Towards thought as a logical picture of signs

J.J. Sarbo, S.J.B.A. Hoppenbrouwers, J.I. Farkas

Computing Science Institute/

CSI-R0119 September 2001

Computing Science Institute Nijmegen
Faculty of Mathematics and Informatics
Catholic University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands

On the isomorphism of sign, logic and language

A novel framework for language modelling

Janos J. Sarbo and József I. Farkas
University of Nijmegen, The Netherlands

Abstract. Formal models of natural language often suffer from their excessive complexity which, in our opinion, may be due to the underlying approach itself. In this paper we introduce a semiotic model of language which is only *linearly* complex. The existence of such a model is conform with the experience that language is a real-time complex phenomenon.

Keywords: semiotic, logic, syntax

Introduction

Formal models of natural language often suffer from their excessive complexity which, in our opinion, may be due to their underlying approach to language. Indeed, their formal character –nomen est omen– implies that they are doomed to reflect what is 'natural' in language in an ad hoc fashion only.

In this paper we introduce –through the bias of logic– an alternative model of language which stresses its sign character. It is argued that language signs (like the signs of logic) are isomorphic and analogous to 'real' world signs. More specifically, we assume that (1) language symbols are signs, (2) signs arise from a dichotomous relation of perceived qualities, and (3) the meaning of signs emerges via mediation. We argue that on the basis of these assumptions and the properties of signs, a linearly complex parsing algorithm can be defined.

The first part of this paper is an attempt to offer a cognitive explanation of a theory of signs. On the basis of this, we introduce a model of logical signs in the second part. In the third part, we show how this approach may provide a framework for language modelling. We exemplify the potential of such a model, introduce a formal definition for its parser and prove its complexity.

1. Sign

In our conception of signs we follow the principles of Peirce's semiotic[1] (Peirce, 1931), (Tejera, 1988), (Jakobson, 1980). Peirce defines a sign as

[1] Familiarity with Peirce's theory is not assumed in the paper.

anything that stands for something else. That for which a sign stands, he calls the object of the sign. But that is only part of the story. Equally important is that a sign always stands for something else *in some respect.* For instance, it is usually acknowledged that smoke stands for fire. But smoke does not always stand for fire in the same respect. For a person lost in the wilderness, smoke may stand for fire in the sense that it indicates that people may be living there. But it may also indicate some danger. The element of a sign in virtue of which a sign stands for its object is what Peirce calls an interpretant. Sign, object and interpretant (each of which is a sign, recursively) form an irreducible relation which is called the *triadic relation of sign.*

In this paper the focus will be on signs. In our analysis of signs we start from the observation that the ground for any sign is a contrast in the 'real' world. Because sign and object are the primary representation of such contrast, sign and object must be different from each other.

In what follows we will use the sample phenomenon John likes Mary as our working example. We will assume that John and Mary are present for some time, that John is smiling, and that suddenly it is observed that likes occurs between them, and that Mary has flowers in her hand. What signs do we 'see'?

1.1. Cognitive grounds

Before answering the question above, we must first address a more basic one: how can we know about signs?

According to cognition theory (Harnad, 1987), the recognition of a sign begins with the sensation of a physical input. Physical stimuli enter the human receiver via the senses which continuously transform the raw data into internal sensation. In its turn, the output of the senses, a bio-electric signal, is processed by the brain in percepts. The generation of such a percept is typically triggered by a change in the input, or by the duration of some sampling time, e.g. in the case of visual perception.

The brain compares the current percept with the previous one, and this enables it to distinguish between two sorts of input qualities: one, which was there and remained there, something stable, which we will call a *continuant*; and another, which, though it was not there, is there now (or the other way round), something changing, which we will call an *occurrent.* We will assume that a percept may always contain qualities from the memory. Clearly, there will be only such qualities in the previous percept in the case of the observation of a new phenomenon.

Following the above model of cognition, we will assume that human sign processing is based on *coherent* sensations of collections of

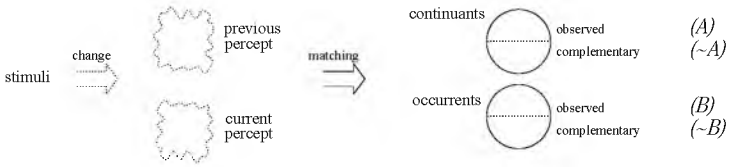


Figure 1. Cognitive model of sign recognition

continuants and occurrents which, by virtue of their coherency, are inherently related to each other. These collections form the basis for the perception of a phenomenon as a sign. We also assume that, by means of *selective attention*, we are able to recognise in these collections coherent groups of qualities: the qualities of the *observed* and those of the *complementary* part of the phenomenon (cf. fig. 1). In this paper, we will collectively refer to these collections as the *input*.

In our example we will assume that the complementary part contains sub-collections, the continuant 'John is smiling'(sm), and the occurrent 'Mary has flowers in her hand'(fl), but also memory knowledge about John(J), Mary(M) and likes(l) may be part of the complementary collections. Using set representation, the continuants(C) and occurrents(O) can be defined as follows: $C = JM \cup sm$, $O = l \cup fl$ where $JM = J \cup M$. Notice that J, l, M, sm and fl are only *denotations* of different collections of input qualities. We will show that on the basis of these collections of qualities the proposition John likes Mary can be derived. Such a proposition is a complex sign which, as we will assume, emerges from more simple signs. The sorts of such signs are introduced in the next section.

1.2. The variety of signs

In Peirce's view, the most complete signs are the icon, index, and symbol which represent their object on the basis of, respectively, *similarity*, *causality* and *arbitrary consensus*. Besides this taxonomy, Peirce also distinguishes signs, respectively, according to the categorical status of the sign, and according to the relationship between object and interpretant. From a categorical perspective signs can be qualisigns, sinsigns or legisigns, which correspond, respectively, to firstness, secondness and thirdness. In other words, a sign can be a *quality*, an *actual event*, or a *rule*. Seen from the perspective of the relationship between object and interpretant, a sign may be a rheme, a dicent or an argument. In other words a sign may signify a *qualitative possibility*, an *actual existence*, or a *proposition*. Thus we obtain nine kinds of sign and aspect which may be arranged in a matrix as shown in fig. 2 (the meaning of the horizontal lines and directed edges will be explained later).

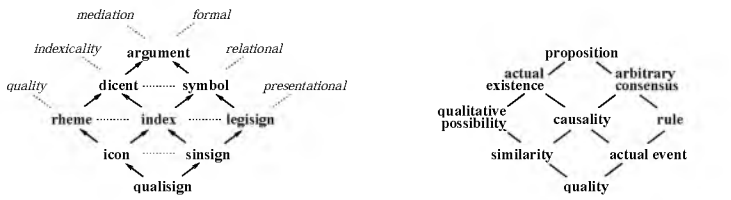


Figure 2. Peirce's classification of signs

Although Peirce defined more complex systems of signs, we hold that his 'simple' classification is the most practical. Here, the expressions 'class of a sign' or 'type of a sign' will be used interchangeably. In our specification of logical and language signs we will make use of Peirce's classes. A comparison between our use of them and his definitions may be found in (Farkas and Sarbo, 2000).

1.3. Primary signs

We recognise the input qualities as a proposition which is a sign. If, as we argue, such a proposition arises from more simple signs, then there must exist primary signs. In this paper it will be assumed that such signs are the input collections themselves. Indeed, these collections satisfy the specification of a qualisign which, according to Peirce, is a special sign that we only experience, but for which we have no denotation. Although our qualisigns are coherent, by definition, we experience them as independent signs.

Because such a proposition is a relation merging the qualities of the observed collections into a single sign, we will assume that those more simple signs are relations as well. Clearly, any such relation must specify the following: which input collections are involved, and which operations are applied. Therefore, such relations can always be interpreted as *formal* signs, that is, as signs which are about the *form*. In order to be able to refer to the qualisigns, we will make use of logical symbols which are the most general of that type of sign (Debrock et al., 1999), (Farkas and Sarbo, 1999). The most simple sort of them are the Boolean logical signs which we will ambiguously call logical signs.

We will represent Boolean logical signs as *functions* on two variables A and B, respectively, for the continuants and the occurrents, over the values 0 and 1, for the complementary and the observed part. We denote the observed qualisigns by the *functions* A and B (short for $\lambda AB.A$ and $\lambda AB.B$), respectively, for the continuant and occurrent collections, and, similarly, the qualisigns of the complementary part by the functions $\neg A$ and $\neg B$ (cf. fig. 1). By representing the input collections as sets (in the mathematical sense), we can define our 'universe' as $U{=}A{\cup}B{\cup}{\neg}A{\cup}{\neg}B$

where A, B etc. now denote the corresponding sets of qualities. Notice that these sets are *coherently* related to each other, contrary to formal logic in which the universe is an *arbitrary* set. We will assume that those sets are finite, therefore U is a 'closed world' by definition.

In our example, $A{=}JM$ and $\neg A{=}sm$ (we will finally recognise the phenomenon as the proposition John likes Mary which does not refer to the continuant sm), $B{=}l$ and $\neg B{=}fl$ (which is an occurrent, but which we are not focusing on).

1.4. The process of semiosis

How do complex signs emerge?

We try to answer this question by introducing a model of signs which we elaborate for logical signs. In this model it will be assumed that the semiosis of the input is a process in which triadic relations are generated recursively, revealing gradually more accurate and clear *approximations* of the full richness of a sign of the observed phenomenon. Accordingly, we will argue that the proposition of the input as a sign arises from the input qualisigns via a number of other signs. We will assume that in the semiosis the most complete signs (icon, index and symbol) function as a sign in the sense of the triadic relation, whereas the other signs function as an object. Notice that in this process the input collections are not a 'thing', but our reaction on the external stimuli. This *reaction*, the qualisigns as the interpretant, includes the generation of the selection of the next sign and object. We emphasise that also in the case of other signs we will talk about the interpretant in *this* sense.

In virtue of the fast and continuous nature of cognition, we will assume that such signs are not recognised isolatedly, but only as 'temporary' signs. Such signs, which are approximations of the final assertion, are *re*-presentations of the input qualisigns. Their types are identical to the classes defined by Peirce. The recognition process we have in mind can be illustrated by the perception of a motion picture. In that process, a series of pictures (cf. percepts) are input which are *not* recognised isolatedly, but which are necessary for observing motion as a change between the first and last picture of such a series.

2. Logic

The qualisigns form the basis for our semiosis. Because such signs are perceived as *independent* signs but it is their unity that signifies the phenomenon as a whole, we may assume that there exists a *need* for the representation of the full richness of their relation, eventually as a proposition.

2.1. First stage of recognition

The process of semiosis is bootstrapped by the definition of the first approximation of sign and object of the phenomenon. We assume that, basically, we are capable of identifying two sorts of such signs. One of them is the sign of the 'part-of' relation of the observed collections. Such a sign is A or B, or both, formally $A+B$. Because each of the collections is by definition a subset of the input thereby *similar* to it, such a sign is an icon. The other sign is the signification of the simultaneous occurrence of those collections. Such a sign is A and B, formally $A*B$. Because such an occurrence is a single event which happens 'now', it is a sinsign. Notice that the object of both the icon and sinsign is the collection of qualisigns.

In the example, we can define an icon as '*JM* or *l*' (both are 'parts' of the input), and a sinsign as '*JM* and *l*' (both appear simultaneously in an actual event). We emphasise that each of these signs is a representation of the collections of the input qualities, a sign that we recognise but we do not 'know'.

2.2. Second stage of recognition

The 'goal' of input recognition is the representation of a relation between the full meaning of A and B. In this respect the icon $A+B$ is more suitable to be used as a sign because it allows 'access' to the individual (although interrelated) collections, A and B, whereas the sinsign can only be interpreted as a 'whole'. The interpretant of this sign (icon) and object (qualisigns) must be some via-similarity-signified-aspect of the input. Because such a sign must refer to its object, but the qualisigns have no denotation, we will assume that in the semiosis the object is represented by a sign of it which must be the sinsign.

Similarity can partial, in which case, it can be signified from the point of view of the sign or the object. Let us begin with the latter one. The sign $A(B)$ appearing in $A+B$ can be said to occur in $A*B$ simultaneously with a 'missing' $B(A)$. By virtue of our assumption of a 'closed' universe, something which is not present can be represented by negation. As a result we get $A*\neg B$ ($\neg A*B$), the inhibition function. Such a sign, which denotes a *qualitatively possible* or abstract $A(B)$, is called in the Peircean terminology a rheme. These rheme signs define the individual 'parts' of the input in an abstract sense, as a *potential subject* of the observed phenomenon.

In the other case, the similarity of $A*B$ with $A+B$ is expressed by the similarity of $A*B$ with the individual signs appearing in $A+B$. This yields $A*\neg B+\neg A*B$, the exclusive-or function. Such a combination of two different kind of possible signs amounts to a *rule* which is called in

the Peircean classification a legisign. From the alternative representation of the exclusive-or function as $(A+B)*(\neg A+\neg B)$ we get the other meaning of the legisign which is an *abstract event* of the observed and complementary collections as a whole, a *potential predicate.*

But similarity can also be interpreted in the total sense. In that case the interpretant refers to the common meaning of the icon and sinsign. These signs are about the pair of collections A and B, which is *linked* to and occur with the pair of collections, $\neg A$ and $\neg B$, *complementing* it. The common meaning of these signs, which is called an index, is represented in Boolean logic by the DeMorgan postulates or, alternatively, the Shäffer and Peirce functions. By virtue of its aspect of linking, an index sign also has the meaning of *factuality*, *causality*, or *conversion.* By linking the icon and sinsign, respectively, the enumeration of the observed qualisigns as 'parts', and the representation of those observed qualisigns as a 'whole', the index (and the DeMorgan rules) can be said to reveal how the 'parts' define a 'whole', and how a 'whole' can be decomposed to 'parts'.

In our example, a rheme can be defined as '*JM* without *l*', denoting the sign of *JM*, one which is *abstracted* from the particular occurrence of *l*; and '*l* without *M*'. Notice that both rheme signs are representations of a 'part' of the observed phenomenon, in an abstract sense. A legisign can be defined as the combination of the interrelated parts, '*JM* without *l*' and '*l* without *JM*', providing us with a rule-like abstraction of the perceived event. The complementary meaning of the index can be expressed by means of 'John is smiling' ($\neg JM$) and 'Mary has flowers in her hand' ($\neg l$). Complementary signs can be necessary for a correct recognition. If, for example, John is not smiling, we may not be able to recognise likes as a property, or, we may do so, but then that likes might have a different flavour.

2.3. Third stage of recognition

We have now three signs, a rheme, a legisign and an index. Again it turns out that with respect to our 'goal' the index can more suitably function as a sign, whereas the rheme and legisign as (the sign of) its object. Indeed, the index signifies the actual relation between the observed collections and their *context* (the complementary qualisigns) whereas the rheme and the legisign are only about A and B, or their relation, in an abstract sense.

This signification of the index can be interpreted as complementation, as the *completion* of (the sign of) the object by the complementary qualities. In the case of the rheme this is formally defined as follows: $\neg(\neg A*B)$, $\neg(A*\neg B)= A+\neg B$, $\neg A+B$. Such an interpretant refers to

something *actually existing*, to $A(B)$ which is related to $B(A)$, but which relation is now signified via the complementary signs, implicitly (cf. *implication*). Such a sign, which is called in the Peircean terminology a dicent, denotes the *subject* of the observed phenomenon. The completion of the legisign is as follows: $\neg(\neg A*B+A*\neg B)= A*B+\neg A*\neg B$. This sign is the equivalence function which denotes a property, or *predicate*. Such a sign involves the aspect of a definition, or *arbitrary consensus*; it is called in the Peircean classification a symbol.

In our example, a dicent sign can be defined as '*l* when *sm*', or '*JM* when *fl*', as the completion of the abstract rheme signs by the context, or, alternatively, '*l* implicating *JM*', or the other way round. A symbol sign can be defined as '*JM* and *l* (observed signs) when *sm* and *fl* (complementary signs)' representing the perceived phenomenon as a property.

2.4. Final stage of recognition

The symbol sign above signifies an actual relation between the parts of the input; the signification of the dicent is implicit. Therefore we will take the symbol as a sign and let the dicent function as (the sign of) its object. Their interpretant will be the proposition merging the qualities of the observed parts into a single sign represented as 'A is B'. Formally, such a proposition is defined to arise by a syllogism (degenerately, in the logical sense). Such an operation, which is also called *predication*, is beyond the scope of Boolean logic.

In our example, the implication relation of '*JM* and '*l*' can be syllogistically combined with the property '*JM* and *l* when *sm* and *fl*'. Their common term can be expressed by the complementary signs, '*sm*' and '*fl*'. The resulting argument sign, which is a proposition, can be paraphrased as: 'There is *l* between *J* and *M*'.

Returning to the qualisigns, we mention that their logical representation, A, B, $\neg A$, and $\neg B$, can be completed with two more functions: 0 ('no input'), and 1 ('input exists'). As a result, we can conclude that in the process of semiosis *all* Boolean functions as signs can emerge and that these signs have indeed the aspects of Peirce's sign classes. The classification of logical signs is depicted in fig. 3 (on the left-hand side). Notice that in our derivation, the interpretant always emerges from *neighbouring* sign and object (such signs are connected in fig. 2 by a horizontal line). Such a mediation amounts to an *interaction* between sign and object (which are called the *constituents*). An alternative derivation of logical signs, in which, mediation is defined as the *application* of the logical function of sign to the one of the object may be found in (Farkas and Sarbo, 2000).

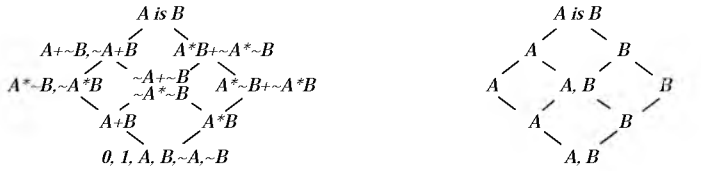


Figure 3. Classification of logical signs

3. Language

We will argue that language is logic, *sequentially*. In this section we will show that by means of this single condition a model of language can be derived which is analogous and isomorphic to the one of logic (Sarbo and Farkas, 2001). Language signs are *symbols* which are subject to syntactic and semantic rules. In this paper we will consider only syntactic rules, and restrict language to syntax (Quirk et al., 1985).

Syntactic symbols emerge from morphological and phonological ones which can be modelled in a similar vein as syntax. In this paper we focus on syntactic and morphological signs which are interrelated. An argument sign generated by a morphological sign recognition becomes a qualisign in a subsequent syntactic semiosis. In what follows, we first introduce a a classification of syntactic signs. In this, we will refer by a sign class to the classification of *syntactic* symbols. We will use the logical and syntactic names of the qualisigns interchangeably.

'Sequential' means that the input symbols appear one after the other as qualisigns. Earlier we pointed out that the qualisigns are the (first) representation of a contrast. In our cognitive model of signs we defined such a contrast between continuant and occurrent. Because language symbols are about 'real' world phenomena (typically), we may assume that an underlying contrast, analogous to the one of cognition, exists in language, too. The language equivalent of continuant and occurrent is identified, respectively, in the aspects, 'thing' and 'change', typically represented by nominals and verbs. Because language signs are inherently related to memory, a 'change' can also refer to an 'appearing' new fact, which is a relative change or alteration of some thing or event, typically represented by adjectives and adverbs.

3.1. A PRELIMINARY CLASSIFICATION

We will derive a model of language by transforming our specification of logical signs to a sequential one. Such a derivation is partly technical. In order not to get drown in the details, we make a preliminary attempt at a semiotic classification of the main syntactic concepts.

The argument, which is a proposition, must correspond to the notion of a sentence in as much as both are expressive of a statement. Hence, the dicent must be subject, and the symbol predicate. By virtue of its factual meaning (cf. 'modification') the index can be an adjective, an adverb, or a complemented preposition (in short, prep-compl). A rheme is a possible for the subject, for example, a noun; a legisign is an actual event in the syntactic sense, that is, a 'structure' defined as a rule, e.g. a verb(-complement).

3.2. Towards a sequential version of logic

Although the qualisigns are the representation of a contrast, in the sequential case when each qualisign consists of a single symbol, also that contrast will 'appear' sequentially. Accordingly, a qualisign will have either the aspect of a 'thing' (A), or a 'change' (B).

We argue that the *representational need* of sign recognition appears in language as the *relational need* of the individual symbols. Accordingly, it will be assumed that the qualisigns possess a *finite* set of *relational qualities.*

Because input symbols, i.e. their qualities, must belong to the *observed* part of the perceived phenomenon, in language we will not be able to distinguish between asserted and negated signs, at least not at this level (we will assume that input symbols are asserted signs). This implies that language phenomena must include their own context, hence a part of the input may have to be devoted to the representation of the context as a sign. We can have access to memory knowledge, e.g. a lexicon, but such knowledge is *not* related to the perceived input in any way. The lexicon is used for the re-presentation of the morphologically finished argument signs as a syntactic qualisign, in the definition of their syntactic relational qualities.

3.2.1. *Unique universe*

Syntactic qualisigns consist of a single symbol and define a unique universe, therefore different qualisigns cannot be merged. Because input symbols appear continuously, (i) *'place' has to be created for the appearing next qualisign.* What can be done with the previous qualisign? Obviously, the answer is that we have to re-present it by another sign. Following our model of sign recognition, such a sign can be an icon, or a sinsign.

Earlier we mentioned that, in the case of language, (ii) *a qualisign is either A or B, but not both.* Because a sinsign is a reference to an event, it must include the aspect of a 'change'. Accordingly, the re-presentation of a qualisign which is B can be a sinsign, hence the

one which is A must be an icon. Such a re-presentation involves the generation of a new sign, the denotation of which is *identical* to the one of the qualisign. From the bottom-up 'nature' of our model of signs it follows that the new sign will include the meaning of the older one degenerately (in the semiotic sense).

3.2.2. *Unique representation*
Icon and sinsign symbols are different representations of the same qualisign. From condition (ii) and the *uniqueness* of signs[2] it follows that such symbols typically will not be adjacent. Accordingly, (iii) *icon and sinsign symbols implement a 'sorting' re-presentation of the qualisigns.* Notice that, in the particular case, different icon and sinsign symbols can be adjacent if they define a *shared* universe.

By virtue of (i), we have to represent the previous qualisign by a new sign, which is an icon or a sinsign. The appearance of this new sign, in turn, may force us to do the same for the previous icon or sinsign and, eventually, signs may have to be generated in any class. In sum, the conclusion can be drawn that in the sequential model of logic there is sign generation *by need.*

3.3. Sign generation by need

In our model of signs the interpretant emerges via the interaction of sign and object. In language, such an interaction is called a *binding.* We assume that the denotation of a symbol emerging from a binding is defined by a homomorphism on the denotations of the constituent symbols.

Due to the sequential nature of language signs, we also have to consider degenerate variants of a binding which are: accumulation and coercion. In an *accumulation*, an existing sign is combined with another sign of the same type. Such an interaction assigns the same meaning to both constituents thereby rendering them indistinguishable. In a *coercion*, a new sign is generated for the denotation of an existing sign (which is said 'coerced'). Coercion applies if the signs, which are to interact, are incapable for accumulation or binding. In this form of an interaction we refer by the 'constituent' to the sign triggering the interaction.

3.3.1. *Default scheme of recognition*
Coercions play an important role in syntactic sign recognition more specifically in, what we call, the *default* scheme. This type of semiosis can be characterised as follows.

[2] Lexically ambiguous symbols have a unique denotation for each meaning.

By virtue of (i), signs are generated by need. The coercion of a qualisign, A or B, yields either an icon or a sinsign. Due to subsequent applications of (i), such an icon or sinsign is coerced, respectively, to a rheme or legisign, and eventually to a dicent or a symbol. An index cannot emerge this way, because there are no negated signs. By virtue of the appearing qualisigns, in the end, we may have a dicent and a symbol sign which are adjacent and generate the argument (the sentence as a sign). Notice that also a dicent or a symbol can be coerced to an argument, but such a sign will be a degenerate one, semiotically, because an argument sign represents the observed phenomenon in its *character* (Peirce, 1931), hence it must include both types of qualities.

In as much as, in our model, an argument arises from a dicent and a symbol sign (subject and predicate), we may conclude that, in the default case, *only* subject and predicate are recognised and their relation represented as a sentence. It will be argued that language sign recognition always follows this scheme and any deviation from it may only occur if otherwise a successful parse cannot be found. Such a case will be described in the next section.

3.4. The genesis of the context

In language we are burdened by the task of the recognition of the entire string of input symbols as a single sign. Although, in some cases, the sign generation operations (coercion, accumulation and binding) may be unsatisfactory, the above goal can yet be achieved if we allow for a sign, which is potentially subject or predicate, to be represented degenerately (in the semiotic sense). Such signs define, what we called, the *context*. The degenerate representation of symbols also plays an important role in the 'stepwise construction' of signs.

Sign degeneration ($\downarrow$) can be explicated by the ontological and phenomenological types of signs indicated in fig. 2, as follows. Dicent$\downarrow$index if the subject meaning is not present 'formally' (such a sign is not the subject of the sentence); symbol$\downarrow$index if the predicate meaning is not present 'mediationally' (such a sign is not capable for mediating between subject and sentence); dicent$\downarrow$rheme if the subject meaning is unfinished 'indexically' (such a sign is not a reference of something actually existing); symbol$\downarrow$legisign if the predicate meaning is unfinished 'relationally' (such a sign is not referring to an actually existing property).

For example, a sample context sign is dicent$\downarrow$index in Mary, John likes (the potential subject Mary becomes a context sign for likes); a sample 'stepwise construction' is symbol$\downarrow$legisign in likes Mary alone

(the potential predicate likes Mary is represented degenerately as a legisign so that the context sign alone can complete its meaning).

3.4.1. *Nested signs*

The existence of degenerate signs is related to our ability of parsing a contiguous segment of input symbols independently from the rest of the input. When such a segment is recognised, its meaning relative to the input as a whole is represented degenerately. Due to this degeneration, such a symbol *will not* become an isolatedly recognised sign. We will call such a segment a *nested input*, its sign a *nested sign*. The class of a nested sign (after degeneration) is equal to the class of one of its constituents (recursively). For example, a segment recognised as a dicent can be represented degenerately as a rheme or an index. The semiosis of a nested input can be implemented by a recursive application of the sign recognition 'machinery'.

In some cases we can know in advance if a symbol eventually will become a context sign and this allows certain optimisations. If a nested sign consists of a single input symbol, the recursive analysis may be replaced by a coercion. For example, an index sign can be directly generated by coercion from such an icon or sinsign. This kind of optimisation is typical for adjective, adverb and prep-compl symbols. Although such symbols are referring to a perfect property they cannot function as a predicate and this 'feature' is *lexically* specified in their syntactic rule. Also the 'stepwise construction' of a sign can be optimised by the immediate generation of a degenerate representation of its meaning.

The representation of logical signs in the sequential case is depicted in fig. 3 (on the right-hand side). Although the same denotations, A and B, appear many times, each of the occurrences has a *different* meaning. We will argue that syntactic signs can be specified on the basis of this classification by mapping the logical functions to syntactic symbols. The definition of such a mapping is the subject of the next section.

3.5. Relational need

Earlier we mentioned that syntactic symbols possess a relational need by means of which syntactic sign interactions emerge. Analogously to logic, such interactions are always between symbols of a *different* type of relational qualities. Clearly, binding is only possible if the universes of the interacting symbols have 'something' in common. More specifically, we demand that the set of relational qualities of such symbols are not disjoint, in which case, they are said syntactically *compatible*.

3.5.1. *The representation of syntactic qualities*
If, as we argue, logical and language signs are analogous, this must hold for syntactic symbols too. We map continuant and occurrent, respectively, via the contrast of language ('thing' and 'change') to the *relational types* 'passive'(p) and 'active'(a). Accordingly, the *relational need* or valency of a syntactic qualisign will be represented as a pair, consisting of a relational type and a finite set of relational qualities which are syntactic properties. Formally, we define the type 'neutral'(n); a sign having such a need is finished (relationally). We will refer to the relational need of a sign by its type, and call it an a-, p- and n-need, ambiguously. Although we will not specify the syntactic properties of language signs, we will introduce a classification for them. It will be argued that such a classification amounts to a (semiotic) definition of the types of speech.

The mapping mentioned above can be used for the definition of an initial representation of the syntactic relational needs of symbols in the different sign classes. In virtue of their semiotic function, this representation can be adjusted as follows.

Syntactic qualisign, icon, sinsign and argument symbols must have an n-need. Indeed, qualisigns are independent signs which cannot interact; icon and sinsign symbols are typically not adjacent and do not establish a relation; an argument which is the sentence as a sign must be finished. If icon, index and symbol signs which function as a sign in the sense of the triadic relation, are considered relationally 'active' and have an a-need by definition, then the signs which function as an object must be 'passive' and possess a p-need. However, in order to be consistent with the traditional lexical definition of language symbols, we have to adjust the above mapping by exchanging the relational need of sign and object. Lexical definitions always specify the relational properties of a symbol from the point of view of the sign in the sense of the triadic relation, but, as we have pointed out, in our model of language a symbol can also function as an object.

For example, the predicate of the sentence may arise from a legisign (a verb) via the complementation of an index. Traditionally, such a complement is lexically specified in the verb's entry. Semiotically, however, it is the complement that points to the verb and selects its actual meaning. This interpretation is also conform with the default scheme of sign recognition, according to which, verbs only function as a sign in the predication symbol interaction. Because also adjectives and adverbs function as a sign in the sense of the triadic relation, in our model of language the concept of 'modifier' and 'complement' *amalgamate.*

Any sign is a re-presentation of the qualisigns, therefore the qualities that contribute to the relational needs in the various classes must be

present in the qualisigns. Accordingly, we will represent the relational need of an input symbol as a set, an element of which is a triple, consisting of a type, a reference to a class in which the input symbol can contribute to that type of a relational need, and a set of syntactic properties.

Due to the sequential nature of language, a *p*-need is typically optional. A sign which can function as an object in the sense of the triadic relation, will only do that if there is a sign referring to it (in English, the *p*-need of a dicent sign is not optional, in virtue of the SV(O)-rule). However, an *a*-need is never optional. The presence of a sign always implies the presence of its object. In sum, the syntactic properties of a symbol only define the symbol's *potential relational need* in the various sign classes. Such a need can become an *actual relational need* in the course of the sign recognition process (when it is clear from the context the term 'actual' may be omitted).

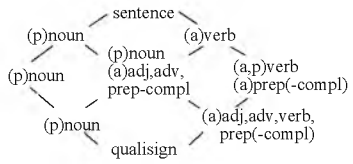
In what follows, the relational need of a symbol will be given in terms of its *a*-need in the different classes (*p*-needs, as well as, syntactic properties are not specified). For example, the valency of a transitive verb will be defined as a set, {legisign,symbol}, referring to an *a*-need, respectively, for the complement and the subject of the verb.

3.5.2. *Implementation*

Syntactic signs are re-presentations of their constituent symbols. Accordingly, the relational need of any sign must be less than, or at most equal to the *union* of the valency of the constituents (in the examples we will assume that the two are always equal).

We demand that an actual relational need is always satisfied. In particular, a binding satisfies a pair of *a*- and a *p*-needs by resolving them; an accumulation merges a pair of needs to a single one; a sign generated by coercion inherits the valency of the sign coerced. The *actual* relational need of a sign in some class can be empty, in which case, the sign is considered neutral (in that class). Such a sign cannot take part in a binding (but it can be subject to accumulation and coercion). This definition of 'neutral' is conform to our earlier use of this notion. Syntactic symbol interactions require the compatibility of their constituents. We demand that the syntactic properties corresponding to the actual relational need of the constituents of an interaction are *equivalent.*

Because the index sign does not partake in the default scheme of syntactic sign recognition, the generation of such a sign is subject to *special conditions.* We demand that a symbol can become an index having a *p*-need, either if any other analysis of that symbol eventually fails, or, if there exists an *a*-need due to a compatible legisign symbol.



	sin-sign	legi-sign	sym-bol
verb	+	+	+
prep	+	+	
adj, adv	+		
noun			

Figure 4. Classification of syntactic symbols

3.6. A Peircean model of syntax

In this section we specify the relational need of language symbols. Instead of using a set representation, we directly refer to the sign classes. For the denotation of language signs we make use the types of speech.

Syntactic qualisigns are specified as follows: A=noun; B=verb, adjective, adverb, prep(-compl), where 'compl' can be a noun, verb, adjective or adverb. Formally, we also define 0='no input' and 1='end of input'. The latter can represent the 'dot' symbol which is considered an *A and B* sign, having an a-need in any class, but incompatible with any sign except for itself. Hence, the dot symbol is capable of 'forcing' the realisation of pending interactions. In the examples we will assume that the input is closed by a finite number of dots.

The classification of syntactic signs is depicted in fig. 4 (on the left-hand side). Here, and also in later diagrams, a potential n- and p-need which occurs in combination with other relational needs is omitted. Notice in the index class the a-need of adjective, adverb and prep-compl symbols. Because the syntactic properties are omitted in our specification, an $a(p)$-need may stand for a number of triples. For example, in the legisign class, the a-need of a verb can represent the relational need for all of its complements. The a-need of syntactic symbols can also be used for a semiotic definition of the (main) types of speech as shown in fig. 4 (on the right-hand side).

The analogy of language symbols with logical signs must be clear. For example, the rheme is a noun from which some or all of its relatedness with the context and the predicate is removed, or *inhibited.* The index sign is an adjective or adverb which represents *complementary* information. The legisign is a verb(-complement) which refers to the event representation of the *exclusive-or* function. The subject is a completed noun which *implicates* the predicate it is related to. The predicate is a completed verb which is a property, an *equivalence.*

3.6.1. *Example*

We consider the syntactic analysis of our sample phenomenon given as the utterance, John likes Mary. The abbreviations J, l and M are now denoting the corresponding syntactic symbols. The relational need of l

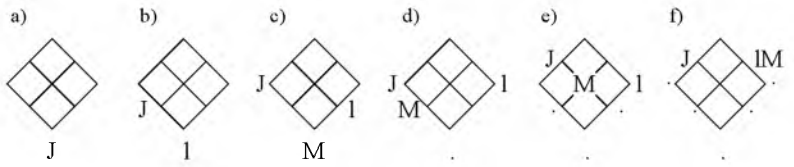


Figure 5. Syntactic analysis of John likes Mary

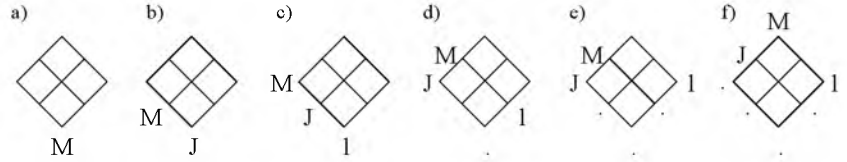


Figure 6. Syntactic analysis of Mary, John likes

is defined as {legisign,symbol}; J and M have no a-need in any class. The analysis is depicted in fig. 5.

In step (a) the symbol J enters as a qualisign. In the next step, the qualisign l appears, but which cannot accumulate with J because qualisigns are independent signs. Therefore J has to be coerced to an icon. The appearing qualisign M forces l to coerce to a sinsign, in step (c). Because icon and sinsign symbols are only 'sorting', binding is not possible and J has to be coerced to a rheme. Notice that J cannot be coerced to an index, because the conditions are not satisfied (cf. sect. 3.5.2). In step (d) the appearing dot symbol forces M to coerce to an icon and this, in turn, makes l to be re-presented as a legisign. Again, it cannot become an index because neither the special conditions are satisfied, nor has l an a-need in this class. At this stage of the analysis we 'know' that J and l are, respectively, the possibles for the subject and predicate.

Due to the next dot symbol, M is coerced in step (e) to an index sign by virtue of the a-need of l in the legisign class. In (f) two recognition steps are merged. In the first, J is coerced to a dicent by virtue of the next dot symbol. In the second, the binding between M and l yielding the predicate lM is established. In the last step (not displayed) J is predicated by lM and the sentence JlM emerges as a sign.

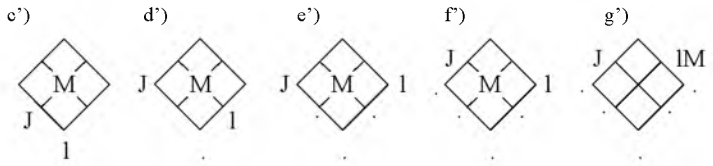


Figure 7. Syntactic analysis of Mary, John likes (cont.)

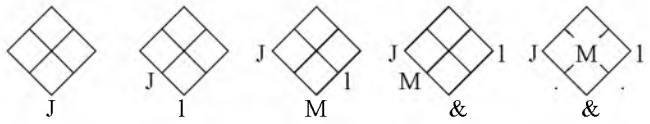


Figure 8. Syntactic analysis of John likes Mary and Kim (until the coordinator)

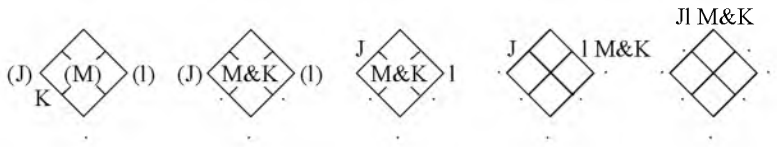


Figure 9. Coordination (alternative #1)

The analysis of a 'marked' word-order variant of our sentence is depicted in fig. 6-7. Here, the potential subject M is represented degenerately as an index. This is a consequence of the failure of the analysis following the default scheme (cf. fig. 6), because the entire input is not merged to a single sign. The analysis is backtracked, as a consequence of which the icon M is coerced, in step (c'), to an index (cf. fig. 7).

3.7. Coordination

In this section we make an attempt at applying our framework to the complex phenomenon of coordination. Because of space, we will restrict ourselves to the description of the *kernel* of such an algorithm.

Basically, coordination consists of three phases. In the first phase, input symbols are analysed up to the coordinator. In the second phase, first, all existing signs are saved, but their relational needs are remembered as 'traces' for later use by the algorithm. Then, an initial segment of the remaining input is analysed as a nested sign. In the course of the recursive analysis, existing signs and saved ones of the *same* class are allowed to be coordinated (this also includes the inheritance of properties). In the last phase, the remaining saved signs are restored. Save and restore, respectively, can be preceded or followed by the completion of a finite number of (pending) interactions.

An example is shown in fig. 8-10 (saved signs are enclosed in parentheses). Notice in fig. 9 the use of the 'traced' a-need of l in the coercion of the icon K to an index. Nesting also plays an important role in the

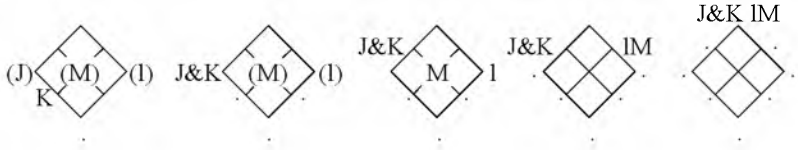


Figure 10. Coordination (alternative #2)

analysis of subordination. Such an application of recursion is exemplified in the Appendix. Notice that the conditions for the save and restore operations described above also apply to nested signs.

3.8. Morphology

We will argue that morphology (or, morpho-syntax) can be specified analogously to syntax. Whereas syntactic sign interactions generate the sign's sentential qualities (which can be used in a sentence level analysis) morphological sign interactions define the sign's syntactic properties.

The classes of morphological signs can be characterised as follows (now we will refer by a sign class to the classification of morphological signs). Dicent and symbol signs are finished morphologically and only accomplish a *sorting* re-presentation of signs as it is justified by their different syntactic properties. A symbol, for example, an adjective, has the *property* that in a syntactic analysis it requires a complement adjacent to it in the input (i.e. on 'surface' level). A dicent, for example, a noun or verb, does not have such a property. Such a symbol represents an *actually existing*, morphologically finished sign.

Rheme, index and legisign symbols, respectively, represent a *qualitative possibility*, a *factuality*, and something *rule-like*. For example, a rheme is a noun or verb, which (possibly) fits in the sentence, syntactically; an index is an article (a sign of definiteness), or, a morphological complement; a legisign is a prep(-compl), a morphological structure defined as a rule. These classes are involved in the generation of syntactic relational needs. For example, an article which is an index sign, can provide a morphological rheme symbol with a syntactic *p*-need. Such a relational need represents certain referential properties which, from the logical point of view, correspond to the inhibition function. A preposition, which is a legisign morphologically, can endow an index sign with a syntactic *a*-need. (N.B. the syntactic relational need should not be mixed up with the morphological valency which, akin to syntax, determines the *morphological* sign interactions.)

Finally, icon and sinsign symbols refer to a sorting re-presentation of the qualisigns alike to syntax. But more typically, such symbols will be adjacent, in which case, their interaction generates a specific syntactic property. For example, the interaction of a verb (icon) and a participle affix (sinsign) can generate a symbol having adjective-like syntactic properties. The constituents of such interactions define a shared universe (cf. sect. 3.2.2). In English, such icon and sinsign are typically written as one word. The classification of morphological symbols is

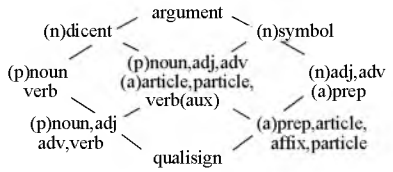


Figure 11. Classification of morphological symbols

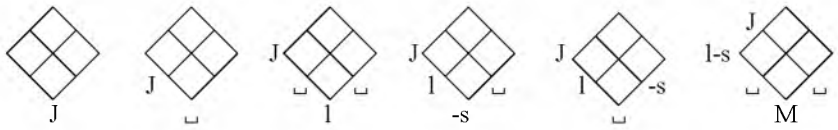


Figure 12. Initial part of the morphological analysis of John like -s Mary

displayed in fig. 11 (dicent, symbol and argument signs are assumed to arise via binding as defined earlier).

Morphological qualisigns are specified as follows: A=noun, verb, adjective, adverb, coordinator; B=preposition, article, particle, affix. Formally, we also define 0='no input', and 1='separation'. The latter can represent a 'space' symbol which is considered an *A and B* sign, incompatible for any interaction, except for accumulation (but only with the sign triggering the operation). Hence, the space symbol can 'sweep' out the morphologically finished signs. Dot symbols are treated in the same way as in syntax. Contrary to syntax, in our model of morphology, recursion is not required. This may be due to the left- and right-linearity property of morphological signs. Such signs typically arise via 'gluing' adjacent input symbols together. A sample analysis of the sentence John like -s Mary is shown in fig. 12 (an accumulated space is omitted). In this, and later examples we allow for subsequent morphological analyses to overlap.

3.9. Formal definition

We specify a recogniser for our model (except for nesting) as a push-down automaton. Formally, the automaton is defined as an 8-tuple $M = (K, C, I, \Gamma, \rho, s_0, F, \Delta)$ where $K=\{s_0, s_1\}$ is a finite set of states, C is a finite set of sign classes, I is a finite set of input symbols, Γ is a finite set of stack symbols, $\rho \in I \rightarrow \Gamma$ is a function defining the relational need of input symbols, s_0 is the initial state, $F \subseteq K$ is a set of final states, Δ is a transition relation consisting of is a finite set of transition rules.

A transition rule is a mapping $(p, u, \beta) \rightarrow (q, \gamma)$ where $p, q \in K$ are, respectively, the states before and after the transition, $u \in I^*$ are the symbols to be read, and $\beta, \gamma \in \Gamma^*$ are the symbols to be popped and pushed. We will assume that the stack is divided into *frames*. A frame

contains a storage area for each sign class, consisting of a location for the *next* and the *existing sign* of the class and a constant number of locations for *temporary* values.

The start rule and the one which handles the input of symbols are specified as follows (ε denotes the empty string, $\epsilon \in \Gamma$ stands for the empty value): **start** $(s_0, \varepsilon, \varepsilon) \rightarrow (s_1, \iota_\epsilon)$; **read** $(s_1, u, \iota_\epsilon) \rightarrow (s_1, \iota_{\rho(u)})$ where ι_x denotes a frame, in which, the existing sign location of the qualisign class contains the value x (the next sign location of this class is not used). For all other locations of the frames we assume that they have an identical value in ι_ϵ and $\iota_{\rho(u)}$. All other rules are 'internal' transition rules which operate only on the stack (ϕ and ϕ' denote frames): **transition** $(s_1, \varepsilon, \phi) \rightarrow (s_1, \phi'\phi)$.

We will simplify the specification of a transition rule by only defining ϕ and ϕ', and only specifying those locations of a frame which are involved in the transition (those not involved are assumed to have an identical value in ϕ and ϕ'). A further simplification is achieved by representing a frame as a *set* of storage areas.

Temporary locations can be necessary, for example, for the evaluation of a condition. The specification of such computations may require a number of internal rules which we alternatively define as a (logical) expression. Accordingly, the specification of temporary locations will be omitted. The value of the next and existing sign location of a class is a relational need, represented as a constant (cf. sect. 3.5.1).

Nondeterminism is implemented by backtracking (Aho and Ullman, 1972). In a transition rule we allow a reference to the actual evaluation mode, forward('f') or backward('b'), via the function *mode*. We will make use of a graph $G = (C, E)$ where $E = E_d \cup E_h$, and $E_d, E_h \subseteq C \times C$ are, respectively, the set of directed edges and horizontal lines (undirected edges) as shown in fig. 2. The successors and neighbours of a class are defined, respectively, by the functions $succ(c) = \{c' | (c, c') \in E_d\}$ and $adj(c) = \{c' | (c, c') \in E_h\}$. An element of $succ(c)$ and $adj(c)$ is denoted, respectively, as c^s and c_a.

In sum, in a transition rule we will refer to a set of triples (the set brackets are omitted). An element of such a set is given as (c, s, s') where c is a class, and s and s' are, respectively, its next and existing signs (any of s and s' may not be specified, in which case, it is denoted by a "_" symbol). The triples on the left- and right-hand side of a rule, respectively, refer to the current(ϕ) and next frame(ϕ') on the top of the stack (but a condition always refers to the current frame). The logical type of the next sign (q_n) of the qualisign class, $lt(q_n)$, is A if q_n has no a-need in any class; and B, otherwise. The names of the sign classes are abbreviated to a four letter name (also in the examples).

sorting

$(qual, \_, r), (icon, \epsilon, \_) \rightarrow (qual, \_, \epsilon), (icon, r, \_)$ IF $lt(r) = A$.
$(qual, \_, r), (sins, \epsilon, \_) \rightarrow (qual, \_, \epsilon), (sins, r, \_)$ IF $lt(r) = B$.

The remaining internal transitions are given by rule schemes for the class variable X ($X \in C \backslash qual$). In virtue of the special conditions required by the index class, the triple corresponding to the legisign class is explicitly defined in some of the rule schemes. We make use of the functions *cmpacc* and *cmpbnd* which, respectively, yield true if their arguments can syntactically accumulate and bind in the class specified. We also use the functions *ntrl*, *pssv* and *actv* which, respectively, succeed if their argument has a *n*-, *p*- and an *a*-need in the class given; and the functions *acc*, *coerce* and *bind* which, respectively, determine the relational need of the symbols yielded by accumulation, coercion and binding. The function *cndix* checks if the special conditions of the index class hold. The degenerate variants of the rule 'binding' are omitted (in such a rule, the result of binding emerges in the class of one of the constituents). The sentence as a sign arises in the next sign location of the argument class.

accumulation

$(X, r, r') \rightarrow (X, \epsilon, acc(X, r, r'))$ IF $cmpacc(X, r, r')$.

coercion$_1$

$(X, r, r'), (X_a, \epsilon, \epsilon), (X^s, \epsilon, \_), (legi, \_, r_l) \rightarrow (X, \epsilon, r), (X^s, r^c, \_)$
IF $ntrl(X, r') \wedge \neg cmpacc(X, r, r') \wedge cndix(X^s, r^c, r_l)$
WHERE $r^c = coerce(X, r', X^s)$.

coercion$_2$

$(X, \epsilon, r'), (X_a, r_a, \epsilon), (X^s, \epsilon, \_), (legi, \_, r_l) \rightarrow (X, \epsilon, \epsilon), (X_a, \epsilon, r_a), (X^s, r^c, \_)$
IF $ntrl(X, r') \wedge cndix(X^s, r^c, r_l)$ WHERE $r^c = coerce(X, r', X^s)$.

binding

$(X, r, r'), (X_a, \epsilon, r'_a), (X^s, \epsilon, \_), (legi, \_, r_l) \rightarrow (X, \epsilon, r), (X_a, \epsilon, \epsilon), (X^s, r^b, \_)$
IF $pssv(X, r') \wedge actv(X_a, r'_a) \wedge cmpbnd(X, r', X_a, r'_a) \wedge cndix(X^s, r^b, r_l)$
WHERE $r^b = bind(X^s, r', r'_a)$.

$cndix(X, r, r_l)$:
$X = indx \wedge (pssv(X, r) \wedge (mode = \text{'b'} \vee actv(legi, r_l))) \vee actv(X, r)) \vee$ TRUE .

A parser can be defined by using temporary locations. Such a location may contain the stack representation of an input symbol, or, one or two constants which are used as pointers to locations of the previous frame on the stack. When a segment of input symbols is to be analysed recursively, transition may proceed until no rules apply. Then, the current frame is pushed to the stack. Upon return from a recursion, the current frame and the saved one are 'merged' according to the particular properties of the nested sign.

3.10. Complexity

Fig. 2 defines a partial ordering on Peirce's classes of signs (horizontal lines are now neglected). Earlier we mentioned that a binding resolves, and an accumulation merges a pair of actual relational needs, whereas a sign generated by coercion inherits the valency of the sign coerced. In sum, there is no increase in the relational needs due to the qualisigns in any interaction. The class of a sign yielded by binding and accumulation is not lower, and the one yielded by coercion is definitely higher in the partial ordering, than the class of its constituent(s).

In the conditions of the transition rules we make use operations on sets which are *intersection* (e.g. for testing the compatibility of symbols) and *union* (e.g. for the generation of the sign yielded by binding). Each condition may need the evaluation of a constant number of such operations. Because the sets are finite (they cannot exceed the size of the lexicon, which is a constant), the complexity of the conditions is $\mathcal{O}(1)$ in the number of set operations. In as much as the number of classes as well as the relational need of input symbols are finite, the processing of an input symbol (which terminates when no transition rule applies) requires a constant number of transitions which are $\mathcal{O}(1)$ complex each. Eventually we get that the complexity of our model (if nesting is not allowed) is $\mathcal{O}(n)$ where n is the number of input symbols.

The complexity remains unchanged if additionally we allow nesting. In such a case, an input symbol is allowed to recursively 'start' recognition, and a later symbol to 'end' it, but each symbol is allowed to start or end recursion only *once*. We assume that the stack frames are linked to each other via a 'previous frame' pointer stored in a temporary location. Upon entering a recursion the current frame is saved. Upon return, there will be a *single* (nested) sign in the topmost frame of the stack. Let k denote the number of input symbols involved in the recursively analysed segment. Then, to find and fetch the values of the last saved frame needs at most $\mathcal{O}(k)$ steps, but the frames involved in this process will *not* be visited anymore. This can be solved by adjusting the previous frame pointers of the frames of the recursively analysed segment. Accordingly, any frame will be visited at most three times and the complexity of the algorithm will be $3{*}\mathcal{O}(n)$ which is $\mathcal{O}(n)$.

4. Summary and comments

A novel framework for language modelling is introduced which is based on Peirce's theory of signs. It is argued that such a model is only linearly complex. This result is conform with the experience that language is

a real-time complex phenomenon (Paul, 1984). An advantage of the Peircean approach is its potential of generalising the many issues of language as sign phenomena. Another advantage of the use of Peirce's semiotic is that the 'intermediate' results of sign recognition, which we called approximations, are always meaningful, both in the linguistic and the logical sense.

There is an interesting coincidence of our model with cognition theory, more specifically, with the concept of the short term memory (Broadbent, 1975). Peirce's classification consists of nine signs. One of them, the qualisign, is a special sign for which we have no denotation. Another one, the argument sign, is either the final result of recognition, or, is a nested sign, in which case it is recognised recursively (the signs saved by virtue of a recursion are assumed to be stored in a different memory area). In sum, the number of signs that have to be simultaneously represented during the process of semiosis, is *seven*, precisely the estimated size of the short term memory.

Finally, the combination of our assumption that, 'real' world phenomena can be observed by means of signs, with the results of this paper allows us to raise a conjecture: Whenever we experience a phenomenon as a qualisign, and we are able to interpret it as an argument sign, then there must exist a linearly complex system of symbols which is *meaningful.*

References

Aho, A. and J. Ullman: 1972, *Parsing*, Vol. 1 of *The Theory of Parsing, Translation and Compiling.* Prentice-Hall.

Broadbent, D.: 1975, 'The magic number seven after fifteen years'. In: A. Kennedy and A. Wilkes (eds.): *Studies in long term memory.* pp. 1–18, Wiley, London.

Debrock, G., J. Farkas, and J. Sarbo: 1999, 'Syntax from a Peircean perspective'. In: P. Sandrini (ed.): *5th International Congress on Terminology and Knowledge Engineering.* Innsbruck (Austria), pp. 180–189.

Farkas, J. and J. Sarbo: 1999, 'A Peircean framework of syntactic structure'. In: W. Tepfenhart and W. Cyre (eds.): *ICCS'99*, Vol. 1640 of *LNAI.* Blacksburg (VA), pp. 112–126, Springer-Verlag.

Farkas, J. and J. Sarbo: 2000, 'A Logical Ontology'. In: G. Stumme (ed.): *Working with Conceptual Structures: Contributions to ICCS2000.* Darmstadt (Germany), pp. 138–151, Shaker Verlag.

Harnad, S.: 1987, *Categorical perception: the groundwork of cognition.* Cambridge: Cambridge University Press.

Jakobson, R.: 1980, *The framework of language.* Michigan Studies in the Humanities.

Paul, W.: 1984, 'On heads versus tapes'. *Theoretical Computer Science* **28**, 1–12.

Peirce, C.: 1931, *Collected Papers of Charles Sanders Peirce.* Cambridge: Harvard University Press.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik: 1985, *A Comprehensive Grammar of the English Language.* London and New York: Longman.

Sarbo, J. and J. Farkas: 2001, 'A Peircean Ontology of Language'. In: H. Delugasch and G. Stumme (eds.): *ICCS'2001*, Vol. 2120 of *LNAI*. Stanford (CA), pp. 1–14, Springer-Verlag.

Tejera, V.: 1988, *Semiotics from Peirce to Barthes*. Leiden: E.J. Brill.

Appendix

In this section we illustrate the analysis of more complex examples and discuss some of the properties of our model of language. The examples are labelled by the corresponding language phenomenon. The presentation of an analysis is simplified by introducing a tabular form for the sign matrix. In such a table, a column corresponds to a sign class and a row to the recognition of an input symbol. In the examples, the treatment of space and dot symbols is omitted and the final step of sign recognition (the generation of an argument sign) is removed. A reference to a transition rule is abbreviated: input(i), accumulation(a), $\text{coerce}_1(c_1)$, $\text{coerce}_2(c_2)$ and binding(b); degeneration is indicated by a subscript 'd'. The accumulation of symbols is denoted by a "/" sign. Input symbols are written in boldface.

PP-attachment

Our first example is the sentence Mary eats pizza with a fork. The morphological analysis is depicted in table I. In step 8, the index sign 'a' binds to the rheme 'fork' and complements it with the property of 'definiteness'. The morphological sign 'a fork' is represented degenerately as an index. This index sign complements the legisign, 'with', yielding 'with a fork', a prep-compl sign having adjective- or adverb-like syntactic properties. The output of the analysis is: (Mary)(eats)(pizza)(with a fork) where an item enclosed in parentheses denotes an argument sign generated.

For the syntactic analysis, the relational needs are defined as follows: 'eats'=$\{legi, symb\}$, 'with a fork'=$\{indx\}$. The parses are displayed in table II and III (in the latter, only the steps deviating from the first one are given).

Coordination

We consider the sentence Mary is a democrat and proud of it which is morphologically analysed as: (Mary)(is)(proud)(of it). The syntactic relational needs are: 'is'=$\{legi, symb\}$, 'proud'=$\{indx\}$, 'of it'=$\{indx\}$. The resulting analysis is depicted in table IV. In step 8, the index signs 'proud' and 'of it' are merged to 'proud of it' having a single a-need in the index class. The coordination of 'proud of it' with 'a democrat' is possible because, syntactically, both signs are 'is'-complements.

Table I. Morphological analysis of Mary eat -s pizza with a fork

nr.	qual	icon	sins	rhme	indx	legi	dcnt	symb	*rule*
0	Mary(M)								i
1	eat(e)	M							i, c_1
2	-s	e		M					i, c_1, c_1
3	pizza(p)	e	-s	M					i, c_1
4	with(w)	p		e-s			M		i, c_1, b, c_1
5	a(a)		w	p			e-s		i, c_1, c_1, c_1
6	fork(f)		a	p		w	e-s		i, c_1, c_1
7		f			a	w	p		c_1, c_1, c_1, c_1
8				f	a	w	p		b_d
9					a-f	w	p		b, c_1
10								w-a-f	c_1

Table II. Syntactic analysis of Mary eats pizza with a fork (alternative #1)

nr.	qual	icon	sins	rhme	indx	legi	dcnt	symb	*rule*
0	Mary(M)								i
1	eats(e)	M							i, c_1
2	pizza(p)		e	M					i, c_1, c_2
3	with a fork(w_{af})	p		M		e			i, c_1, c_2
4			w_{af}	M	p	e			c_1, b_d
5				M	w_{af}	e-p			c_1
6					w_{af}	e-p	M		b
7							M	e-p-w_{af}	b

Table III. Syntactic analysis of Mary eats pizza with a fork (alternative #2)

nr.	qual	icon	sins	rhme	indx	legi	dcnt	symb	*rule*
3'	with a fork(w_{af})	p		M		e			i, c_2, c_1
4'			w_{af}	p		e	M		c_1
5'				p	w_{af}	e	M		b_d
6'					p-w_{af}	e	M		b
7'							M	e-p-w_{af}	b

Table IV. Syntactic analysis of Mary is a democrat and proud of it

nr.	qual	icon	sins	rhme	indx	legi	dcnt	symb
0	Mary(M)							
1	is(i)	M						
2	a democrat (ad)		i	M				
3	and(&)	a-d		M		i		
4					ad	i	M	
save								
5	proud(p)							
6	of it(oi)		p					
7			oi		p			
8					p/oi			
coordination								
9					ad&p/oi			
restore								
10					ad&p/oi	i	M	
12							M	i-ad&p/oi

Table V. Syntactic analysis of A man entered who was covered with mud

nr.	qual	icon	sins	rhme	indx	legi	dcnt	symb
0	a man (a_m)							
1	entered(e)	a_m						
2	who(w_h)		e	a_m				
recursive call								
3	was covered (w_{cd})	w_h						
4	with mud (w_m)		w_{cd}	w_h				
5			w_m	w_h		w_{cd}		
6				w_h	w_m	w_{cd}		
7					w_m	w_{cd}	w_h	
8							w_h	w_{cd}-w_m
return								
9	who-...-mud(w_{cm})		e	a_m				
10			w_{cm}	a_m		e		
11				a_m	w_{cm}	e		
12						e	a_m-w_{cm}	
14							a_m-w_{cm}	e

Discontinuous modification

We analyse the sentence A man entered who was covered with mud. From the morphological analysis we get: (A man)(entered)(who) (was covered)(with mud). In the syntactic analysis (cf. table V) we make use of a recursion initiated by the symbol 'who'. The syntactic relational needs are: 'entered'=$\{legi, symb\}$, 'was covered'=$\{legi, symb\}$, 'with mud'=$\{indx\}$. In the recursively analysed part, who was covered with mud, the symbol 'who' is considered a dummy subject. However, it is its indexical meaning that determines the properties of the degenerate representation of the nested sign (w_{cm}). In the example, w_{cm} is assumed to have adjective-like syntactic properties.