Topological Parsing

Gerald Penn
Department of Computer Science
University of Toronto
gpenn@cs.toronto.edu

Mohammad Haji-Abdolhosseini
Department of Linguistics
University of Toronto
mhaji@chass.utoronto.ca

Abstract

We present a new grammar formalism for parsing with freer word-order languages, motivated by recent linguistic research in German and the Slavic languages. Unlike CFGs, these grammars contain two primitive notions of constituency that are used to preserve the semantic or interpretational aspects of phrase structure, while at the same time providing a more efficient backbone for parsing based on word-order and contiguity constraints. A simple parsing algorithm is presented, and compilation of grammars into Constraint Handling Rules is also discussed.

1 Motivation

There is a growing awareness among computational linguists that, in order for the functionality of current real-world natural language applications to progress to the next level, access to thematic roles and grammatical function assignment, i.e., "who did what to whom," will be just as important as a probabilistic model's ability to predict the next word in a string. In striving to represent useful meaning relations, we, and the annotated corpora we use, have dutifully followed the common assumption in linguistics that the assignment of relations are artifacts of configurational ones — primitive relationships between nodes in phrase-structure trees licensed by a grammar.

In the case of parsing with English, there have been some remarkable successes in the last five years, most notably that of Collins (1999) and several successive improvements, who use knowledge about headedness and subcategorisation, traditional n-grams and some information about unbounded dependencies to dramatically improve on our ability to predict the most likely phrase-structure tree given a string of words — with the tacit assumption that this tree has something to do with interpretation. While there have also been more modest successes with purely dependency-based grammars in the realm of freer word-order (FWO) languages, these often map dependency trees to phrase structure trees, and even agreeing on what the best phrase-structure tree should be in these languages is not easy. Predicting the tree from data, moreover, seems utterly intractable, given the number of movement operations and empty projections that would be involved in the standard approach.

While dependency-based grammar seems like a very appealing alternative in that context, phrases are a fact of life. No FWO language is completely free, and while the subunits like NPs that seem semantically intuitive to us may not always be realized as contiguous substrings in the strings of a language, there are often other contiguous substrings defined on the basis of prosodic effects, discourse relationships and/or purely formal syntactic rules that are adhered to. Invariably, dependency-based approaches must use various *ad hoc* devices under names such as "emancipation" to make exceptions where these notions

of contiguity do not agree. The constraints from these levels of linguistic structure interact, and phrases — of some variety — are the basic units for defining this interaction. For computational purposes, these constraints are interesting because they can be used to restrict search and, in the context of statistical parsing, to militate against less likely interpretations.

2 Kinds of Constituency

There has, in fact, been a considerable undercurrent of linguistics research, beginning as early as Curry (1961), that challenges the Chomskyan assumption that one flavour of constituency exists on which constraints from all of these levels of linguistic structure can happily agree. Curry (1961) distinguished what he called *tectogrammatical* structure, on which semantic interpretation takes place, from a *phenogrammatical* structure, which deals with word order, morphology and (dis)contiguities. Much of this work, including Curry's, has not been very formal. The purpose of this paper is to present one possible formalisation of it, and in a manner particularly consistent with how Curry's work has developed within HPSG (Kathol, 2000).

The one exception to this informality, Lexical-Functional Grammar (Kaplan and Bresnan, 1982), is worth noting, since it is also widely used by computational linguists. LFG, to its credit, had the foresight to distinguish two different kinds of structure very early on. One of them, *functional* or *f-structure*, is represented using a feature structure that directly indicates thematic role and grammatical function assignment, among other things, without any appeal to a primitive "f-constituent." While a more conservative representation (a phrase structure tree) will be used here for tectogrammatical structure, it would be entirely consistent with the spirit of the present work to use feature structures or even dependency trees in the context of this level of phrase structure. In LFG, the other, *constituent* or *c-structure*, which corresponds roughly to phenogrammatical structure, uses a phrase structure tree labelled with very tectogrammatical-looking categories: nouns, PPs, on occasion NPs, etc. Where these are not realised as contiguous substrings, c-structure trees are generally just flatter and wider-branching, in order to match the daughters of these discontiguous constituents directly, contiguously, and in the acceptable orders.

What is missing here is a primitive in the formalism for talking about contiguous substrings that may not have a semantic, tectogrammatical significance, and a primitive for talking about non-tectogrammatical regions over which word order constraints are expressed. Examples of the former are quite evident in the Slavic languages, such as with second-position clitics. As their name suggests, these clitics occur after something in first position. That something can be a normal tectogrammatical constituent, like an NP, or it can be a prosodic word, such as a preposition and first adjective of an NP (Browne, 1974), or, in certain circumstances, it can be a sequence of discourse-linked NPs (Penn, 1997). Of the latter, probably the best-known example is the German *Mittelfeld*. Within this field, pronouns generally precede prosodically heavier NPs (and with a particular order prescribed among multiple pronouns), and temporal adjuncts generally precede locatival adjuncts. It is false to claim that these ordering constraints holds only within a VP or over an entire clause. The *Mittelfeld* is, in fact, defined in linear terms, as the substring bounded on the left by either a complementizer or a finite verb, and on the right by a periphrastic verbal complex.

Linear fields like the *Mittelfeld*, usually called *topological fields* in the HPSG literature, are defined relative to some region, in this case German clauses, in which other fields may also be defined. These fields are linearly ordered with respect to one another, and sometimes have constraints on how many words or tectogrammatical constituents they can contain. Regions may also occur inside fields of larger regions, such as with embedded clauses in German. What emerges from this characterisation is an extended context-free formalism in which right-hand-sides of rules can use the Kleene star (as in LFG c-structures). What is different about these extended CFGs is that they do not provide interpretations — only a parse into linearly defined fields and regions. The present formalism consists of two parallel representational devices, one being this extended CFG and the

other, an interpretive tectogrammatical tree structure with potentially crossing links. Along with these come constraints that associate substructures from the two representations, in a very similar spirit to LFG structural correspondence functions.

The idea of using topological fields as a guide for general parsing appears to have originated with Oliva (1992); more recent work primarily folds in parochial facts from German, including Duchier (2001), which presents German topological parsing as a constraint satisfaction problem. The present approach actually received its inspiration initially from Slavic language word-order data, but can be applied equally well to German. Synchronous tree-adjoining grammars (Shieber and Schabes, 1990) bear some resemblance to the parallel derivations used here, although the same constituents are used there in both.

3 Formalism

We can state three characterizing assumptions that restrict the expressive power of this formalism:

- **Topological Linearity:** all word-order constraints can be witnessed by a topology defined on some *linear region.*
- **Topological Locality:** discontiguities may exist due to scrambling, but they are not unbounded.[1] Hence all discontiguities can be characterized in some *local region* of bounded topological size.
- **Qualified Isomorphy:** While linear and local regions are not always the same as traditional (tectogrammatical) constituents, they themselves are the same. Furthermore, principles governing linear order and discontiguity are stated relative to the *smallest common region* that witnesses the substrings being ordered or dislocated.

Topological Linearity agrees with the assumption made in traditional ID/LP grammars that linear precedence constraints apply within some region, although with ID/LP, that region is a tectogrammatically defined subtree. Linearity can be enforced by assigning substrings to different topological fields. Compared to relative statements of linear order, e.g., NP $<$ VP, topological fields allow one to make more absolute statements about linear position that are crucial for thinking about FWO syntax in a more modular fashion. Qualified Isomorphy refines an assumption made in earlier work on topological fields that every word simply bears a unique topological field. Topological structure is nested because the tectogrammatical structures it constrains are. Using phenogrammatical trees of nested regions allows us to order the words of an embedded clause, for example, without contradicting their placement relative to the words of a matrix clause because which field a word bears is relative to the region being considered.

We begin with the basic primitives from which grammars are constructed:

Definition 1 *A* topological signature *is a quintuple,* $\langle \mathcal{L}, \text{Field}, \text{Region}, \Sigma, \text{Phon} \rangle$*, such that:*

- $\mathcal{L}$ *is a* constraint language *for describing tectogrammatical categories, with a countable set of variables,*
- Field *is a set of* topological fields, *such as the German* Mittelfeld,
- Region *is a set of* regions, *such as clauses, relative to which topological fields are defined,*
- Σ *is a* lexicon, *and*
- $\text{Phon} : U^I \rightarrow \Sigma^*$ *is a function that maps elements in an interpretation,* I*, of* $\mathcal{L}$ *to phonological strings.*

$\mathcal{L}$ could be as simple as variables and constants representing atomic categories like `NP`, or a description logic for feature structures, for example. It can be a language with disjunction, although there is obviously a computational cost to be paid for this.

Definition 2 *A set of topological fields,* Field*, induces a unique set of* field descriptors, Desc(Field)*, such that for every* $f \in$ Field*:*

[1] While we do not deny the existence of unbounded dependencies, we believe they deserve a much different treatment. Our current approach has been to handle them within the tectogrammatical categories themselves, such as in the SLASH feature of an HPSG feature structure. These will not be discussed further here.

- $f \in$ Desc(Field) *(unique field),*
- $\{f\} \in$ Desc(Field) *(optional field),*
- $f* \in$ Desc(Field) *(0 or more fields),*
- $f+ \in$ Desc(Field) *(1 or more fields).*

We can now define our phenogrammatical structures. These are the extended CFG rules that divide regions topologically into fields:

Definition 3 *Given a topological signature, a* phenogrammatical rule *is of the form* $r \stackrel{P}{\Longrightarrow} d_1 \ldots d_n$, *where* $r \in$ Region, *the* $d_i \in$ Desc(Field), *and* $n > 0$.

When we look at the parse tree that corresponds to a derivation with a set of pheno-rules over some string, we see that every field and region can account for some contiguous substring that its subtree dominates. This is called the *yield* of that field or region. We can extend this notion of yield to tectogrammatical categories, although the substrings that correspond to these may not be contiguous. We could, following Johnson (1985), think of yields as bit vectors defined over a fixed length corresponding to the length of the input, for example.

Structural constraints constrain pheno-yields in terms of tecto-yields and vice versa. We look at them in terms of whether one *covers* another, i.e., substring inclusion.

Definition 4 *Given a topological signature, the* structural constraints, $\mathcal{C}$, *over that signature are, for every* $\phi \in \mathcal{L}$, $f \in$ Field, *and* $r \in$ Region*:*

- **covering:** ϕ covers f, ϕ covers r, f covered_by ϕ, r covered_by ϕ,
- **matching:** ϕ matches f, ϕ matches r, f matched_by ϕ, r matched_by ϕ,
- **linkage:** $r \twoheadleftarrow f$, $f \twoheadrightarrow \bigvee_i r_i$,
- **compaction:** $\langle\phi\rangle$.

Structural constraints are interpreted with universal quantification on their left-hand sides and existential quantification on their right hand sides, so ϕ REL f is not equivalent to its dual f REL_*by* ϕ. Covering constraints specify that the phonological yield of every/some tectogrammatical category denoted by ϕ consumes, or includes, the phonological yield of some/every field or region f/r, although the yield of ϕ may also extend into other phenogrammatical constituents. A special case of covering is matching, e.g., ϕ *matches* f. This is when the phonological yield of every/some tectogrammatical category denoted by ϕ is exactly the same as the phonological yield of some/every field or region f/r. As shorthand, we also allow $\phi \longleftrightarrow f/r$ for ϕ *covers* f & f/r *covered_by* ϕ. Similarly, matching constraints with universal quantification on both sides are written as $\phi \Longleftrightarrow f/r$.

Linkage constraints are essentially the converse of phenogrammatical structural rules: they license the links in a phenogrammatical tree with field mothers and region daughters. Linkage rules are always unary-branching. A field contains at most one region, with the alternative being one lexical item, i.e., a pre-terminal field.

Compaction constraints indicate that a tectogrammatical constituent has a phonological yield with no discontiguities. To these, we can also add universally quantified **implication** constraints, $\phi \rightarrow \psi$, which are the usual ones from constraint-based grammar — any tectogrammatical constituent in the denotation of ϕ is also in the denotation of ψ.

We are now in a position to introduce the tectogrammatical rules, which tell us how to build tectogrammatical structures. These are subject to the universally quantified constraints above, but can also specify constraints on a particular daughter:

Definition 5 *Given a topological signature and* $n \in \mathbb{N}$, *the* indexed structural constraints, $\mathcal{C}_n$, *over that signature are, for every* $1 \leq i,j \leq n$, $0 \leq k \leq n$, $f \in$ Field, *and* $r \in$ Region*:*

- **covering:** i covers f, i covers r,
- **matching:** i matches f/r, f/r matched_by i,
- **precedence:** $i < j$,
- **immediate precedence:** $i << j$,
- **compaction:** $\langle k\rangle$.

Definition 6 *Given a topological signature, a*

tectogrammatical rule *is of the form* $\phi_0 \stackrel{T}{\Longrightarrow} \phi_1 \ldots \phi_n; \rho$, *where the* $\phi_i \in \mathcal{L}$, $n \geq 1$, *and* $\rho \in \mathcal{C}_n$.

The indices in indexed constraints refer to the mother or daughter constituents in a tectogrammatical rule. In the absence of any indexed constraints, a tecto-rule makes no assumptions about the linear relationships among its daughters. Relative precedence and immediate precedence can be used to describe traditional phrase structure, where it exists, which can also be provided as an idiom: $\phi_0 \Longrightarrow \phi_1 \ldots \phi_n$. Note that, as with traditional ID/LP, compaction can be specified in the absence of precedence, which serves to specify contiguity separately from linear order; unlike ID/LP, precedence can be specified in the absence of contiguity (Goetz and Penn, 1997; Suhre, 1999). Manandhar (1995) has a similar approach to linear precedence.

4 Parsing

Just as with CFGs, there are a number of different control strategies that could be imagined for parsing with this topological formalism. The one presented here incorporates elements that are reminiscent of naive bottom-up, top-down and left-corner parsing. Information about headedness or statistically estimated parameters would be incorporated into a more sophisticated large-scale parser. For simplicity, the exposition here assumes that for every field or region, f/r, there is at most one structural constraint of each variety that universally quantifies over f/r.

The flow of the parsing algorithm is shown schematically running on a German example in Figure 1. Parsing begins after consulting a lexicon to find the tecto-categories associated with each word of input. These categories are then mapped by structural constraints to topological fields or regions (leftward arrows). From there, pheno-structure is built bottom-up using pheno-rules, much as in a bottom-up CFG parser. In German, it is often assumed that clauses have the following topology defined on them:

$$\texttt{clause} \stackrel{\texttt{P}}{\Longrightarrow} \texttt{vf}, \texttt{cf}, \texttt{mf}*, \{\texttt{vc}\}, \{\texttt{nf}\}.$$

where `mf` marks the *Mittelfeld* mentioned above. It is listed as `mf*` because the *Mittelfeld* can contain a sequence of regions. At fields or regions f/r where there are structural constraints universally quantified on f/r, we then predict some tecto-category (rightward arrows). In the figure above, for example, it is assumed that there is a constraint in German that:

$$\texttt{clause}\ \textit{matched\_by}\ (\texttt{s} \vee \texttt{rp} \vee \texttt{cp}).$$

which encodes our knowledge about the contiguity and position of these three categories' yields. Parsing proceeds in tectogrammar top-down in a manner restricted so that only what is *topologically accessible* to f/r can be matched, as explained below. During top-down parsing, derivations are checked against structural constraints universally quantified on descriptions ϕ that are consistent with the current category. Further bottom-up pheno-parsing can in principle be interleaved with top-down tecto-prediction in any manner.

4.1 Edges

Specifically, in a chart-parser implementation, we require four kinds of edges:

- **pheno-edges**: by parsing right-to-left and interpreting pheno-rules left-to-right, we need only passive (inactive) edges for bottom-up pheno-parsing. These record the **field/region** recognised and the **interval spanned** by the edge.
- **active tecto-edges**: these are the edges predicted during pheno-parsing. They record the **category predicted**, the field/region that predicted them, called the **sponsor**, and two bit vectors: one denoting the substring that can be used (**can-BV**), and one denoting the substring that can optionally be used (**opt-BV**). Their difference is what must be consumed by the category being sought. They also carry a set of **keys** for topological accessibility (explained below).
- **passive tecto-edges**: They record the **category found**, the **sponsor** that predicted them, a bit vector denoting the substring used (**used-BV**), and a set of **keys** that they confer.
- **frozen tecto-edges**: These are essentially active tecto-edges that are waiting for their can-BV. They record the **category predicted**, their **sponsor**, and a bit vector that must be consumed (**req-BV**).

Every edge also has a unique **ID**.

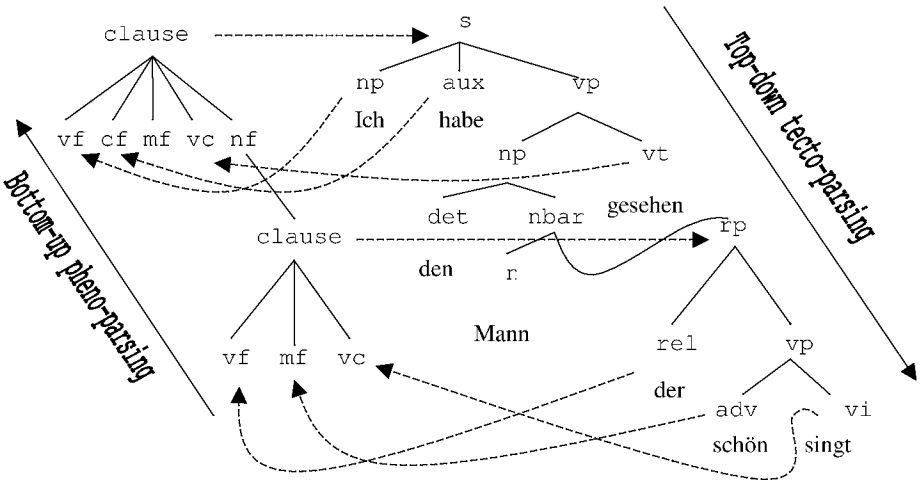

Figure 1: Flow of control in the topological parsing algorithm.

4.2 Rule Operation

There are four main combinations we must then implement once the input has been scanned:

• **pheno-completion**: given pheno rule $r \stackrel{P}{\Longrightarrow} d_1 \ldots d_n$ and pheno-edges for $d_1 \ldots d_n$, add a pheno-edge for r with the union of their intervals (likewise for linking).

• **tecto-prediction**: given an active tecto-edge with category consistent with ϕ_0, and tecto-rule $\phi_0 \stackrel{T}{\Longrightarrow} \phi_1 \ldots \phi_n$, predict ϕ_1 with the same sponsor, can-BV, and keys, but with an opt-BV equal to its can-BV — everything is optional because another daughter may consume the rest.

• **tecto-completion**: given an active tecto-edge with category consistent with ϕ_0, tecto-rule $\phi_0 \stackrel{T}{\Longrightarrow} \phi_1 \ldots \phi_n; \rho$, passive tecto-edges consistent with $\phi_1 \ldots \phi_j$. Then:

- **non-final**: if $j < n - 1$, predict an active tecto-edge for ϕ_{j+1}, with can-BV and opt-BV equal to the can-BV of ϕ_0 less the used-BVs of the passive edges, the keys of the passive edge for ϕ_j, and the same sponsor.
- **penultimate**: If $j = n - 1$, then predict the same for ϕ_n, but set its opt-BV to the opt-BV of ϕ_0 less the used-BVs of the passive edges — this is the last daughter and must consume the remainder of what is required.
- **ultimate**: If $j = n$, then check that what the union of the used-BVs does not cover in the can-BV of ϕ_0 is in opt-BV, and create a passive tecto-edge for ϕ_0, with the unions of the keys and used-BVs of the passive daughters. If the active edge was an initial prediction from pheno-structure, add the sponsor to the set of keys too. This can be interpreted as an exchange in which some higher active edge will be given access to this sponsor's yield in exchange for using this passive edge.

If a passive edge is lexical (produced by the input scan), we must ensure that its bit is topologically accessible to the sponsor of the active edge. If a tecto-rule has indexed constraints, then these constraints must be checked in addition (with bit-vector arithmetic, mainly).

• **tecto-unfreezing**: given an active tecto-edge and a frozen tecto-edge with consistent categories and accessible sponsors, if the req-BV of the frozen edge is contained in the can-BV of the active edge, then create a new active tecto-edge, with the same sponsor and can-BV, with an opt-BV less the req-BV, and a set of keys augmented by the sponsor of the frozen edge. This can be interpreted as an exchange in which the active edge promises to consume req-BV, and in turn receives a key to access some topological field/region.

4.3 Structural Constraint Operation

The first three of these are a variation on context-free parsing, in which bit-vectors are maintained instead of intervals. Active tecto-edges are initially predicted from pheno-structure by f/r *matched_by* ϕ constraints. Once we know that the yield of some f/r in a particular interval is matched by a ϕ, we can predict ϕ with the can-

BV of that interval without necessarily finishing pheno-parsing. Once we have built the ϕ, we will refuse admission to this f/r to higher active edges unless they agree to use ϕ. In this way, we can ensure that every f/r contains a ϕ in the final tecto-structure for the input.

Frozen edges are added to the chart by f/r *covered_by* ϕ constraints. We know that f/r should be consumed by a ϕ in tectogrammar, but ϕ may be larger. Frozen edges refuse admission to f/r to every active edge except those trying to build a ϕ.

Constraints of the form ϕ *matches* f/r restrict the can-BVs of active edges for ϕ to the maximal topologically accessible f/rs they cover, and enforce the requirement that the passive edges of ϕ match some f/r interval. Constraints of the form ϕ *covers* f/r enforce their interpretation on passive ϕ edges, and eliminate active edges in which can-BV covers no f/r.

Clearly, the idioms introduced above can be compiled specially to exploit the combination of constraints they provide. Input is accepted as grammatical if it is possible to build both a spanning pheno-edge of a distinguished region and a spanning tecto-edge of a distinguished category.

4.4 Topological Accessibility

Not all subconstituents in tectogrammar are compatible with each other just because their categories combine in a tecto-rule. The reason for this is that multiple pheno-structures are being built simultaneously in the chart. These pheno-structures can have different fields and thus, unlike CFGs, different structural constraints. As a result, we need some means of ensuring that passive edges from one pheno-tree are not being used by active edges predicted by another pheno-tree with incompatible structural constraints.

In order for a lexical passive edge to be incorporated into a tecto-structure, there must be an *accessible* path in the corresponding pheno-structure from the sponsor of the active edge at the root of the tecto-structure down to the lexical item. Every daughter field/region is accessible to its mother in a pheno-structure, but transitive closure of this relation is blocked by fields/regions that appear on the left-hand-sides of structural constraints, i.e., the fields/regions that predict tecto-edges. The sponsor of an active edge only has access through a blocking field/region if it possesses the key issued by that field/region. This key is given to the predicting edge only if it agrees to use up all the daughters under the blocking node.

Topological inaccessibility makes parsing of scrambling-related constructions more efficient. In a typical grammar, active tecto-edges are either prevented outright from using large portions of inaccessible input, or required to use an existing passive edge as the only means of access. Figure 2 shows a well-formed pheno-tree for German

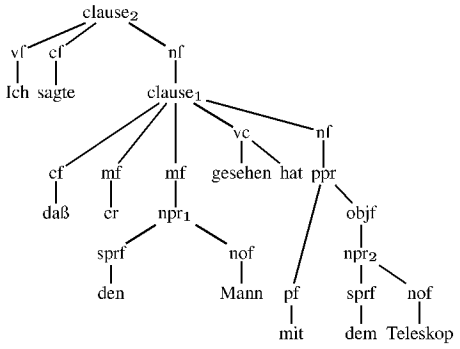

Figure 2: A well-formed pheno-tree for German.

with not only clauses but NP regions and PP regions that define those categories' internal structures. Figure 3 shows its corresponding tecto-

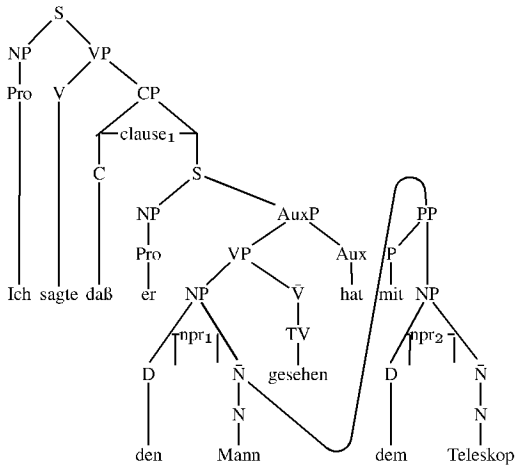

Figure 3: The well-formed tecto-tree for Figure 2.

tree. When the embedded VP dominating *gesehen* seeks an NP daughter, it must simply match the

NP edge for npr_1. The other embedded NP is inside a blocking ppr, and the subject NP is not in the yield of the $clause_1$ that sponsored this VP. Notice that the internals of $clause_1$ are inaccessible to the VP dominating *sagte* apart from the CP it offers because of a *matched_by* constraint. The result is that clauses are parsed largely independently.

5 Future Work

We are currently implementing a compiler based on this formalism in SICStus Prolog. The input to the compiler is a topological grammar and the output is a Prolog parser for that grammar. While there are no corpora sufficiently annotated for this model, the topologically annotated Verbmobil II corpus of German comes the closest. Based on a grammar with 74 phenogrammatical rules and 72 tectogrammatical rules extracted from 87 reannotated sentences of this corpus, our parser takes an average of 8.26 seconds per sentence to parse a larger set of 125 sentences from the same corpus on a Celeron 600 MHz computer running Windows XP. Much of the VM-II corpus consists of relatively simple utterances, and there were no recursive tectogrammatical rules, a significant obstacle for any purely top-down parser. The next step in implementation is to integrate bottom-up or mixed control into tectogrammatical parsing to more closely constrain the number of active edges predicted. A great deal more static analysis of parsing rule interaction and morphological analysis must also be performed for tractable parsing.

The space of parsing algorithms that this formalism supports needs to be mapped out to match syntactic properties of grammars with optimal algorithms for them. A significant amount of experimentation also needs to be done on providing the right higher-level constructs to grammar-writers that will reduce the complexity that comes with using this more flexible formalism. This may also lead to simplifications that could eventually be parametrised and statistically estimated to produce efficient large-scale language models for FWO languages that can capture more semantic information.

References

W. Browne. 1974. On the problem of enclitic placement in Serbo-Croatian. In R. D. Brecht and C. V. Chvany, editors, *Slavic Transformational Syntax*, volume 10 of *Michigan Slavic Materials*. Dept. of Slavic Languages and Literatures, University of Michigan.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

H. Curry. 1961. Some logical aspects of grammatical structure. In Jacobson, editor, *Structure of Language and its Mathematical Aspects: Proceedings of the Twelfth Symposium in Applied Mathematics*, pages 56–68. American Mathematical Society.

D. Duchier. 2001. Lexicalized syntax and topology for non-projective dependency grammar. In G.-J. Kruijff, editor, *Proceedings of the Joint Formal Grammar Conference / Seventh Meeting on the Mathematics of Language*.

T. Goetz and G. Penn. 1997. A proposed linear specification language. Arbeitspapier des SFB 340 134, Ebarhard-Karls-Unviersität Tübingen.

M. Johnson. 1985. Parsing discontinuous constituents. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*.

R. Kaplan and J. Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press.

A. Kathol. 2000. *Linear Syntax*. Oxford Univ. Press.

S. Manandhar. 1995. Deterministic consistency checking of LP constraints. In *Proceedings of the 7th Conference of the EACL*, pages 165–172.

K. Oliva. 1992. The proper treatment of word order in HPSG. In *Proceedings of the 14th International Conference on Computational Linguistics*, volume 1, pages 184–190.

G. Penn. 1997. On the plausibility of purely structural multiple wh-fronting. In *Proceedings of the Second European Conference on Formal Description of Slavic Languages*.

S. Shieber and Y. Schabes. 1990. Synchronous tree adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, volume 3, pages 1–6.

O. Suhre. 1999. Computational aspects of a grammar formalism for languages with freer word order. Master's thesis, Eberhard-Karls-Universität Tübingen.