Functionality in grammar design

Anders Søgaard
Center for Language Technology
University of Copenhagen
Njalsgade 80
DK-2300 Copenhagen, Denmark
anders@cst.dk

Petter Haugereid
Dpt. of Language and Comm. Studies
Norwegian University of Science
and Technology, Dragvoll
N-7491 Trondheim, Norway
petterha@hf.ntnu.no

Abstract

The implementation of the Scandinavian Grammar Matrix gave rise to a number of methodological and theoretical discussions about the desiderata for a formal theory of natural language grammar. In this paper, a strong hypothesis of *the functionality of grammar* is presented. Functionality is imposed on constraint-based grammars, on their lexicons and on the grammars themselves. It is demonstrated how this radically reduces complexity, i.e. the recognition problem is in PTIME. Certain aspects of the hypothesis are motivated by methodological and psycholinguisic considerations too. The implementation employs underspecification techniques, type inference and some amount of constructional specification.

1 Credits

Our joint work was supported by the Scandinavian Network of Grammar Engineering and Machine Translation (NorFa) and conducted at various institutions: Bremen Universität, Copenhagen Business School, Göteborg Universitet, and the Norwegian University of Science and Technology in Trondheim. In the weeks and months spend at these institutions, we benefited from discussions with a number of people, incl. Lars Ahrenberg, Felix Bildhauer, Dan Flickinger, Lars Hellan, Per Anker Jensen and Stefan Müller.

2 Introduction

We begin with a text-book definition of a mathematical function:

> *In mathematics, a function is a relation, such that each element of a set is associated with a unique element of another (possibly the same) set.*

It is in this sense we will speak of functionality. The next preliminary step is to bring to mind how natural language grammar is often defined as "a finite set of computable functions on signs". If functionality is understood mathematically, many interesting consequences can be derived from this definition. In fact, it turns out that this is a very strong hypothesis about the nature of natural language grammars; a hypothesis few (if any) grammatical frameworks pursue in full.

In this paper, we are interested in two different applications of functionality. Functionality is implemented at two levels of natural language grammar, so to speak; the lexicon and the grammar itself. In other words, the lexicon is designed such that a unique phonological string is associated with a unique lexical entry. Similarly, the grammar only produces one, possibly underspecified, analysis per unique string.

The structure of the paper is simple. In the next section, the computational and methodological advantages of functionality are demonstrated. It is shown that the recognition problem of fully functional (and off-line parsable) constraint-based grammars is in PTIME, compared to the NP completeness results of Berwick (1982) and Trautwein (1994) for off-line parsable, but non-functional constraint-based grammars without lexical rules. The reader may suspect that by imposing this functionality constraint on the lexicon, complexity is merely moved from the lexicon into the rules. This can be avoided in constraint-based grammar formalisms with typed feature structures. The relevant mechanisms are illustrated by a discussion of the syntax of the prenominal field in Mainland Scandinavian. The discussion focuses on distribution, modification and quantification. The next section presents an approach to argument structure under this kind of functionality. Since the lexicon (in the worst case) only contains sound-meaning pairs and closed class items, argument structures are inferred in syntactic composition. Some possible objections to functionality are mentioned that all relates to ambiguity, and a novel argument that (some) natural languages are non-context-free is given. The last section briefly discusses the role of syntax as perceived here in a system of natural language understanding. All data is from the Scandinavian Grammar Matrix (Søgaard and Haugereid, 2005); see `www.cst.dk/anders/matrix/main.html`.

3 Complexity of constraint-based grammars

Of the set of constraint-based grammar formalisms (PATR, LFG, ...), our interest lies with those formalisms which employ typed feature structures, e.g. LKB-HPSG (Copestake, 2001) and other computational variants of HPSG. It is a common feature of constraint-based formalisms that they combine a generative-enumerative backbone with a model-theoretic perspective on well-formedness of structures described in terms of some feature logic. The model-theoretic perspective tells us that the set of well-formed strings of a grammar with principles π is the strings for which the generative-enumerative backbone generates analyses that satisfy π. The most common backbone is a context-free grammar. Consequently, a *typed* constraint-based grammar is a tuple $\mathcal{T} = \langle\langle\{\tau_1, \ldots, \tau_n\}, \sqsubseteq\rangle, \pi, \mathcal{C}\rangle$, where $\langle\{\tau_1, \ldots, \tau_n\}, \sqsubseteq\rangle$ is a partial order on a set of typed feature structures $\{\tau_1, \ldots, \tau_n\}$.

Definition 3.1. A context-free grammar is a tuple $\mathcal{C} = \langle C, s, W, R, L\rangle$ where C is a finite set of categories (feature structures), $s \in C$ is the distinguished start symbol, W is a finite set of words, $R \subseteq C \times C^*$ is a set of grammar rules, and $L \subseteq C \times W$ is a lexicon.

Consider only grammars for which $\epsilon \notin L(\mathcal{G})$, in which all rules are unary or binary, and in which no cyclic unary extensions are possible. This is an *off-line parsability* constraint. Call an off-line parsable context-free grammar a CFG$^-$. In constraint-based formalisms, off-line parsability ensures the decidability of recognition, e.g. Johnson (1988). Call a CFG$^-$ with a functional lexicon a CFG$^-_{lx}$.

Definition 3.2. A CFG$^-_{lx}$ is a tuple $\mathcal{C} = \langle C, s, W, R, lx\rangle$ where C is a finite set of categories (feature structures), $s \in C$ is the distinguished start symbol, W is a finite set of words (and $\epsilon \notin W$), R is a set of binary grammar rules, and $lx : W \longrightarrow C$ is a lexicon. $\mathcal{C}$ is off-line parsable.

CFG^-_f is the subset of CFG$^-_{lx}$ where each string in the language receives exactly one parse, i.e. there is a function $f : \sigma \longrightarrow \tau_s$ for every grammatical string σ s.t. $\tau_s \sqsupseteq s$.

Three classes of grammars are now defined. The classes have different properties. For instance, a non-functional lexicon means multiple input for the combinatory parsing procedure, since the lexical resolution results in a disjunction of possible inputs. Consequently, if l is the number of lexemes associated with a string of n length in the language of CFG$^-$ and CFG$^-_{lx}$, then if the complexity of CFG$^-_{lx}$ is c, the complexity of CFG$^-$ is $(l-n)c$.

Trautwein (1994) shows that a restricted version of constraint-based HPSG, roughly the off-line parsable fragment corresponding to CFG$^-$ in LKB-HPSG with no lexical rules, is NP complete.

The standard NP hardness proof for constraint-based formalisms relies on a translation of the 3SAT problem into the feature logic. The correspondence depends on assignments, i.e. $g : \phi \longrightarrow \bot/\top$ translates into $L \subseteq C \times W$, truth preservation, i.e. reentrancies, and preservation of satisfiability. In other words, "the ability to have multiple derivation trees and lexical categorizations for one and the same terminal item plays a crucial role in the reduction proof" (Berwick, 1982). The standard proof does not apply to a CFG_f^- constraint-based grammar. It is thus natural to ask if there is a tractable (PTIME) parse algorithm for constraint-based grammars with restricted backbones, i.e. CFG_f^-?

Since weak subsumption of typed feature structures (without disjunction or negation) is polynomial (Dörre, 1991), the recognition problem of CFG_f^- constraint-based grammars with simple typed feature logic, say $\mathcal{T}_f$, is also in PTIME.

Theorem 3.3. *The recognition problem of $\mathcal{T}_f$ is in* PTIME.

Proof. This follows immediately from the polynomial nature of satisfiability in standard feature logic where unification is defined wrt. weak subsumption. □

$\mathcal{T}_f$ provides a tractable, but very restrictive grammar formalism. Since it is a trivial task to write a tractable grammar formalism, the next sections are devoted to demonstrating the *adequacy* of $\mathcal{T}_f$. First, however, let us briefly address the psychological plausibility of a strong functionality hypothesis.

Functionality has vast consequences for the lexicon. For instance, any open-class lexeme must be underspecified with respect to syntactic category, if there is a remote possibility that at some point it is used in the clothings of two different syntactic categories. Since this is very likely for most open-class items, the functional lexicon generally underspecifies open-class items for syntactic categories. The intuition is that the closed-class items coerce the open-class items into nouns, verbs and adjuncts in syntactic composition.

Functionality is thus falsifiable and is open to psycholinguistic evidence that the lexicon includes category information at the grammatical level, i.e. that the verbal and nominal associations cannot be reduced to the ontology of the referents. A recent article by Collina et al. (2001) reviews the literature on the lexical organizations of (what functions as) nouns and verbs.

It is often remarked in the literature that agrammatic patients experience greater difficulties in the production of verbs that in the production of nouns. Syntactic and semantic explanations of this category specific deficit have been proposed. Some influential studies have reported that dyslexics showed *no* category dissociation when presented with nouns and verbs of matched imageability and frequency. Such a claim supports our view of the lexicon. The dissociation comes from ontological concerns and language use. Similar studies have proposed that patients' performance reflects an object-action distinction. This again supports our view of the lexicon. Collina et al. (2001) show that there is a relational noun deficit too, i.e. argument structure or relationality is important for the performance of agrammatic patients. If the deficit is in fact due to argument structure or (ontological) relationality seems an unsettled issue. See Borer (2005) for more linguistic and philosophical arguments for a functional lexicon.

4 Linearized modification and quantification

The prenominal field in Mainland Scandinavian comes in the following set of configurations:

(1) (a) $\text{Quantifier}^? \ll \text{Demonstrative}^? \ll \text{Possessive}^? \ll \text{Numeral}^? \ll (\text{Adjective}^*) \ll \text{Noun}^?$

(b) alle disse mine fem røde bamser ('all these my five red teddybears')

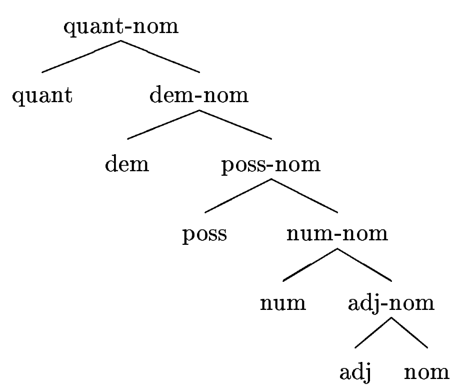
The point is that all constituents are optional, but the order is fixed. The configuration of the prenominal field can be analyzed in two ways, roughly. The first analysis is a right branching analysis, while the other bundles the prenominal elements one way or another; cf. Neville (2003). A conventional CFG posits 10 rules to account for (1). Since the category labels in the CFG backbone of an HPSG are feature structures, complexity moves from the rules into the lexical entries. Conventionally, the HPSG Specifier Head Schema and Modifier Head Schema are used to collapse the 10 rules into two, e.g.:

(2) $X"[\text{SPR}\ \langle\rangle] \longrightarrow \boxed{1}Y"[\text{SPEC}\ \boxed{2}] + \boxed{2}X'[\text{SPR}\ \langle\boxed{1}\rangle]$

The schema subsumes both the right branching and the bundling approach. Interestingly, they seem to be equally complex, roughly. The complexity does not come from the number of rules now, but from the number of lexical entries. This is easily illustrated. If no specifier phrase is assumed, only one constituent can function as the specifier of the noun phrase, but since – according to (1) – the specifier can be of any category licensed in the prenominal field, a demonstrative pronoun, for instance, must be listed as both a specifier and a non-specifier. On this right branching analysis, the total number of lexical entries for our five word fragment in (1b) is 11. On the bundling analysis, the number is 9.

If type inference is exploited, it is possible to clean up this redundancy. In addition, the approach can be extended to adjuncts. Crucially, the type hierarchy below is employed:

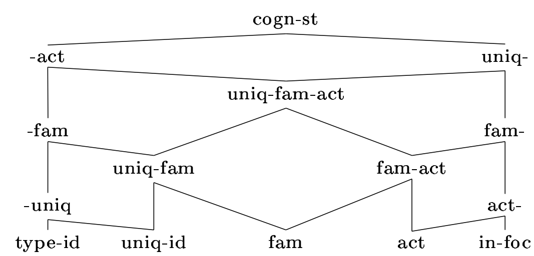
(3)



Introduce now some feature β s.t. a prenominal type α_i constrains the type it combines with (to the right) to be of type $\beta(\alpha_i) = \alpha_j$. The appropriate constraints are easily implemented relative to the type hierarchy, e.g. $\delta(\text{quant}) = \text{dem-nom}$, $\delta(\text{dem}) = \text{poss-nom}$, etc. The exceptions to this systematic behavior are easily added, e.g. the constraints imposed by articles and definite inflections. Søgaard and Haugereid (2004; 2005) describe how this account of the prenominal field fits into a general account of the distributional syntax of Mainland Scandinavian noun phrases.

Since there is no way in (1) to know which elements introduce quantification and determination, our setup calls for a constructional account of such phenomena. Our solution is to basically treat quantification and determination as agreement phenomena. Consider, for illustration, the givenness hierarchy of the Scandinavian Grammar Matrix: (The type names and their model-theoretic interpretation are introduced in Borthen and Haugereid (to appear).)

(4)



The intuition here is that while the givenness value of a nominal percolates up the noun phrase tree, it is forced down this hierarchy in search of the greatest lower bound of all the prenominal constituents' givenness values. At the constructional level, i.e. the mother node of the noun phrase, an actual givenness predication is then inserted. Similarly for quantification.

The constructional approach is not just an artefact of our design, motivated only by efficiency, but results in a linguistically more adequate treatment of quantification and determination, we claim. Our arguments for a constructional account of quantification relate to floating quantifiers, measure phrases (Søgaard and Haugereid, 2004), number agreement and anaphora resolution (Søgaard, to appear). Our arguments for a similar account of determination relate to the behavior of light pronouns and bare singulars (Borthen and Haugereid, to appear), and declension, vocatives and specification in logical form (Søgaard and Haugereid, 2004; Søgaard and Haugereid, 2005).

The conceptualization of grammar that is proposed here is quite simple: A grammar determines word order (by partial orders), agreement (by unification) and linking. Linking is discussed in the next section and is more configurational than the "linearized" structure of the prenominal field. The linearized structure is found in Mainland Scandinavian with modification more generally. Consider also the distribution of adverbs, e.g. in the Danish *Nachfeld* (Nimb, 2005):

(5) Manner or free adverbs* ≪ Predicative adverbs* ≪ Bound adverbs* ≪ Free adverbs*

5 Underspecification of argument structure

Åfarli (2005) claims that verbs in a language like Norwegian can have five different argument frames, an intransitive, a transitive, a ditransitive, a resultative and a ditransitive-resultative frame. These frames are illustrated in (6-10). Åfarli also points out that some verbs ('danse' *dance*) are found with all these frames.

(6) *Marit grubler.*
Marit ponders
'Marit ponders.'

(7) *Marit kasta steinen.*
Marit threw stone.DEF
'Marit threw the stone.'

(8) *Marit innvilga oss lånet.*
Marit granted us loan.DEF
'Marit granted us the loan.'

(9) *Marit la arket på bordet.*
Marit put sheet.DEF on table.DEF
'Marit put the sheet on the table.'

(10) *De puster oss dårlig ånde i ansiktet.*
they breathe us bad breath in face.DEF
'They breathe us bad breath into our face.'

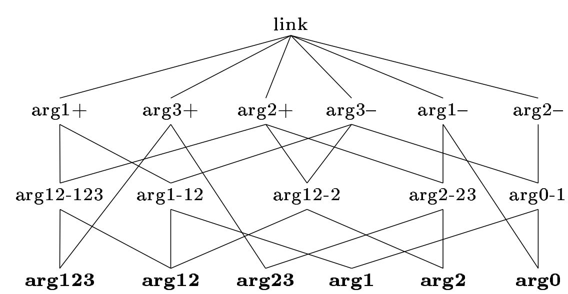
Haugereid (2004) suggests to encode linking information in the syntax rather than in the lexicon, using a neo-Davidsonian semantics and letting valence rules introduce relations which identify the relation between the main event and the index of the argument. Consequently, a arg1-rule, an arg2-rule, etc., and a set of single-valued features (ARG1, ARG2, etc.) were introduced. It is described below how this leads to a satisfactory account of Åfarli's valence frames. Ergatives and null-verbs are also treated as frames. For expository reasons, resultative frames are omitted here.

It is assumed that phrases which mark the outer boundary of a clause (like the head-filler phrase) constrain the verb projection to have *minus* link types as values of the valence features. In (11) it is shown how the ARG1 feature of the head daughter has a *arg1–* value, the ARG2 feature has a *arg2–* value and the ARG3 feature has a arg3– value.

(11)
$$\begin{bmatrix} \textit{head-filler-phrase} & \\ \text{SYNSEM} \mid \text{NON-LOCAL} \mid \text{SLASH} & \langle ! \ ! \rangle \\ \text{H-DTR} \mid \text{SS} & \begin{bmatrix} \text{LOC} \mid \text{CAT} \mid \text{VAL} & \begin{bmatrix} \text{ARG1} \mid \text{LINK} & \textit{arg1–} \\ \text{ARG2} \mid \text{LINK} & \textit{arg2–} \\ \text{ARG3} \mid \text{LINK} & \textit{arg3–} \end{bmatrix} \\ \text{NON-LOCAL} \mid \text{SLASH} & \langle ! \ \boxed{1} \ ! \rangle \end{bmatrix} \\ \text{NON-HEAD-DTR} \mid \text{SYNSEM} \mid \text{LOCAL} & \boxed{1} \end{bmatrix}$$

When a valence rule applies, it changes the *minus* link type in the mother to a *plus* link type in the head daughter as shown in Figure 1. For example, the arg1-rule changes the *arg1–* type in the mother node to *arg1+* in the head daughter.

The tree in Figure 1 shows how information about which valence rules have applied, ends up in the verb word. The verb word will now unify the linking types (13). In the case of *Avisen leser Marit* ('The newspaper Marit reads'), the types *arg1+*, *arg2+* and *arg3–* are unified. The type hierarchy of linking types in (12) then ensures that we get a transitive frame *arg12*.

(12)


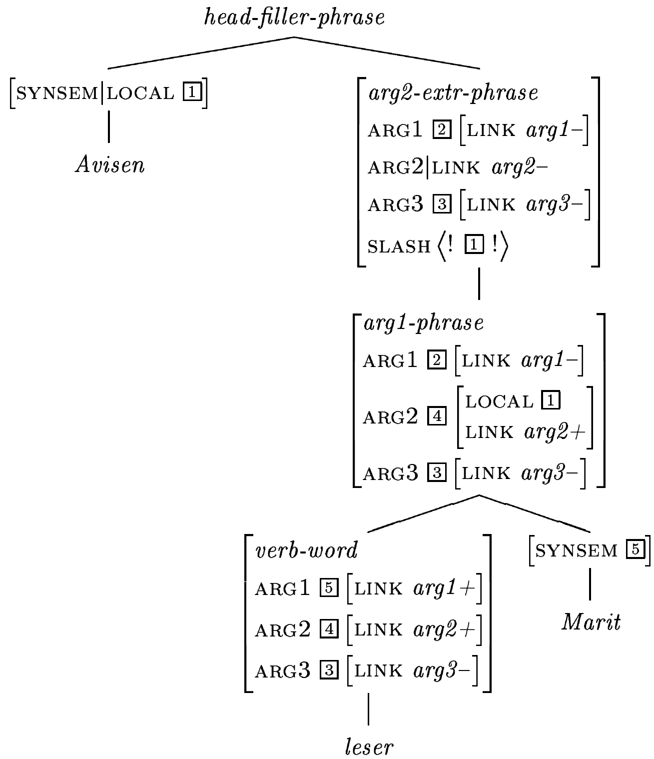
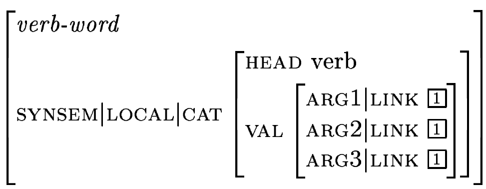
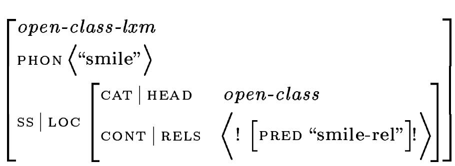


Figure 1: *Avisen leser Marit*, one reading.

(13)


Open class lexemes can be underspecified, e.g. *smile* in (14). This makes it possible for the same lexical entry to enter all kinds of argument frames, and it can be the input of a verb word inflectional rule as well as a noun word inflectional rule. The head value *open-class* is compatible with *adjective*, *noun* and *verb*.[1]

(14)


If one on the other hand wants to stipulate that a verb like *burn* can be both agentive-transitive and ergative, one can do that by giving it an *intermediate* link type like *arg12-2* in (12). That will make the lexical entry compatible with the *arg12* frame, which is the active version of the transitive verb, and the *arg2* frame, which is the ergative frame.

5.1 Ambiguities

The obvious question if one wants to maintain a functional grammar, is how to represent natural ambiguities? Our answer is equally obvious: underspecification. The technicalities are not always straight-forward, however. This is the main concern of a grammar writer, as we see it, to construct adequate type hierarchies. The hierarchies are supposed to reflect all sorts of generalizations, e.g. in the Scandinavian Grammar Matrix the type hiearchy directly encodes cross-linguistic generalizations. On the other hand, a type hierarchy must also provide an underspecified type whenever it is relevant to obtain functionality in the language-specific component.

The analysis in Figure 2 obviously subsumes the analysis in Figure 1 when the complements

[1] Selection of a direct object of a certain case, for instance, is still possible, since unary relations subsume binary and ternary ones. Consequently, the type hierarchy can be used to enforce, say, a dative direct object, but only in the case there is one.

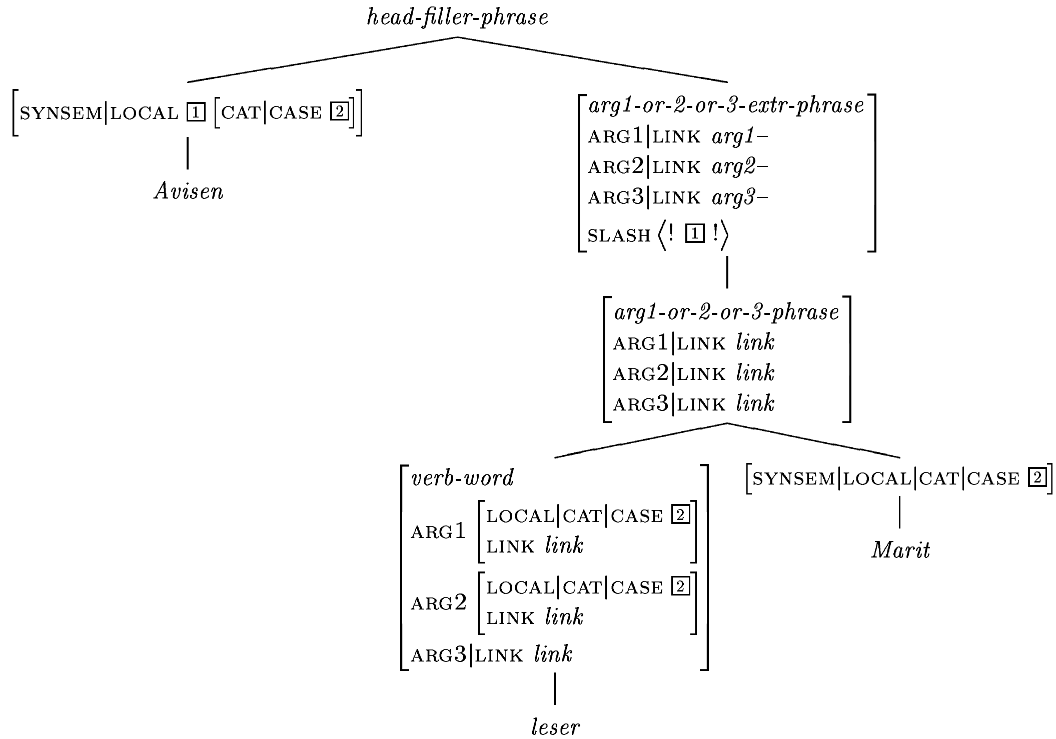
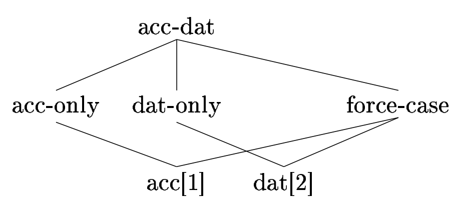


Figure 2: *Avisen leser Marit*, underspecified.

are unmarked. Consequently, if no further restrictions are imposed, *Avisen leser Marit* would result in three parses, two specified ones and the underspecified analysis in Figure 2. How is this avoided? (15) presents a rather complex alternation of the encoding of case to account for similar ambiguities with scrambling in Western Germanic, but in Scandinavian it is actually a lot easier, since only pronouns are case-marked. Consequently, the specified rules only apply to pronominal complements. The underspecified rule of course has more specified subtypes, so that the placement of sentence adverbials, for instance, can coerce a particular reading.

(15)



(16) [*acc* CASE-TYPE 1]

In (15), acc[1] is an abbreviation of (16). The case types are lexically declared, e.g. proper names are specified to be CASE-TYPE = 3, but only in rules that enforce case by force-case CASE-TYPE values are relevant, i.e. proper names are still compatible with acc-only and dat-only. The specific arg1- and arg2-rules are thus incompatible with proper names and most common nouns, i.e. an underspecified rule combines verbs and their non-case-marked complements. However, the underspecified phrase can be further specified by context (e.g. the position of sentence adverbials in Danish). This is why it is important that non-case-marked nominals are still compatible with acc-only and dat-only. In this way, functionality is maintained, since the case types are incompatible.

A third alternative, and perhaps a more elegant one, is to stipulate one underspecified rule with an underspecified valence-type, which subsumes a hierarchy of different valence types. The valence type can then be further specified by its context. So, for instance, if the rule applies for the first and only time, and the verb is agentive, the valence type will be forced into an arg1-val. This of course complicates the type hierarchies somewhat, but the underspecified rule seems rather elegant.

Similarly, attachment ambiguities must be addressed. This issue is ignored here, though, since the topic is already covered in the literature, e.g. Chrysmann (2004). A more serious problem would be functor-argument ambiguities, but it is not clear to us if such actually exist (at least not if punctuation is said to serve as sentence type classifiers).

5.2 Another argument for non-context-freeness

The feature logic of LKB-HPSG of course extends the expressivity of context-free grammars. Some evidence for the non-context-freeness of natural languages, i.e. that there are constructions in natural language that cannot be generated by a context-free grammar, has been presented, for instance by Shieber (1985). If functionality, in our sense, is found to be a realistic requirement for models of natural language grammar, another argument for non-context-freeness can be made. The reason is that certain context-free languages are inherently ambiguous, i.e. there is no context-free grammar that can derive all the strings of one of these languages unambiguously. Consider, for instance, the two languages $L_1 = \{a^n b^n c^m | n, m > 0\}$ and $L_2 = \{a^n, b^m, c^m | n, m > 0\}$. Any grammar that generates $L_1 \cup L_2$ is inherently ambiguous (Sudkamp, 2005). Such a language translates into a natural language where you have (i) case-marking, (ii) SVO and OVS constructions, and (iii) nominative objects. In Mainland Scandinavian, only the two first requirements are satisfied. Constructions with nominative objects are attested in other languages, however, and include the Icelandic copula construction (17) and various constructions in Korean, e.g. with verbs of direct perception (18):

(17) *Hún* *spurði* *hvort* *sá*
she asked whether the.NOM
grunaði *væri* *örugglega*
suspected.NOM was.3SG.SUBJ surely
þú.
you.SG.NOM
'She asked whether the suspect surely was you.'

(18) *Minwu-ka* *umak-i/*ul* *tulli-ess-ta*
Minwu.NOM music.NOM/*ACC hear.PST.DECL
'Minoo heard the music.'

Icelandic also satisfies (i), obviously, and (ii). In Korean, (ii) is paralleled by the co-existence of SOV and OSV constructions (SOV is canonical). Such SOV-OSV ambiguities are also found in West Germanic subordinate clauses. Consequently, both languages are non-context-free, if natural language grammar is fully functional.

6 Parsing, generation and inference

What are the consequences of functionality for a natural language processing system? The amount of underspecification complicates the type declarations considerably, and the information addressed by the syntactic component should thus be as limited as possible. Selectional restrictions, anaphora resolution and ellipsis, for instance, are treated in terms of *inference.* On our view, grammatical composition is interfaced with two inference components, a monotonic and a non-monotonic one. The monotonic component performs highly efficient, taxonomic inferences which are to some extent subconscious. The studies of Søgaard (to appear) and Hardt (to appear) suggest that preliminary model generation also takes place at this step of natural language processing and is in fact crucial to the resolution of anaphora and ellipsis. The non-monotonic component is conscious higher-order cognition and performs more complicated pragmatic accommodation.

Is there functionality, is it a relevant notion, in these components? The distinction between the monotonic component and the non-monotonic one is important here. On our view, the non-monotonic component is clearly non-deterministic. This is evidenced by the relative freedom of thought. The monotonic component, however, is believed to be deterministic, and for each logical form it outputs a unique structure or model. In Konrad (2004) and Søgaard (to appear), this is the *locally minimal* model.

Generation can not be made functional in LKB-HPSG. This requires some method for underspecification of phonological strings. Some of this relates to word order, but also, for instance, to homographs. A separation of linear precedence from immediate dominance may allow some underspecification of word order, but novel techniques must be invented to underspecify the phonology of a lexeme. This may be an interesting techical challenge, but it is far from certain that it has any empirical value. It is very unclear, at least to us, if natural language generation is in fact functional and monotonic.

7 Computational ressources

In the first report of the Scandinavian Grammar Matrix (Søgaard and Haugereid, 2004), a broad coverage noun phrase grammar was presented. It covered various word order phenomena, agreement, double articulation, superlatives, secondary modification, coordination, genitival possession, quantifying nouns, pseudopartitives, partitives, comparatives, compounds, postnominal possession and postgenitival quantification. The report came with a test suite. The functionality ratio of the noun phrase fragment of the Scandinavian Grammar Matrix was relatively low. The test suite consisted of 138 sentences covering the phenomena just listed. Four sentences were assigned two readings, and one sentence was assigned three readings. The rest of the sentences which were grammatical, were assigned only one parse. The functionality ratio is not perfect, however. So we realize it is not an easy task to obtain a fully functional grammar, but we conjecture that this is a realistic goal and a sound objective for grammar engineers.

In addition, Petter Haugereid is currently finishing an implementation of a considerable fragment of Norwegian which employs a high degree of functionality. It also relies on the principles for underspecification of argument frames and syntactic categories presented here. A preliminary version of the grammar is available from Petter's website; see the link from the website of the Scandinavian Grammar Matrix. The topological approach sketched in this paper is extended in a Danish grammar that Anders Søgaard is now working on. Some preliminary work on a new formalism, designed to implement functionality in a more direct way, has also begun.

8 Conclusions

This paper discussed various aspects of functionality in grammar design. It was shown how a typed feature structure grammar can, however complicated this task is, implement a strong functionality hypothesis. It was shown that such a grammar is also learnable and, apparently, quite realistic. Some fragments with a functionality ratio close to one has already been written for Mainland Scandinavian as part of the Scandinavian Grammar Matrix project. These fragments were written in LKB-HPSG. It is uncertain if other formalisms implement a strong functionality hypothesis, but it was shown that if the adequate grammar is functional, natural language is not context-free.

References

Tor Åfarli. 2005. Do verbs have argument structures? Unpublished manuscript, Norwegian University of Science and Technology.

Robert C. Berwick. 1982. Computational complexity and lexical-functional grammar. *American Journal of Computational Linguistics*, 8(3–4):97–109.

Hagit Borer. 2005. *Structuring sense*, volume 1. Oxford University Press, Oxford.

Kaja Borthen and Petter Haugereid. to appear. A grammar component for referential properties of nominals. *Readings in Language and Computation.*

Simona Collina, Paola Marangolo, and Patrizia Tabossi. 2001. The role of argument structure in the production of nouns and verbs. *Neuropsychologia*, 39:1125–1137.

Ann Copestake. 2001. *Implementing typed feature structure grammars.* CSLI, Stanford, California.

Jochen Dörre. 1991. Feature logic with weak subsumption constraints. In *Proceedings of the*

29th annual meeting on Association for Computational Linguistics, Berkeley, California.

Daniel Hardt. to appear. Natural language inference as triggered submodel search. In *The 18th International Florida Artificial Intelligence Research Society Conference*, Clearwater Beach, Florida.

Petter Haugereid. 2004. Linking in constructions. In Stefan Müller, editor, *Proceedings of the 11th International Conference on Head-Driven Phrase Structure Grammar*, Stanford.

Mark Johnson. 1988. *Attribute-value logic and the theory of grammar.* CSLI, Stanford, California.

Karsten Konrad. 2004. *Model generation for natural language interpretation and analysis.* Springer, Berlin.

Anne Neville. 2003. Danish pre-nominals as specifier heads. In *Proceedings of Scandinavian Conference of Linguistics 19*, Tromsø, Norway.

Sanni Nimb. 2005. *Danske adverbier mellem leksikon og syntaks.* Ph.D. thesis, Center for Language Technology, Copenhagen, Denmark.

Stuart Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics & Philosophy*, 8:333–343.

Anders Søgaard and Petter Haugereid. 2004. The noun phrase in mainland scandinavian. In *The 2nd Meeting of the Scandinavian Network of Grammar Engineering and Machine Translation*, Gothenburg, Sweden.

Anders Søgaard and Petter Haugereid. 2005. A brief documentation of a computational hpsg grammar specifying (most) of the common subset of linguistic types for danish, norwegian and swedish. In Henrik Holmboe, editor, *Nordisk Sprogteknologi 2004*, pages 247–256. Museum Tusculanum, Copenhagen.

Anders Søgaard. to appear. Model generation in a dynamic environment. In Takashi Washio, Akito Sakurai, Satoshi Tojo, and Makoto Yokoo, editors, *New frontiers in artificial intelligence.* Springer, Berlin.

Thomas Sudkamp. 2005. *Languages and machines.* Pearson, Boston, Massachusetts, 3rd edition.

Marten Trautwein. 1994. The complexity of structure sharing in unification-based grammars. In *The 5th Computational Linguistics in the Netherlands*, Entschede, the Netherlands.