Natural Language Concept Analysis

Vera Kamphuis
Dept. of Language and Speech
University of Nijmegen
The Netherlands
v.kamphuis@let.kun.nl

Janos Sarbo
Computing Science Institute
University of Nijmegen
The Netherlands
janos@cs.kun.nl

Abstract

Can we do text analysis without phrase structure? We think the answer could be positive. In this paper we outline the basics of an underlying theory which yields hierarchical structure as the result of more abstract principles relating to the combinatorial properties of linguistic units. We discuss the nature of these properties and illustrate our model with examples.

Keywords Language processing, relational model.

1 Introduction

In most mainstream approaches to natural language modelling and parsing, some form of hierarchical structure plays a central role. The most obvious case in point is phrase structure. However, while the latter notion has shown its theoretical relevance in many ways, practical applications based on phrase structure description are not without problems. The main reason for this is the high flexibility of natural language. In performance data (i.e. actual language use), many disruptions of, and variations on standard phrase structure patterns occur. As a result, application of phrase structure-based parsers in natural language processing shows only limited success. This has inspired a search for alternative methods, such as statistical based or lexicon-driven parsing.

In our search for a solution to the problems mentioned we decided to take one step back, and examine the underlying nature of hierarchical structure in general, and phrase structure in particular. Our aim in doing so was to find a more principled solution to the problems of linguistic analysis and parsing. We looked for ways to derive structural information from input, and to incorporate this in a mathematically well-founded theory of knowledge representation. As a result we found a level of abstractness that, in principle, allows language-independent modelling and analysis.

In our approach we capitalise on the property that the information carriers, the lexical items, are 'willing' to combine. These combinatorial properties are determined by inherent characteristics of lexical items. Hierarchical structure follows naturally from the interaction of these properties, while leaving room for variation and flexibility in structural patterning.

In our model of natural language (NL) the input is represented as a binary relation. This is due to the *dichotomy* of language, meaning that a classification of lexical items as objects and attributes can be made (we use the term "dichotomy" in a restricted sense: a division into two mutually exclusive parts). The two classes are interrelated, and their relation can be determined merely on the basis of lexical information and some general principles, like word order (e.g. SVO). The relation between the classes is due to the principle of *relatedness*. This principle entails that any non-empty set of objects implies the existence of a non-empty set of attributes (properties) it is related to, and vice versa. Minimally, an observable entity (object) has the property of existence (attribute). This principle gives rise to a relation representing the semantics of the 'thought' described by the sentence in terms of a set of related items, called *observations*. An observation captures a set of objects and properties that are mutually characteristic of each other. Such a notion corresponds to a *formal concept* in lattice theory. We will show that the above relation is supported by linguistic considerations.

Our approach to language, Natural Language Concept Analysis (NLCA), constitutes a linguistically and mathematically based theory. This is reflected by the different readings of the acronym, as follows. NLC(A): the analysis of concepts that play a role in natural language; (NL)CA: the lattice the-

V. Kamphuis and J.J. Sarbo (1998) **Natural Language Concept Analysis.** In D.M.W. Powers (ed.) *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning,* ACL, pp 205-214.

oretical model of formal concept analysis applied to natural language; N(LCA): a natural transformation on language (in concrete, on functor–argument relations).

1.1 Related research

Our theory goes together with a movement of modern formalisms in computational linguistics which can be characterised by a shift of emphasis from a large, detailed syntax and simple lexicon, to a compact syntax and rich lexicon. Amongst other works, one can cite HPSG (Pollard and Sag, 1994), and most recently, a proposal by Berwick and Epstein (Berwick and Epstein, 1995).

Berwick and Epstein outline a model that, in accordance with Minimalist principles, does not posit "any syntactic entities at all beyond what [is] absolutely necessary for linguistic description and explanation." The necessary machinery, as they point out, is one based on categorial grammar (Lambek, 1988). Their argument follows from the fundamental idea that natural languages are limited to rules specifying how constituents can be concatenated to form larger constituents. Berwick and Epstein introduce a single syntactic operation, Hierarchical Composition (HC), for the realization of such syntactic constraints.

With respect to the above mentioned movement in natural language processing, we note that the endeavour to move (almost) all information to the lexicon can be theoretically justified. Intuitively, practical NL formalisms like HPSG can be seen as variants of two-level, e.g. attribute grammars. Theoretically, for such a grammar, a weakly equivalent grammar using only a single nonterminal symbol exists (Franzen, 1983). In such a grammar all structural information is specified by attribute functions. These functions can be defined by the lexicon.

2 A supporting theory

Natural language modelling usually assumes some form of hierarchical structure as given. Experience shows that practical application of such an approach to a non-trivial subset of the language can be a highly complex task (Aarts, 1991). In our search for a more flexible basis we arrived at the question: How does phrase and clause structure emerge in natural language? It appeared that this question is related to a more general one: How can knowledge about real world be structured?

We found a philosophical background in C.S. Peirce's pragmatism (Peirce, 1931) and a mathematical formalisation of Peirce's ideas in R. Wille's theory on Formal Concept Analysis (FCA) (Wille, 1982). Relatedness, for example, relies on Peirce's epistemological argument saying that "... there is no judgment of pure observation without reasoning" (Houser and Kloesel, 1992). This means that an observation is always tied to "judgment"; in other words, in our case, observation of an object always implies the presence of an attribute, and some interpretation of their relation.

In the FCA framework, observable world is described by a binary relation between the sets of objects and attributes. These sets give a dichotomous characterisation of observable entities, and together with their relation formalise Peirce's universal categories: firstness, secondness and thirdness. These are defined as follows: "The first is that whose being is simply in itself, not referring to anything nor lying behind anything. The second is that which is what it is by force of something to which it is second. The third is that which is what it is owing to things between which it mediates and which it brings into relation to each other" (Houser and Kloesel, 1992).

For the time being we adopt the interpretation of Lehmann and Wille (Lehmann and Wille, 1995) who state that "*the object g is a [f]irst ... to which the attribute m is a [s]econd ...*". According to Lehmann and Wille, this interpretation is compatible with Peirce's general understanding of firstness and secondness.

In FCA, observations, or concepts, are mathematically formalised. Traditionally, the philosophical notion of a concept is determined by its *extension* and its *intension*. The extension consists of all elements (set of objects) belonging to the concept while the intension covers all properties (set of attributes) valid for all those elements.

In the mathematical model, the triple consisting of the sets, objects (G; Gegenstände) and attributes (M; Merkmale), and the relation between them (R), is called the *context* (we assume that G and M are finite sets). We say, for $g \in G$, $m \in M$, $(g,m) \in R$ or equivalently, (gRm), iff the object g has the attribute m.

For a context the following mappings are defined: $A' = \{m \in M \mid gRm \text{ for all } g \in A\}$ for $A \subseteq G$; and $B' = \{g \in G \mid gRm \text{ for all } m \in B\}$ for $B \subseteq M$. A *(formal) concept* of a context (G, M, R) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, which satisfies the conditions (i) $A' = B$ and (ii) $A = B'$.

Informally, A' is calculated from A by considering the elements of A and accumulating the properties common to them all. B' is calculated dually. We say (A, B) is a concept if, by the above calculation, A and B mutually determine each other.

For any concepts (A_1, B_1) and (A_2, B_2) of a context the *hierarchy of concepts* is captured by the definition: $(A_1, B_1) \leq (A_2, B_2)$ iff $A_1 \subseteq A_2$ (or equivalently, iff $B_1 \supseteq B_2$). When the above order relation holds, (A_1, A_1') is called the *subconcept* of (A_2, A_2'), and (A_2, A_2') the *superconcept* of (A_1, A_1'). The set of all concepts of a context with this order relation is called the *concept lattice*.

3 Linguistic relations

NLCA applies Wille's theory to natural language by the equivalence: attributes are functors, and objects are arguments. Functor-argument relations, the manifestations of the (combinatorial) properties of lexical items, have various realizations on the linguistic level. For example, the verb–complement relation is not the same as the relation of modification. This becomes clear when we look at the optionality of modifiers. In English, we cannot say, on the basis of encountering a noun, that it needs an adjectival modifier; however, when we encounter an adjective, we do know that at some level it needs a noun because it is a semantic predicate (functor) taking an argument of which it is predicated. In this case, then, there is an asymmetrical relation between functor and argument. On the other hand, the relation between a verb and its complementation is a symmetrical one.

In NLCA, we distinguish between two kinds of relations: major and minor. These types of relations can be recursively nested, and their sum uniquely characterises the input. The first type of relation, the *major relation*, or predication, is a pair (p, a), where a functions as an argument to the predicate p. A major relation may involve the distinction between an action/state and its participants (symmetrical relation: each requires the presence of the other) and between an action/state or participant on the one hand and its properties on the other (asymmetrical relation, or modification: the predicate requires the argument of which it is predicated, but the reverse does not hold). We call predicates of the first type *major predicates*, and predicates of the second type *minor predicates*.

It is interesting to note that this distinction reflects the difference between *constituency* on the one hand, and *dependency* on the other. In linguistics, these two relations are often treated as (formally) equivalent alternatives.[1] In the current view, they entail a difference in status of the units that are involved in the relation. The nature of the relation in both cases is that of predication; however, in the constituency case each part assumes the presence of the other, whereas in the dependency case, the predicate is optional.

There are various distinguishing factors between major and minor predicates. In English, major predicates (usually) relate to the noun-verb division; minor predicates do not. Major predicates are typically realized by verbs; minor predicates by adjectives and adverbs. There is never more than one major predicate associated with an argument; there may be several minor predicates related to the same argument. (This reflects the possibility of having zero or more modifiers of an action or participant.) Both major and minor predicates can provide semantic roles, but major predicates introduce participant roles for their arguments; minor predicates can introduce additional roles (such as location or manner) or properties of their arguments.

The second type of relation, the *minor relation*, or qualification, distinguishes between the core content of a linguistic expression and some qualification of it. At the level of an action and its participants, for example, this qualification may relate to referential status of NPs (e.g. definite vs. indefinite article), or to tense and aspect information at clause level. Intensifying adverbs (e.g. *very, extremely, deeply*) and comparative adverbs (e.g. *more* and *most*) also belong to the class of qualifiers. These examples suggest that qualification may also have a symmetrical and asymmetrical variant: article, tense, aspect etc. being of the first type, and intensifying and comparative adverbs of the second. However, this is still an object of further study. In this paper we will restrict ourselves to the distinction between qualifier and core in general.

The difference between a minor predicate and a qualifier is that the latter does not introduce a meaning that is independent of the element it qualifies.[2] The presence of a qualifier of a specific type, therefore, also signals the presence of its counterpart. Furthermore, there can be several modifiers associated with an argument or predicate; typically, however, there will only be a single (possibly composite) qualifier. In the case of referential information, for instance, the qualifier situates the argument or predicate in its referential context of which there will only be one. In some cases different aspects of the

[1] However, see (Fraser, 1996) for some qualifying remarks on this topic.

[2] By contrast, a minor predicate has some aspect of meaning that is independent of the element it combines with. This is illustrated by the fact that minor predicates can be used in different contexts. For example, a prepositional phrase can modify an argument (e.g. noun) but also a predicate (e.g. verb). An adjective phrase can be used as a modifier of a noun, but also in the complementation of a verb.

qualifier can be expressed separately (such as tense and aspect); in that case these different aspects must be unifiable but there cannot be more than a single qualifier relating to the same domain.

The qualifier evokes its counterpart; nevertheless, the semantic 'core' is also complete in itself, in that it forms a full account of semantic relationships. Therefore it does not require realization of the qualifier as such: cf. the use of such bare relations in captions or telegram style speech (e.g. "*Lion attacked woman!*").

Summing up, we distinguish between the following relations:

- *major predication*
- *minor predication*
- *qualification.*

A schematic representation of these possibilities is given in Fig. 1.

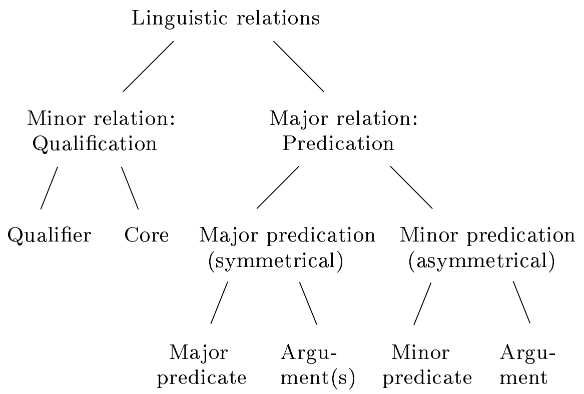

Figure 1: Inventory of linguistic relations in NLCA

It is important to note that this diagram does not represent the hierarchical structure of sentences, or the organisation of conceptual content within the sentence (we will come back to this below); it merely shows the different types of relations that our approach identifies. These relations lie at the heart of structure formation in NL. How phrase structure emerges as a result of their interaction is explained in Sect. 6.

As mentioned, the different relations can be recursively nested. For example, at the level of argument, a modifying predicate may be added in the form of an adjective phrase, or a qualifier may be present in the form of a determiner. Each element that is added stands in a certain relation to its counterpart, based on the type of relation that was applied.

There is a potential mapping between the linguistic relations displayed in Fig. 1 and the hierarchical organisation of information structure. For example, it is likely that the major predication relation is the most important information carrier with respect to the semantic content of the sentence, and that the minor predication relation reflects additional information of less importance. This illustrates the relative contribution of the different linguistic relations to information content. In information retrieval this could help to generate a concise representation of retrieved text. It would also be in line with the use, already mentioned, of the major predication relation in captions or telegram style speech; furthermore, it could be a possible explanation for the ability of speed-reading that readers may develop.

The qualification gives concrete reference to all the items involved in the predication relation, and as such is relevant for all levels. The presence of the qualifier at all levels of representation is a matter of some importance: word order, for instance, may also be classified as part of the qualification relation (e.g. in English, word order is relevant for identifying questions, and also in assigning thematic roles to participants). The relationship between qualifiers in NLCA, and operators in the semantically based hierarchy of Role and Reference Grammar (Van Valin, 1993) would be a potentially useful area to investigate.

4 A first sketch of the model

The distinctions made above have been incorporated into the NLCA-model on the basis of the abstraction of FCA: the *context.* In the dyadic model of FCA, a context allows only two kinds of entities: object and attribute. Therefore, each lexical item has to be classified as one these, based on its lexical type. Typical objects are nouns; typical attributes are verbs (major predicates), and adjectives and adverbs (minor predicates). We refer to these attributes uniformly as *major attributes* (involving predication). Qualifiers are classified as attributes, as well. We call them *minor attributes* (involving qualification).

A comment with respect to the classification of lexical items and its relation to Peirce's universal categories is in place. We mentioned that objects and attributes formalise the categories firstness and secondness. Each item of these classes may evoke a different relation (called interpretant) depending on the item's syntactic and semantic properties, and in general, the properties of the item as a sign (Liszka, 1996). In NLCA these interpretants are instantiations of the linguistic relations formalising the category of thirdness. For example, the interpretant created by a verb, an instance of a major predication, may 'explain' how that verb binds its arguments together "in a bundle of interlocking relationships" (Sowa, 1996).

The surjective mapping from lexical types to the sets of the dyadic model can be defined without causing confusion. The set of lexical types defines a partition of L, the set of lexical items and semantic roles involved in the analysis, which is further partitioned according to the dyadic model, yielding the sets G and M. From $G \subseteq L$ and $M \subseteq L$ follows that there is an embedding of the relation $R \subseteq G \times M$ in $L \times L$. This means that any pair $(g, m) \in R$ can be defined as the unique yield of l_1 and l_2 ($l_1, l_2 \in L$) by the assignments $l_1 \mapsto g$ and $l_2 \mapsto m$, where $\mapsto$ respects the mapping of lexical types.

As said above, each lexical item is classified according to its type. Furthermore, with each lexical item is associated a number of positions for *internal* and *external arguments*, denoted as suffixes, _int and _ext, respectively.[3] Internal arguments contain information regarding the item itself. External arguments relate to combinatorial demands to make a complex linguistic unit, according to the linguistic relations described above. We say the input is *well-formed* if the combinatorial demands of each lexical item are satisfied.

The internal argument positions are filled (i.e. assigned) by modifiers and qualifiers, which refer to distinct domains of analysis. For each domain holds that when an argument position is filled by more than a single element, these different elements have to be compatible (possibly depending on the context). For example, with multiple modifiers, e.g. two or more adjectives modifying a noun, the modifiers have to be semantically compatible in order to make a sensible construct: cf. *the tall happy girl* vs. ?*the tall short girl*.

The external arguments of a verb (major predicate) are determined by the verb's *valency*: the subject is also an external argument. These external arguments are involved in a symmetrical relation: an object fills the external argument position of an attribute, and vice versa, the attribute fills the external argument position of that object.

The external argument of a modifier (minor predicate) is involved in an asymmetrical relation: an object fills the external argument position of an attribute, and the attribute fills the (modifier) internal argument position of that object.

The qualifier-domain of the internal argument contains specific information that relates to the type of lexical item. For nouns, it is information regarding reference: specific/generic/unique reference; number. For verbs, it is information regarding finiteness/tense/aspect, etc. Thus, when the qualifier-domain of the internal argument of both the object and the major attribute is filled, there is explicit reference with respect to the action and the participants involved. Since qualifiers contain a specific type of information, they can be regarded as a syntactic pointer to the qualified element itself: if this domain of the internal argument is filled, there must also be an object/attribute of the type that the internal argument belongs to. In case the qualifier precedes its argument, this feature is reflected in the computational model by introducing placeholders, called *Proto*-items (cf. Sect. 6). Proto-items can only be introduced by qualifiers. When the argument of the qualifier is found, it replaces the Proto-item and fills the external argument position of that qualifier. The relations that the Proto-item is involved in are inherited by that argument.

Besides these relations, NLCA applies a set of general principles, like word order (e.g. SVO), inheritance of relation and 'greedy' binding of lexical items. By the latter principle, the input string, forming the context of each lexical item, is evaluated from the perspective of that item and its needs: functors take the textually nearest arguments available, and vice versa. In NLCA input is analysed from left to right.

Summarising thus far, we have incorporated the following aspects in our model:

- each linguistic unit is classified according to the object/attribute dichotomy
- each linguistic unit has positions for internal and external argument(s)
- the internal argument positions are filled by qualifiers and modifiers
- the external argument positions are filled by elements that are involved in the predication relation.

5 Relation Matrix

The analysis of examples in NLCA is represented in terms of a so-called Relation Matrix (RM). A Relation Matrix shows the relation between objects (represented in rows) and attributes (columns).

Conform to our definition in Sect. 4, we say a Relation Matrix is *well-formed* if each external argument position of each attribute is filled (meaning that the

[3] In procedural terms, argument position and argument correspond to formal and actual parameter, respectively.

semantic roles of major predicates are realized and that all other combinatorial demands of attributes have been fulfilled) and each object is the external argument of some attribute. This implies that the input corresponding to a well-formed RM must consist of one or more clauses. In this paper we focus on the case that there is only a single clause.

We represent a symmetrical relation by a pair of asymmetrical relations, and an asymmetrical relation by a directed relation, called a *pointer* (PTR). Technically, the value of a matrix element, RM[i,j], is a tuple encoding a Boolean variable and a set of PTRs.

In the graphical representation the value of a Boolean variable is represented by a '+' (*true*) or the empty string (*false*). We may also use the notation '$+_i$' referring to the ith *true*-value assignment. A PTR is depicted as a directed edge. Internal argument positions of objects and attributes are displayed to their left-hand side (there is one argument position for the qualifiers, and one for the modifiers); external argument positions to their right. Empty argument positions are omitted.

If the external argument position of an object (attribute) is filled by an attribute (object), we assign *true* to the Boolean variable of the corresponding cell in the RM. These variables will be used for the representation of linguistic structure. The assignments can take place after the analysis is completed, or, in most cases, during the analysis.

Attributes may have more than one external argument position, and each of these may be involved in a different relation. Therefore, we use the convention that external argument positions of verbs are displayed in separate columns. The relation of attributes and their external argument positions can be traced back in the Relation Matrix, however, in the examples, we do not graphically represent it.

6 Examples

It is now possible to illustrate the model by discussing some examples in detail. The language of illustration will be English.

Example 1 *The door squeaks.*

the Minor attribute; it generates a column. Articles belong to the class of qualifiers, and thereby require the presence of their counterpart. They create a Proto-object that needs to be filled, of which they themselves are the internal argument. The Proto-object allocates a row in the RM; 'the' points at its internal argument.

· the → Proto-object_int

door Object; fills the Proto-object slot created by 'the' and thus finds 'the' pointing at its internal argument position. 'door' itself points to the external argument position of 'the' (on the basis of the combinatorial demands of the latter), leading to the linguistic unit 'the door' and yielding a '+' in the RM.

This leads to the following *postulate*: whenever the external argument position of an attribute (except for a verb) is filled, the elements transitively involved in the relation constitute a phrase.

· 'door' replaces Proto-object
· door → the_ext
· '+' in RM in cell door/the

squeaks Major attribute; its internal argument is the present tense marker 's'; its external argument is the θ-role of THEME (but see below), filled by 'door'; hence there is a pointer from 'door' to this role,[4] and a '+' is put in the RM. 'squeaks' itself is the external argument of 'door'; again a '+' in the RM. This time the relation involves the external argument positions of both attribute and object (as opposed to before, when the internal argument position of the object was involved). There are no external argument positions left unfilled; this signals a clause.

Again we can formulate a *postulate*: there is a well-formed clause when all external argument positions of a major predicate (attribute) are filled by objects, and the attribute fills the external argument positions of those objects, and no external argument positions of items involved in the major predication are left unfilled. (N.B. a VP can be found as a subset of a clause.)

· squeak → door_ext
· door → squeak_ext (THEME)
· '+' in RM in cell door/squeak

The precise labelling of thematic roles varies across different models. The current role is suggested by Haegeman's THEME$_2$ (Haegeman, 1991), in Role and Reference Grammar (RRG) (Van Valin, 1993) it would probably be called

[4] There could in fact also be pointer from the third person present tense marker to the object or THEME-role: this can be relevant especially in head-marking languages where the verb carries morphemes indicating the person and number of its arguments. Note, incidentally, that the latter situation is totally unproblematic for NLCA, since it is based on the abstract linguistic relations rather than on concrete syntactic realizations.

an EFFECTOR. Where Van Valin and Haegeman are in conflict, we shall at this point choose the more general role of the two. However, a more detailed analysis of predicates into different types (*accomplishment*, *activity*, *state* and *achievement*) with associated logical structure (as in RRG) is a desirability for the development of the NLCA-lexicon.

The Relation Matrix for this sentence is displayed in Fig. 2 below.

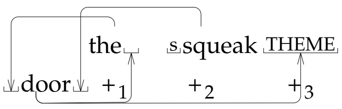

Figure 2: *The door squeaks*

Note that objects may have pointers to several external argument positions of attributes. This corresponds to saying that the object may fulfil the argument role of more than one attribute. This is in fact the case: cf. the occurrence of multiple modifiers of a single head, but also, in the current example, 'door' functioning as argument to both the article 'the' and the verb 'squeaks'. That the two have different status is not a problem and is in fact relevant: when 'door' fulfils the external argument role of a verb it is involved in the major predication, because it itself requires a verb, but when it fulfils the external argument role of adjectives or of articles it is not; it just completes their demands.

There is another aspect of the model that can be illustrated on the basis of a sentence of this kind. For this purpose, let us use the sentence *The moon rose.* The lexical item 'rose' is ambiguous: it can either be the past tense of the verb 'rise', or it can be a noun referring to a flower.[5] In the former case, the analysis will take place in the same way as in the example above. But let us look at what would happen in the latter case.

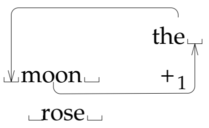

Figure 3: *The moon rose*

[5]We disregard for the moment the possibility of analysing the two nouns as a compound.

In this example, we can see that assigning 'rose' to the object class leaves the analysis incomplete: the object is not connected to any attribute. Hence, under this reading, the sentence is ungrammatical (cf. Fig. 3).

Example 2 *The happy girl bought some flowers.*

the cf. Ex. 1.

· the → Proto-object_int

happy Attribute involved in the predication relation (minor predicate); generates a column. Its internal argument position is not filled. Its external argument position needs to be filled with an element of type Object (as an adjective, it is predicated of nominal elements). There is a Proto-object present, hence there is a pointer from the Proto-object to the external argument position of the attribute (greedy binding), so we can put a '+' in the RM (under 'happy'). The attribute itself points to the internal argument of the Proto-object. Note that this leads to a *chain* of PTRs from 'the' via the Proto-object to an external argument that has been filled; such a chain gives rise to *inheritance* of the '+' to all relations involved in it. Therefore, there will also be a '+' under 'the'. This, in fact, creates a *nominal adjective phrase* with an implied head (the Proto-object).

· Proto-object → happy_ext
· happy → Proto-object_int
· '+' in RM in cell Proto-object/happy
· '+' in RM in cell Proto-object/the

There is an important difference here between the role and treatment of the article and the adjective. Note that the nominal adjective phrase would not be created without the article: it is the article that supplies the Proto-object that functions in the nominal adjective phrase. It can do so because it belongs to the class of qualifiers; they require the presence of their counterpart and therefore can be said to create a Proto-object. The adjective, on the other hand, belongs to the class of modifiers that are involved in the relation of minor predication. For this reason, they can be said to have an implicit object required to fill the place of their external argument position. The Proto-object generated by the qualifier can fill this role. Both qualifier and modifier belong to the class of internal arguments; however, they do not have the same status and are treated differently. The qualifier can generate a Proto-object (or a Proto-attribute if it is the qualifier of an attribute) but this Proto-item does not fulfil its external argument need: otherwise, it would be wrongly assumed that a string consisting of the qualifier only would be grammatical. Hence there is no pointer from the Proto-object to the external argument of the qualifier. With the modifying adjective, exactly the reverse situation holds. As adjectives can be used in different types of contexts (e.g. attributively or predicatively), they do not create a Proto-object. However, since they are involved in the relation of predication, their external argument position can be filled by a Proto-object generated by a qualifier. They themselves then, also may point to the internal argument of that Proto-object.

girl Object; replaces the Proto-object. The object points at the external argument position of 'the'. There is still a phrase, but now it is a full noun phrase rather than a nominal adjective phrase. Since there is not yet a pointer to the external argument of the noun, we still do not have a clause, only a phrase.

· 'girl' replaces Proto-object
· girl → the_ext

bought Major attribute; generates a column. Its internal argument is filled by the feature PAST; its external arguments are AGENT and THEME. Since 'buy' is a major predicate, it is the external argument of the object 'girl', and 'girl' points to the AGENT role. As a result, there is a '+' in the Relation Matrix in cells girl/AGENT and girl/buy. However, since only one of the external argument positions of the transitive verb is filled, the clause is not yet complete.

· buy → girl_ext
· girl → buy_ext (THEME)
· '+' in RM in cell girl/AGENT
· '+' in RM in cell girl/buy

some A quantifying pronoun which may function as a determiner or as an independent pronoun. We can make a unified account if we treat it as an attribute that, like the article, introduces a Proto-object; however, unlike with articles the Proto-object now also points to the external argument of the attribute. As a result, there is a '+' in the Relation Matrix in cell Proto-object/some. This explains the possibility of e.g. *She bought some*, which, indeed, is complete but has an implicit object. In this view, then, quantifying pronouns are treated as an intermediate type between the pure qualifier-class of articles and the class of adjectives, which do allow a Proto-object as their external argument.

The Proto-object also realizes the external argument THEME, causing a '+' to be placed in the appropriate cell of the Relation Matrix.

· some → Proto-object_int
· Proto-object → some_ext
· Proto-object → buy_ext (THEME)
· '+' in RM in cell Proto-object/some
· '+' in RM in cell Proto-object/THEME
· '+' in RM in cell Proto-object/buy

flowers Object; replaces the Proto-object. 's' can be regarded as part of the internal argument (qualifier). Note that this does not conflict with the fact that 'some' also is an internal argument: they are unifiable within the same domain (both can signify plural; together they are plural indefinite).

· 'flowers' replaces Proto-object

The Relation Matrix for this sentence is displayed in Fig. 4.

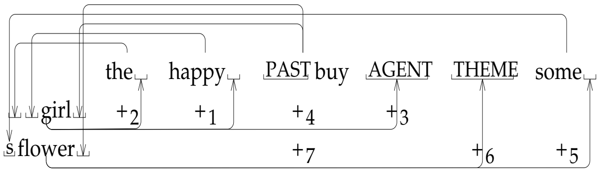

Figure 4: *The happy girl bought some flowers*

This treatment of quantifying pronouns has two important advantages. First, it does not require ambiguous lexical entries. The same can be said for demonstrative pronouns, numerals and other function words that are ambiguous between independent and adjectival use. Second, the use of Proto-objects makes it unnecessary to have a rule defining noun phrase heads as realized either by nouns, or by numerals, quantifying pronouns, demonstrative pronouns etc. In fact, this also applies to nominal adjective phrases: there is no need to define adjectives as possible realizations of noun phrase heads. The nominal adjective phrase follows naturally from the presence of the article (creating the Proto-object) and the adjective (combining with the Proto-object). Furthermore, this approach also accounts for the potential structural ambiguity of a quantifying pronoun or a nominal adjective phrase followed by a plural noun phrase, as in apposition. (Example: 'On Monday she got a big bunch of flowers. The white, lilies, wilted after a mere few days.').

Going through the sentence from left to right, we see the following structure emerge:

- At word 'happy' we obtain the nominal adjective phrase ($+_1$ and, through inheritance, $+_2$);
- At word 'girl' we obtain the noun phrase (PTR from 'girl' to the_ext);
- At word 'some' we obtain the clause with an independent pronoun ($+_5$, $+_6$ and $+_7$);
- At word 'flowers' we obtain the clause with 'some' as determiner.

As shown in these examples, the phrases and clauses can be found in the Relation Matrix. The concept lattice representation is especially valuable for the *observations*, which are essential for information retrieval (Sarbo and Farkas, 1995). Information present in the Relation Matrix is accessible from the concept lattice and can be used in the explanation of the concepts and sublattices. We note that in our application of FCA a concept containing the empty set is meaningless, because it is in conflict with relatedness. The concept lattice of Fig. 4 is shown in Fig. 5.

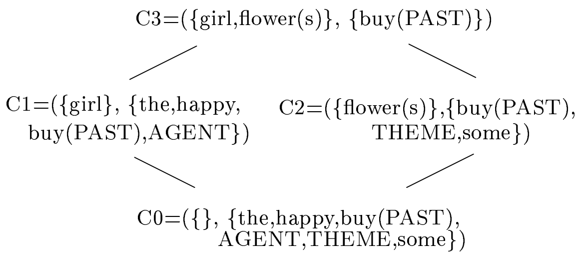

Figure 5: Concept lattice of Fig.4

For example, the concept C1 denotes the observation: a particular ('the') object ('girl') has a property ('happy') and is the AGENT participant of the action 'buy(PAST)'. The concept C3 denotes the observation that 'girl' and 'flower(s)' are related to the action 'buy(PAST)'. It is interesting that these concepts correspond to the focus of WH-questions: (C1) Who bought (something)? (C3) What happened? This suggests a potential correspondence between the linguistic device of question-formation and the information reflected in the lattice. Moreover, it exemplifies the postulate (Sarbo, 1997) that a concept lattice is an appropriate representation for what is referred to in artificial intelligence as 'cooperative communication' (Grice, 1975).

In addition to the individual concepts, a (sub)lattice also has information content, comparable to that of a clause. In this example, C0-C1-C2-C3 represents the clause. The use of sublattices is potentially relevant for interpreting discourse relations.

7 Summary and conclusions

In this paper we have focused on the underlying principles of hierarchical structure in language. We have discussed the theoretical foundations of Natural Language Concept Analysis. We have shown that hierarchical structure, which is commonly taken as given in linguistics, emerges as the result of more abstract principles relating to the combinatorial properties of linguistic units of different types. These properties derive from the inherent characteristics of lexical items and the different linguistic relations that they can take part in. The major relation, predication, has a symmetrical instantiation (predicate–participants) and an asymmetrical instantiation (predicate/participant–modifier). The minor relation distinguishes between semantic core and a qualification of that core. The application of NLCA was illustrated on the basis of examples. Current research on NLCA focuses on an elaboration of the linguistic and philosophical foundations on the one hand and algorithmic implementation on the other.

Acknowledgement

We are grateful to József Farkas for his pioneering work and inspiration in the initial stages of this project.

References

Jan Aarts. 1991. Intuition-based and observation-based grammars. In K. Aijmer and B. Altenberg, editors, *English Corpus Linguistics*, pages 44–62. Longman, London and New York.

Robert C. Berwick and Samuel D. Epstein. 1995. On the convergence of 'minimalist' syntax and categorial grammar. In A. Nijholt, G. Scollo, and R. Steetskamp, editors, *Algebraic Methods in Language Processing (TWLT 10)*, pages 143–148, Universiteit Twente, Enschede.

Helmut Franzen. 1983. Compiler generation: From compiler descriptions to efficient compilers. Bericht nr. 83 – 20, Technische Universität Berlin, May.

Norman M. Fraser. 1996. Dependency Grammar. In K. Brown and J. Miller, editors, *Concise Encyclopedia of Syntactic Theories*, pages 71–75, Pergamon, Oxford etc.

Herbert P. Grice. 1975. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and semiotics: Speech acts*, volume 3, pages 41–58, Academic Press, New York.

Liliane Haegeman. 1991. *Introduction to Government and Binding Theory.* Basil Blackwell, Inc., Cambridge, MA.

Nathan Houser and Christian Kloesel, editors. 1992. *The Essential Peirce: Selected Philosophical Writings (1867–1893).* Indiana University Press, Bloomington.

Joachim Lambek. 1988. Categorial and Categorical Grammars. In R.T. Oehrle, E. Bach, and D. Wheeler, editors, *Categorial Grammars and Natural Language Structures*, D. Reidel Publishing Company, Dordrecht-Boston.

Fritz Lehmann and Rudolf Wille. 1995. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J.F. Sowa, editors, *Third Int. Conf. on Conceptual Structures, ICCS'95.* Springer-Verlag.

James J. Liszka. 1996. *A General Introduction to the Semeiotic of Charles Sanders Peirce.* Indiana University Press, Bloomington and Indianapolis.

Charles S. Peirce. 1931–35. *Collected Papers.* Harvard University Press, Cambridge.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar.* The University of Chicago Press, Cambridge, MA.

Janos J. Sarbo. 1997. Building Sub-Knowledge Bases Using Concept Lattices. *The Computer Journal*, volume 39, no. 10, pages 868–875, Oxford University Press.

Janos J. Sarbo and Jozsef I. Farkas. 1995. Knowledge representation and acquisition by concept lattices. In Shaul Markovitch, editor, *Proc. of the 11th Izraeli Symposium on Artificial Intelligence (ISAI'95)*, Hebrew University of Jerusalem, Izrael.

John F. Sowa. 1996. Processes and participants. In P.W. Eklund, Gerard Ellis, and Graham Mann, editors, *Conceptual Structures: Knowledge Representation as Interlingua (ICCS'96)*, volume 1115, pages 1–22, Springer-Verlag.

Robert D. Van Valin, Jr. 1993. A synopsis of Role and Reference Grammar. In Van Valin, editor, *Advances in Role and Reference Grammar*, pages 1–164, John Benjamins Publishing Company, Amsterdam-Philadelphia.

Rudolf Wille. 1982. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered sets*, pages 445–470. D. Reidel Publishing Company, Dordrecht-Boston.