
87-08	Renate Bartsch	The Construction of Properties under Perspectives
87-09	Herman Hendriks	Type Change in Semantics: The Scope of Quantification and Coordination

1988

Logic, Semantics and Philosophy of Language:

LP-88-01	Michiel van Lambalgen	Algorithmic Information Theory
LP-88-02	Yde Venema	Expressiveness and Completeness of an Interval Tense Logic
LP-88-03		Year Report 1987
LP-88-04	Reinhard Muskens	Going partial in Montague Grammar
LP-88-05	Johan van Benthem	Logical Constants across Varying Types
LP-88-06	Johan van Benthem	Semantic Parallels in Natural Language and Computation
LP-88-07	Renate Bartsch	Tenses, Aspects, and their Scopes in Discourse
LP-88-08	Jeroen Groenendijk, Martin Stokhof	Context and Information in Dynamic Semantics
LP-88-09	Theo M.V. Janssen	A mathematical model for the CAT framework of Eurotra
LP-88-10	Anneke Kleppe	A Blissymbolics Translation Program

Mathematical Logic and Foundations:

ML-88-01	Jaap van Oosten	Lifschitz' Realizabiility
ML-88-02	M.D.G. Swaen	The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03	Dick de Jongh, Frank Veltman	Provability Logics for Relative Interpretability
ML-88-04	A.S. Troelstra	On the Early History of Intuitionistic Logic
ML-88-05	A.S. Troelstra	Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:

CT-88-01	Ming Li, Paul M.B.Vitanyi	Two Decades of Applied Kolmogorov Complexity
CT-88-02	Michiel H.M. Smid	General Lower Bounds for the Partitioning of Range Trees
CT-88-03	Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas	Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04	Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette	Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05	Peter van Emde Boas	Machine Models and Simulations (revised version)
CT-88-06	Michiel H.M. Smid	A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07	Johan van Benthem	Time, Logic and Computation
CT-88-08	Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas	Multiple Representations of Dynamic Data Structures
CT-88-09	Theo M.V. Janssen	Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10	Edith Spaan, Leen Torenvliet, Peter van Emde Boas	Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11	Sieger van Denneheuvel, Peter van Emde Boas	Towards implementing RL

Other prepublications:

X-88-01	Marc Jumelet	On Solovay's Completeness Theorem

1989

Logic, Semantics and Philosophy of Language:

LP-89-01	Johan van Benthem	The Fine-Structure of Categorial Semantics
LP-89-02	Jeroen Groenendijk, Martin Stokhof	Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03	Yde Venema	Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04	Johan van Benthem	Language in Action
LP-89-05	Johan van Benthem	Modal Logic as a Theory of Information
LP-89-06	Andreja Prijatelj	Intensional Lambek Calculi: Theory and Application
LP-89-07	Heinrich Wansing	The Adequacy Problem for Sequential Propositional Logic
LP-89-08	Víctor Sánchez Valencia	Peirce's Propositional Logic: From Algebra to Graphs

Mathematical Logic and Foundations:

ML-89-01	Dick de Jongh, Albert Visser	Explicit Fixed Points for Interpretability Logic
ML-89-02	Roel de Vrijer	Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03	Dick de Jongh, Franco Montagna	Rosser Orderings and Free Variables
ML-89-04	Dick de Jongh, Marc Jumelet, Franco Montagna	On the Proof of Solovay's Theorem
ML-89-05	Rineke Verbrugge	Σ-completeness and Bounded Arithmetic
ML-89-06	Michiel van Lambalgen	The Axiomatization of Randomness
ML-89-07	Dirk Roorda	Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08	Dirk Roorda	Investigations into Classical Linear Logic
ML-89-09	Alessandra Carbone	Provable Fixed points in $I\Delta_0+\Omega_1$

Computation and Complexity Theory:

CT-89-01	Michiel H.M. Smid	Dynamic Deferred Data Structures
CT-89-02	Peter van Emde Boas	Machine Models and Simulations
CT-89-03	Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas	On Space Efficient Simulations
CT-89-04	Harry Buhrman, Leen Torenvliet	A Comparison of Reductions on Nondeterministic Space
CT-89-05	Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree	A Parallel Functional Implementation of Range Queries
CT-89-06	H.W. Lenstra, Jr.	Finding Isomorphisms between Finite Fields
CT-89-07	Ming Li, Paul M.B. Vitanyi	A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08	Harry Buhrman, Steven Homer, Leen Torenvliet	Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09	Harry Buhrman, Edith Spaan, Leen Torenvliet	On Adaptive Resource Bounded Computations

Other Prepublications:

X-89-01	Marianne Kalsbeek	An Orey Sentence for Predicative Arithmetic
X-89-02	G. Wagemakers	New Foundations: a Survey of Quine's Set Theory
X-89-03	A.S. Troelstra	Index of the Heyting Nachlass
X-89-04	Jeroen Groenendijk, Martin Stokhof	Dynamic Montague Grammar, a first sketch
X-89-05	Maarten de Rijke	The Modal Theory of Inequality