

Lars Borin
Uppsala University
Center for Computational Linguistics
Box 513
S-751 20 Uppsala, Sweden

A constraint-based approach to morphological analysis (preliminaries)

0. Introduction

This paper is intended to give the background to ongoing work on a constraint-based system for morphological analysis, which I intend to present in more detail later. The system represents one stage in the development of a (computational) morphological formalism suited for modelling the mechanisms of word formation, a development that began about four years ago, when I started experimenting with Koskenniemi's (1983) two-level model (Borin 1985; 1986a; 1986b). A search of the relevant literature in both linguistics and computational linguistics showed remarkable similarities in many newer approaches to language, similarities that in some ways represent a return to an older linguistic tradition. These approaches can all be said to advocate *relational* models of language, a notion that will be discussed below.

The particular approach proposed here was directly inspired by a somewhat older linguistic model, but still a relational one, namely stratificational grammar and its offshoots (see e.g. Gleason 1964; Lamb 1966; Lockwood 1972; Reich 1969; 1970), to my knowledge the most thorough attempt to formalize the structuralist notion of language as a system where everything depends on everything else, i.e. a system of relations.

In section 1 below I try to give a characterization of relational linguistic models and also to give an overview of some of the relational models found in the literature. Section 2 discusses the possibilities of using a recent artificial intelligence technique, constraint systems (Hein 1981; 1982; Maleki 1987), as a general implementation language for these models. Conclusions and some directions for further research are the topic of section 3.

1. Relational linguistic models

In the last few years, there has been a convergent development in the closely related areas of linguistics, computational linguistics, artificial intelligence and cognitive science towards relational models of language and language use. The models I am referring to are (at least), for linguistics: the Meaning $\ll = \gg$ Text model of Mel'čuk and others (Mel'čuk 1974; Mel'čuk & Pertsov 1987), autosegmental phonology (Goldsmith 1976) and morphology (McCarthy 1981; 1982); for computational linguistics: two-level morphology (phonology) and other finite-state phonological and morphological models (Koskenniemi 1983; Kay 1987) and, at least partly, some of the unification-based formalisms, like Lexical-Functional Grammar (LFG) (Bresnan & Kaplan 1982), Functional Unification Grammar (FUG) (Kay 1985) and Uppsala Chart Parser (UCP) (Sågvall Hein to appear); for artificial intelligence: associative networks, also called semantic networks and conceptual graphs (see e.g. Sowa 1984 or the articles in Findler 1979); for cognitive science: connectionist models (e.g. Dell & Reich 1980; Dell 1985).

Tentatively, we may characterize relational models of language as models with the following properties:

- relations are more important than processes in the model, this in contrast to e.g. generative grammar or Hockett's (1954) IP (Item-and-Process) model. As a rule, there is only a small number of relations in the model.

- linguistic units, or items, if they have any theoretical status at all, are defined through their relations to other units;

From the point of view of computational linguistics, this has some important consequences for the way processing is considered to be carried out in a relational linguistic model:

- the model is decentralized, in the sense that there are many autonomous processing elements. On the other hand, there are only a few element types;

- the processing elements and their interconnections (links) may be seen as a *network*, where the topology of the network is an important part of the model. Just as there are only a few element types, there is only a small number of possible link types between the elements;

1.1 Some relational linguistic models

First of all, one must mention classic Saussurean structural linguistics. Here, language is viewed as a system where every unit is defined by its place in the totality, i.e. by its relations to other units:

Units and grammatical facts would not be confused if linguistic signs were made up of something besides differences. But language being what it is, we shall find nothing simple in it regardless of our approach; everywhere and always there is the same complex equilibrium of terms that mutually condition each other. Putting it another way, *language is a form and not a substance*.

(de Saussure 1959:122, emphasis in the original)

Being in Copenhagen, I should not forego to mention Hjelmslev and glossematics in this context. The notion of language as a system of relations was very much present in Hjelmslev's work (e.g. Hjelmslev 1961), and workers in stratificational grammar usually mention him as their single most important source of inspiration.

Against this background, the more recent theories represent a return to a tradition that has lived in the shadow of the preoccupation with process-based linguistic description that has characterized much of (especially American and generative) linguistics during the last three decades or more. This does not mean that history has taken a full circle; rather, the insights of the structuralists are now combined with a formal rigour that to date has been the foremost contribution to linguistics by generative grammar and computational linguistics.

1.1.1 The Meaning $\ll = \gg$ Text model

The Meaning $\ll = \gg$ Text model concerns itself with the relation between the two entities in its name, i.e. for a given meaning, how do you get to the (very large number of) texts that express this meaning, and conversely, how do you get to the (several possible) meanings of a particular text. To this end, the model operates with seven levels of linguistic representation: semantic, deep and surface syntactic, deep and surface morphological and deep and surface phonetic. To get from one level to the next level up or down, there are interlevel relations, many-to-many mappings between the levels. These mappings have been described in various ways: in an earlier version the model (Mel'čuk 1974) as chains of ordered transducers, while at present they are held to be a set of unordered correspondence rules, which

are conceived of not as prescriptions, or instructions of an algorithm, but rather as permissions and prohibitions, or statements in a calculus.

(Mel'čuk & Pertsov 1987:35)

Since this model was proposed by Russian linguists, the lexicon has a very important role in it. There is a carefully specified format for the lexicon to be used in the model, and some actual lexicons (Mel'čuk 1984; Mel'čuk & Žolkovskij 1984) have been prepared according to this format, which all have the common trait that they hold very much information about each lexical unit. E.g. in the Russian lexicon (Mel'čuk & Žolkovskij 1984) the entry for the word *čuvstvo* 'feeling' is 16 pages long.

1.1.2 Autosegmental models

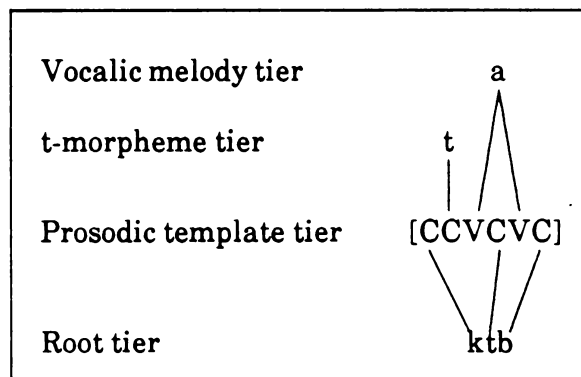
It has often been observed that some phonological phenomena are difficult to handle in a segmental phonology, because the phonological segment does not seem to be their right domain, but rather the syllable, the morph(eme) or the word. Among such phenomena are stress, tone (pitch), vowel harmony and synharmonism. Various solutions have been proposed for overcoming these difficulties; often stress and tone, at least, have been segmentalized into the phonemic string. Autosegmental phonology was developed as a reaction against linear, or segmental phonology, to deal with the phenomena of accent, tone etc., that present a problem to the segmental approach. The approaches most relevant in the present context, however, are prosodic analysis (Firth 1948; Robins 1957) and Harris' (1944) long and contour components. Unlike Harris' approach and like prosodic analysis, autosegmental phonology divides the traditional segmental phonemic level into a number of simultaneously occurring components. In autosegmental phonology all these components are segmental, something that sets them off from the prosodies of prosodic analysis. Furthermore, among the segmental components, or tiers, one has a special status. This is the traditional segmental tier, which serves as the coordinating tier, to which all the other - autosegmental - tiers are mapped by a general mapping relation, called the well-formedness condition (WFC). Autosegmental phonology is declared to be a further development of generative phonology, but it differs considerably from the latter in spirit:

Autosegmental phonology is a particular claim, then, about the *geometry* of phonetic representations; it suggests that the phonetic representation is composed of a set of several simultaneous sequences of these segments, with certain elementary constraints on how the various levels of sequences can be interrelated -- or, as we shall say, "associated."

(Goldsmith 1976:16, emphasis in the original)

Autosegmental phonology has been used to describe, e.g., tone and accent (Goldsmith 1976; Withgott & Halvorsen 1984) and vowel harmony (Clements 1980).

The formal devices of autosegmental phonology have also been used in morphology, notably by McCarthy (1981; 1982) for describing Classical Arabic and Hebrew morphology, i.e. strongly non-concatenative systems. McCarthy introduces some additional autosegmental tiers in the model. In his version, the material from the traditional segmental tier is distributed over several *morphemic tiers*, and the coordinating component, to which all other tiers are mapped, is the *prosodic template* or *CV-skeleton*, a (partly specified) phonotactic constraint. The general "geometry of the representation" will hopefully be illustrated by the following figure, the autosegmental representation of the Classical Arabic verb stem *ktatab* 'was registered' from the root *ktb* 'to write, writing', taken from McCarthy (1982:193):



1.1.3 Finite-state phonology and morphology

The by now well-known two-level formalism is the brainchild of Kimmo Koskenniemi of Helsinki University (see e.g. Koskenniemi 1983; Karttunen 1983), and it has given rise to a fair number of both applications to specific languages and similar formalisms,

collectively referred to as finite-state morphology. Koskenniemi is quite insistent on two-level morphology being a relational formalism:

The two-level formalism is neutral with respect to production and analysis because it describes morphological phenomena as relations between lexical and surface representations. The relations are seen as correspondences, not as segments being transformed into other segments.

(Koskenniemi 1983:10)

There are other variants of finite-state morphology that use more than two levels in the description, e.g. Kay's two-level morphology with tiers (Kay 1987), which is an implementation of autosegmental morphology using the finite-state transducers of two-level morphology for the WFC (see 1.1.2 above).

1.1.4 Stratificational grammar and relational grammar

Stratificational grammar (SG) was developed as a purely linguistic theory, just like its contemporary, generative grammar, but it was developed in a machine translation project, and its theoretical devices presumably were influenced by this fact. Stratificational grammar and its offshoot relational grammar (RG) see language as a network of relations. Some versions of SG do not give items any status whatsoever in the theory, stating that the items of linguistic description are simply nodes in the overall network. The network connects to items at its both ends, however - phonetic units at one end and conceptual units at the other. SG, but not RG, also holds that language is *stratified*, i.e. there are layers or strata of linguistic description, normally corresponding to the traditional linguistic divisions of language into phonology, morphology, syntax and semantics¹.

The relations allowed in a stratificational or relational description are usually taken from a small set of primitive relations, the most important being conjunction (symbolized by AND nodes in the network diagrams, see below) and disjunction (OR nodes). There are also the interstratal relations of realization ('is realized by'), composition ('is composed by') and their inverses ('is a realization of' and 'is a part of'). The number of relation types postulated varies among different authors, but at least these types, in some form, are present in all descriptions. The difference between the relations of realization and composition gives rise to two subsystems on each stratum, the realizational part and the tactics, where the latter acts as a filter on the realizations allowed by the former. Seeing language as a system of relations, stratificational grammar has no place for process description in the sense of Hockett's (1954) IP model, or in the interpretation of 'generate' in 'generative grammar' as meaning the same thing as 'produce' (this is not the interpretation intended originally, but common nevertheless); just like in the two-level model, all the strata are considered to exist side by side, simultaneously. An attractive trait in relational models is their inherent non-directionality (bidirectionality in this case); if you put in a text at one end of a relational network, a meaning or meanings will appear at the other end, and vice versa. Many stratificationalists have also found graphical network descriptions ("lambograms"; see section 3 below) to be a convenient tool, preferable to algebraic rule systems for the description of language.

1.1.5 Associative networks

Associative networks are perhaps more commonly known as semantic nets. The term associative network was introduced, as far as I know, by Findler (1979). Other terms that have been used for the same thing are semantic memory (Quillian 1968) and conceptual graphs (Sowa 1984). Associative network formalisms have been used in artificial intelligence both for general knowledge representation and for storing more specifically linguistic knowledge. Associative networks are directed graphs, with labelled nodes and edges. The common interpretation is that the nodes represent entities and the links relations between these entities; in a predicate calculus setting, the links would be the predicates and the nodes the arguments (constants and variables) of these predicates. In this connection it is worth noting that the highest -

the sememic - stratum in a stratificational grammar is considered by most stratificationists to be a structure very similar to an associative network; to avoid terminological confusion, this structure is often called a *reticulum*, since the term network is reserved for the less densely connected lower strata, where temporal relations play an important role.

1.1.6 Connectionist models

Connectionist models share many of the assumptions of the models described above, but the fundamental assumption behind them is "that theories and scientific languages based on the computational character of the brain are productive (even essential) in many areas of Cognitive Science" (Feldman 1985:1). In other words, connectionist models are based on computational architectures with many autonomous processing elements, which can be linked up in a small number of ways, and communicate with a limited repertory of (simple) signals, just like neurons. Among the other models mentioned in this section, it is perhaps stratificational grammar that has the closest affinity to connectionist models. The speech production model used by Dell & Reich (1980) is a refined variety of the relational networks earlier described by Reich (1970), and Schnelle's (1981) neurologically inspired *net linguistics* works with more or less the same element types as stratificational and relational grammar.

1.1.7 Unification-based models

Unification is a technique that has become increasingly popular in natural language processing systems. The basic data structure in unification-based grammatical formalisms is the attribute-value graph, a directed acyclic graph (DAG) made up of attributes (like *case*) which have values (e.g. *nominative*). Instead of being atomic, the values may, in turn, be attribute-value graphs. Unification is an operation for ascertaining that two (or more) attribute-value graphs are compatible with each other, which involves checking the values of identical attributes in the two graphs. If the values are atomic, they must be identical to be compatible; if they are attribute-value graphs, unification is carried out recursively on these; if one of the values is undefined, both values become identical to the defined value. If unification succeeds, the two graphs will become identical, i.e. attribute-value pairs that appeared in only one of them will now be present in the other one as well. Unification is unordered, so in order to handle the temporally ordered surface structure of language, most unification-based formalisms have two components: a context-free phrase structure grammar which is used to build a phrase structure tree from the surface morphological and syntactic representation, annotated with functional structures. These functional structures are then unified with partly specified lexical and grammatical functional structures, sometimes called *constraining equations* (e.g. in Withgott & Halvorsen 1984), to yield a fully specified functional structure as the analysis of the linguistic unit being parsed.

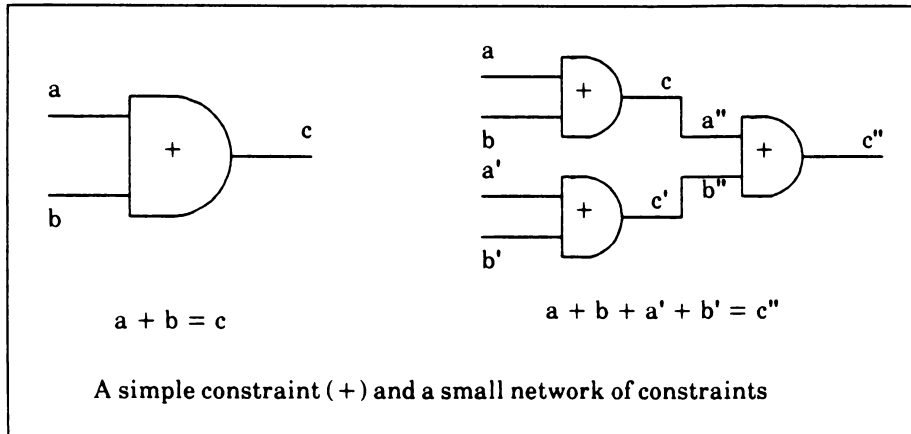
2. Constraint systems

Constraint systems is an AI technique for representing knowledge about relations among entities, values and the like. The basic building block of constraint systems is the constraint,

an active relation between a (usually small) set of objects. The relation is termed active since it exhibits two crucial features. It establishes itself as soon as enough information about the participating objects is available and it enforces the relation once it has been established. (Hein 1981:3)

The figure below is intended to serve as an illustration of the way constraint systems are set up. The left half of the figure shows a simple constraint of the equation kind ($a + \text{constraint}$), which expresses the relation $a + b = c$ in the following way. As soon as the values of at least two of the three variables in the equation are known, the third variable will automatically be set to a value that satisfies the equation. After

this, the constraint will enforce the relation by reacting to changes in the values of the variables. The exact nature of the reaction, however, is dependent on the kinds of constraints in the system and their use. The right half of the figure shows that simple constraints can be connected together - via their variables - into constraint networks. The interconnections are made via equality constraints.



Constraint systems combine the object-oriented and declarative programming paradigms. The characteristic features of both of these are generally considered important in the perspective of computational linguistics. Object-orientedness implies decentralized control: processing control is local to a constraint. This in turn means that constraint systems would be fairly easy to implement on parallel computer architectures, like connection machines (Hillis 1984). Declarative programming, on the other hand, fits well in with the relational linguistic models discussed above. The metaphors used in talking about relational linguistic models are conspicuously close to the kinds of phenomena constraint systems are supposed to be good at handling, namely geometrical and topological relationships (see the section on autosegmental models above), equations and (of course) constraints holding between objects and values. This makes one suspect that constraint systems should be fairly easy to use for implementing these linguistic models on a computer.

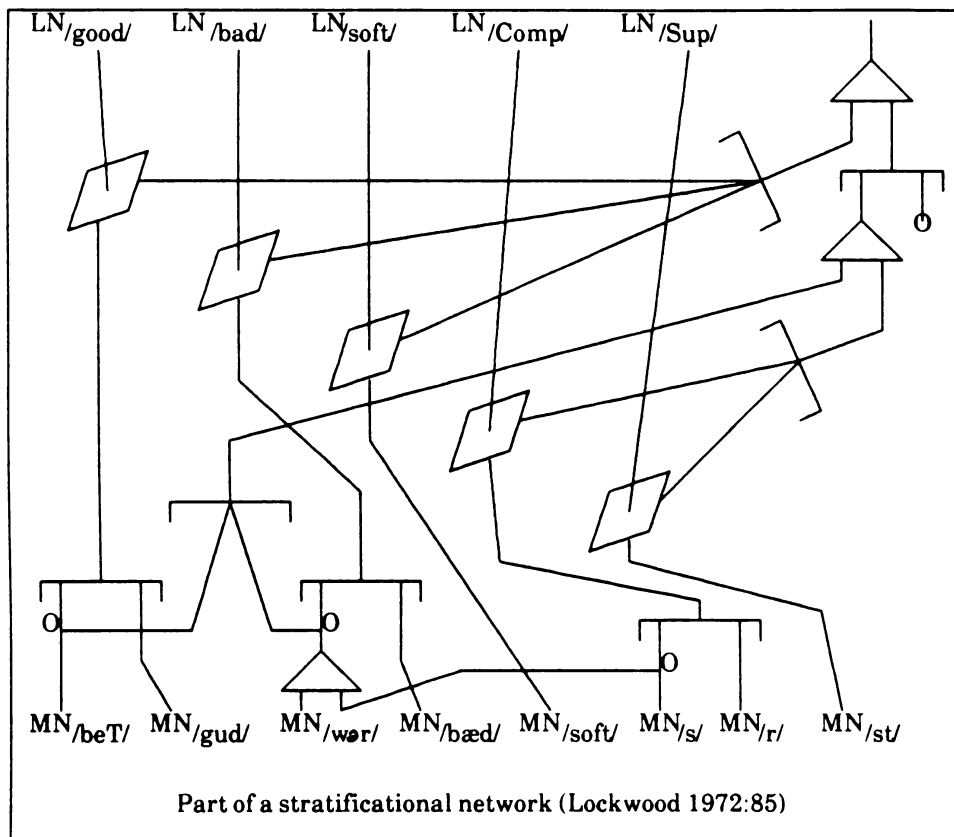
3. Conclusions and further research

Recently I have got access to an experimental constraint system implementation (ICONStraint, written by J. Maleki of Linköping University for the Xerox 11XX Lisp machines, and described in Maleki (1987)), which has made it possible for me to start experimenting with a relational model of linguistic structure, expressed as a network of constraints.

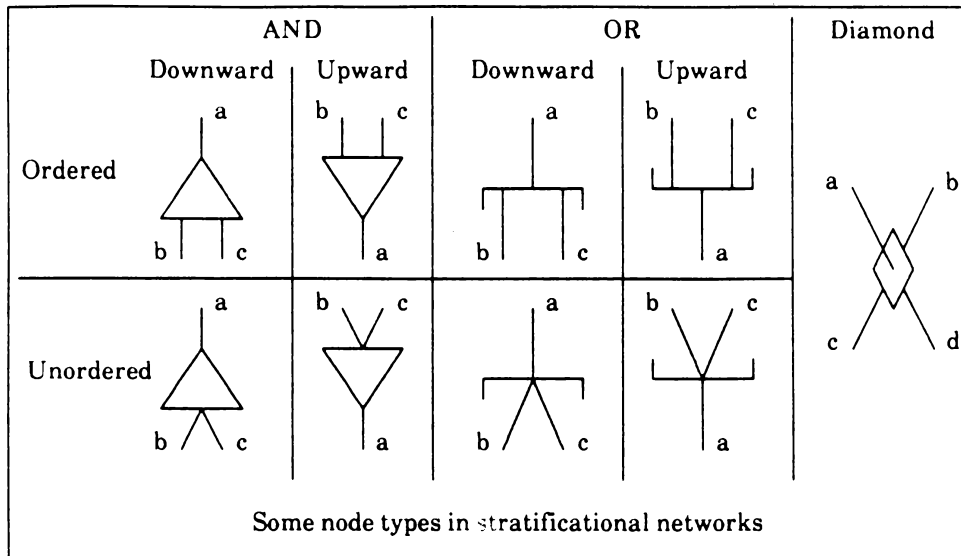
The structure of this network is heavily influenced by the networks used in presentations of stratificational grammar, mostly because this is the way of least resistance, since stratificational grammar is one of the more thoroughly formalized relational linguistic models. Its cousin, relational grammar, has been partly implemented as a computer model, Reich's (1970) *relational network simulator*. I say partly, because Reich discusses mostly language production - or encoding - while decoding is mentioned only in passing, like in most stratificational descriptions I have seen. This is one reason that I have chosen to lay the emphasis on the decoding direction in my work and consequently talk about "morphological analysis" in the title of this paper. Since more attention has been given to the encoding direction in a stratificational grammar, it seems natural to start with the structures and relations that have been shown to work in encoding and somehow reverse them to do the decoding. In the case of unrestricted rewrite rule systems, like the tree transformations used in generative grammar, this is generally not possible (King 1983), but

relational models should in principle be able to cope with the task. This is where constraint systems enter the picture. Being made up of active relations in the sense stated above (section 2), once a constraint network that models encoding has been built, the same network should work just as well for decoding purposes. An added requirement could be that the basic constraints in the network as closely as possible mimic the primitive relations (nodes) that are postulated in (some version of) stratificational or relational grammar, since it would then be possible to test published descriptions directly. This presupposes that these relations are well-defined in both directions.

It seems, however, that the latter is not true for the nodes in a stratificational grammar. This is partly due to the nature of the relations involved, and partly because of a certain vagueness in their definitions. I will discuss the last point first, but first I will make a small digression into the written format of stratificational descriptions. As I indicated above, the normal way of presenting such a description is in the form of one or more graphs ("lambograms"), for example the following graph that describes comparison of English adjectives:



The nodes always represent the same relations, while the meaning of the lines is dependent on the context; sometimes a line expresses the realization relation, sometimes it just connects one of the participants in some relation to that relation. One of the problems with the node definitions, as they are given in the literature, is that they are not detailed enough, meaning either: 1) that not all input/output combinations are accounted for, or, more often: 2) that for a given input, the output is indeterminate. The following figure shows some common node types used in stratificational networks, taken from Christie (1974).



The OR and AND nodes are discussed below. The diamond node connects the tactics and the realizational part together.

In the worst case, the nodes are given only informal definitions, like:

Another fundamental linguistic relationship is that which a class bears to its members. In stratificational terminology, this is called an OR relationship.
(Lockwood 1972:34)

Even if formal node definitions are given, they often are not quite formal enough (cf. Schreyer 1980; 1981); sometimes the formal definition of an individual node type leaves unclear some aspects of its function in the network as a whole. A notorious problem in this respect appears in the definition of the unordered OR, in the case of how the plural side depends on the singular side. The most informal definitions simply ignore the problem. Also, there are serious problems with timing, which are seldom or never discussed (see, however, Gleason 1980), but which become very real once an actual implementation is considered. There are two model-internal aspects to the timing problem. The first is the need of some synchronization mechanism in the model; as I said above, relational models are basically decentralized: there is no central processor that distributes processing tasks according to some internal state and clock, only many autonomous processing elements, working in parallel². Since there is no central clock and since some of the nodes are described as temporal, there must be other means of synchronizing signals in the network. The other problem is in some ways dependent on the first: how do you define the relevant temporal units to use in the network and their interrelations? For example, it is said that the ordered AND describes the "important relationship in linguistic structure [...] of a combination to its [...] constituents or components" (Lockwood 1972:31), when "the order of constituents is significant" (*ibid.*). Implicit in most, if not all, definitions of the ordered AND is not only that the constituents on the plural side are temporally ordered, but also that they are temporally *contiguous*. Then it becomes important to define temporal contiguity; is it absolute, defined in terms of some minimal temporal unit, or is it relative to some events in the network? Another notion that is in need of a definition before the relations can be modelled, is that of *simultaneity*; the unordered AND node is described as one where the constituents appear "simultaneously or in no specified order" (Lockwood 1972:33). Also, constraint systems have been used mostly to represent atemporal relations, due to the nature of the problems they have been used to solve. This must not necessarily be the case; the author of the ICONStraint system has indicated the need for modelling time and change within the constraints paradigm (Maleki 1987:81).

These are the two problems that I am concentrating on at the moment: the formal node definitions and the introduction of time into the implementation language.

Acknowledgement

The research described in this paper would have been much more difficult to carry out without the ICONStraint programming system developed by Jalal Maleki of Linköping University, who kindly put the system at my disposal.

Notes

¹For another relational theory of language, where language is stratified, and the important relations "realized by" and "composed of" are kept more consistently apart than in stratificational grammar, see e.g. Sgall *et al.* (1969).

²Nothing changes, in principle, even if the actual implementation, like the one described here, is made on a serial machine with one central processor, i.e. the parallelism is simulated.

References

- Borin, L. 1985. *Tvånivåmorfologi: Introduktion och användarhandledning*. UC DL-L-3. Uppsala University, Center for Computational Linguistics.
- Borin, L. 1986a. What is a lexical representation? Karlsson, F. (ed.): *Papers from the Fifth Scandinavian Conference of Computational Linguistics*. Publications, No. 15. University of Helsinki, Department of General Linguistics.
- Borin, L. 1986b. *Swedish Two-level Morphology: Some Remarks*. UC DL-R-86-1. Uppsala University, Center for Computational Linguistics.
- Bresnan, J. & R.M. Kaplan 1982. Lexical-functional grammar: a formal system for grammatical representation. Bresnan, J. (ed.): *The Mental Representation of Grammatical Relations*. Cambridge, Massachusetts - London.
- Christie, W.M. Jr. 1974. A stratificational view of linguistic change. Anderson, J.M. & C. Jones (eds.): *Historical Linguistics I*. Amsterdam.
- Clements, G.N. 1980. *Vowel Harmony in Nonlinear Generative Phonology: An Autosegmental Model*. Indiana University Linguistics Club. Bloomington, Indiana.
- Dell, G.S. 1985. Positive feedback in hierarchical connectionist models: applications to language production. *Cognitive Science*, Vol. 9, No. 1.
- Dell, G.S. & P. Reich 1980. Toward a unified model of slips of the tongue. Fromkin, V.A. (ed.): *Errors in Linguistic Performance*. New York.
- Feldman, J. 1985. Connectionist models and their applications: Introduction. *Cognitive Science*, Vol. 9, No. 1.
- Findler, N.V. (ed.) 1979. *Associative Networks*. New York.
- Firth, J.R. 1948. Sounds and prosodies. *Transactions of the Philological Society 1948*. Oxford. Reprinted in Makkai 1972.
- Gleason, H.A. Jr. 1964. The organization of language: a stratificational view. *Report on the 15th Annual (First International) Round Table Meeting on Linguistics and Language Studies*. Georgetown University Monograph Series on Languages and Linguistics, No. 17. Washington, D.C.
- Gleason, H.A. Jr. 1980. Comments on William J. Sullivan's Syntax and Linguistic Semantics in Stratificational Theory. Kac, M. (ed.): *Current Syntactic Theories: Discussion Papers from the 1979 Milwaukee Syntax Conference*. Indiana University Linguistics Club. Bloomington, Indiana.
- Goldsmith, J. 1976. *Autosegmental Phonology*. Indiana University Linguistics Club. Bloomington, Indiana.
- Harris, Z.S. 1944. Simultaneous components in phonology. *Language* 20. Reprinted in Makkai 1972.
- Hein, U. 1981. *A Short Description of CMI - a Programming System Based on Constraints*. AILAB Working Paper No. 2. Artificial Intelligence Laboratory, Software Systems Research Center, Linköping University.

- Hein, U. 1982. *Constraints and Event Sequences*. Research Report LiTH-MAT-R-82-02. Software Systems Research Center, Linköping University.
- Hillis, D. 1984. The connection machine: a computer architecture based on cellular automata. *Physica* 10D. Amsterdam.
- Hjelmslev, L. 1961. *Prolegomena to a Theory of Language*. Madison, Wisconsin.
- Hockett, C.F. 1954. Two models of grammatical description. *Word*, Vol. 10, No. 2-3.
- Karttunen, L. 1983. KIMMO: A general morphological processor. *Texas Linguistic Forum*, No. 22.
- Kay, M. 1985. Parsing in functional unification grammar. Dowty, D.R., L. Karttunen & A.M. Zwicky (eds.): *Natural Language Parsing*. Cambridge.
- Kay, M. 1987. Nonconcatenative finite-state morphology. *Third Conference of the European Chapter of the Association for Computational Linguistics*. Copenhagen.
- King, M. 1983. Transformational parsing. King, M. (ed.): *Parsing Natural Language*. London.
- Koskenniemi, K. 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. Publications of the Department of General Linguistics, University of Helsinki, No. 11.
- Lamb, S. 1966. *Outline of Stratificational Grammar*. Washington, D.C.
- Lockwood, D.G. 1972. *Introduction to Stratificational Linguistics*. New York.
- Makkai, V.B. (ed.) 1972. *Phonological Theory*. New York.
- Maleki, J. 1987. *ICONStraint, A Dependency Directed Constraint Maintenance System*. M.Sc. thesis at Linköping University. LiU-Tek-Lic 1986:11. Linköping.
- McCarthy, J. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry*, Vol. 12, No. 3.
- McCarthy, J. 1982. Prosodic templates, morphemic templates and morphemic tiers. van der Hulst, H. and N. Smith (eds.), *The Structure of Phonological Representations* (Part I). Dordrecht - Cinnaminson.
- Mel'čuk, I.A. 1974. *Opyt teorii lingvističeskix modelej "Smysl $\Leftarrow = \Rightarrow$ Tekst"*. Moskva.
- Mel'čuk, I.A. 1984. *Dictionnaire explicatif et combinatoire du français contemporain (Recherches lexico-sémantiques)*. Montreal.
- Mel'čuk, I.A. & N. Pertsov 1987. *Surface Syntax of English*. Amsterdam - Philadelphia.
- Mel'čuk, I.A. & A.K. Žolkovskij 1984. *Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka /Explanatory-Combinatorial Dictionary of Contemporary Russian/*. Wien.
- Quillian, M.R. 1968. Semantic memory. Minsky, M. (ed.): *Semantic Information Processing*. Cambridge, Massachusetts.
- Reich, P. 1969. The finiteness of natural language. *Language*, Vol. 45, No. 4.
- Reich, P. 1970. *A Relational Network Model of Language Behavior*. University of Michigan Ph.D. Distributed by University Microfilms International, Ann Arbor, Michigan.
- Robins, R.H. 1957. Aspects of prosodic analysis. *Proceedings of the University of Durham Philosophical Society* 1. Reprinted in Makkai 1972.
- Sågvall Hein, A. to appear. Parsing by means of Uppsala Chart Processor (UCP). To appear in: Bolc, L. (ed.), *Natural Language Parsing Systems*. Berlin - New York.
- de Saussure, F. 1959. *Course in General Linguistics*. Ed. by Charles Bally and Albert Sechehaye, in collaboration with Albert Riedlinger. Translated, with an introduction and notes by Wade Baskin. New York - Toronto - London.
- Schnelle, H. 1981. Elements of theoretical net-linguistics. Part 1: Syntactical and morphological nets - neuro-linguistic interpretations. *Theoretical Linguistics* Vol. 8, No. 1-3.
- Schreyer, R. 1980. The definition of nodes in a stratificational grammar. McCormack, W.C. & H.J. Izzo (eds.), *The Sixth LACUS Forum 1979*. Columbia, South Carolina.
- Schreyer, R. 1981. The linearization of speech. Some basic problems. Copeland, J.E. & P.W. Davis (eds.), *The Seventh LACUS Forum 1980*. Columbia, South Carolina.
- Sgall, P., L. Nebeský, A. Goralčíková & E. Hajičová 1969. *A Functional Approach to Syntax in Generative Description of Language*. New York.
- Sowa, J.F. 1984. *Conceptual Structures*. Reading, Massachusetts.
- Withgott, M. & P.-K. Halvorsen 1984. *Morphological Constraints on Scandinavian Tone Accent*. Report No. CSLI-84-11. Center for the Study of Language and Information, Stanford University.