Univerzita Karlova v Praze
Filozofická fakulta
Ústav teoretické a komputační lingvistiky

Alexandr Rosen

A constraint-based approach to dependency syntax applied to some issues of Czech word order

filologie – matematická lingvistika

vedoucí práce – prof. PhDr. Petr Sgall, DrSc.

2001

Prohlašuji, že jsem disertační práci vykonal samostatně s použitím uvedených pramenů a literatury.

Contents

Acknowledgements

It is a great pleasure to express my gratitute to many people for their willingness to share their deep knowledge and experience while this thesis has been in the process of birth.

First of all, I would like to thank my teacher and supervisor, professor Petr Sgall. Without him, it is hard to imagine that formal linguistics would exist as a discipline in Prague at all, and I am just one of many who consider themselves lucky to benefit from his incredible linguistic talent and thorough knowledge. Without his guidance and relentless encouragement I would have never brought the work to its end. On every occasion, he was prepared to discuss and comment anything I suggested or put on paper. I hope his kind attention to my (often naive) ideas will continue even after I submit this thesis.

Years ago I was introduced to machine translation by Zdeněk Kirschner. It was he who opened my eyes to the world of computational linguistics and to a down-to-earth view of language as a collection of unavoidable hard facts. Ever since I cherish memories of those long days of exciting discussions and work on the system that was not yet translating a particular construction to the full satisfaction of his author. I would like to thank Zdeněk for all this.

I also feel very much obliged to two other people whose profound linguistic knowledge and friendly attitude helped me not only to understand linguistic theory in the Praguian sense, but also to feel comfortable as a part of the team: Eva Hajičová and Jarmila Panevová.

I was very lucky in spending most of my working hours in a most friendly and stimulating surroundings. Confronted with an occasional impulsive idea, my colleagues Milena Hnátková, Jarda Peregrin, Niki Petkevič, and Hana Skoumalová never failed to respond to the best of their knowledge. I wish to thank especially Niki Petkevič and Hanka Skoumalová for many most stimulating discussions and lots of valuable advice.

I would also like to thank Karel Oliva, for helping me realize that linguistic theory and programming code need not be far apart, for sharing so many of his original ideas, and for being so helpful on many occasions.

Furthermore, I have benefited from occasional discussions with a number of other people: Tania Avgustinova, Dale Gerdeman, Jiří Hana, Ania Kupść, Detmar Meurers, Gerald Penn, Martin Plátek, Adam Przepiórkowski, Frank

Richter, and Hans Uszkoreit. I am grateful to each of them and I regret I have not planned my time better in order to ask for comments on this work.

I also should not forget the teachers and organizers of the ESSLLI summers schools and Mathesius courses: they have made my linguistic horizons wider and clearer.

Last but not least I wish to thank my family: Anna, Eva, and especially my wife Jarka, for bearing with me in general and during the last hectic stages in particular. It is due to their understanding, love and support that I was able to finish this.

Chapter 1

Introduction

This thesis has two main objectives. The first is to integrate two different components into a coherent linguistic framework: *Functional Generative Description* (FGD), a dependency-based linguistic theory rooted in the Prague School tradition, and Relational Speciate Re-entrant Logic (RSRL), a formal language suitable for constructing constraint-based grammars, created as a descriptive tool for *Head-driven Phrase Structure Grammar* (HPSG). The second aim is to show that the combination may be used to describe surface word order as conditioned by several factors, originating at various levels of linguistic description.

In more concrete terms, the goal is to provide a declarative, constraint-based account of a number of Czech word order phenomena using FGD as the theoretical foundation. Both surface-level constraints and deep word order (a concept of FGD) should have their proper role in such an account. I hope to justify the integration of the two components by showing that there is more to gain than to lose in terms of the standard criteria applied to hand-crafted grammars.

1.1 Motivation

An adequate description of the interaction between the underlying syntactic structure and its surface realization is an important goal of linguistic research. As the starting point, the main premises of FGD have been adopted, namely the structuring of language description into levels, the distinction between the system of language and its semantic and pragmatic interpretations, and the relevance of communicative dynamism and topic-focus articulation for 'linguistic meaning' – a notion corresponding to the level of underlying syntax, where the structure of a sentence is presented in the shape of a dependency tree with annotated content words as nodes (Sgall, Hajičová, and Panevová, 1986).

According to the original version of the theory, a language is described in procedural terms by a grammar generating the underlying structure, and transducing components, including movement rules, providing the interface between formally distinct levels of description.

This procedural specification has its drawbacks, such as that it is biased towards one direction of processing. It has been replaced for the purpose of this thesis by a declarative formulation, allowing for parallelism in representing and describing different language levels, inspired by constraint-based theories such as HPSG (Pollard and Sag, 1994), where the actual formalism employs as its main descriptive device a system of types, ordered within an inheritance hierarchy and supplemented by attribute-value pairs with the possibility of value sharing. This system constrains language objects, modelled as typed feature structures.

The other challenge comes the empirical domain: word order in Czech is a difficult topic and one that still poses unanswered questions. According to Vilém Mathesius (Mathesius, 1975), factors of various kind are responsible for word order not only in Czech, but also in English (and probably in human languages generally). The differences between languages with the so-called free and fixed word order are due to the relative weight of these factors. This view is compatible with a constraint-based formalism, advocated as an appropriate formal language even for FGD, and is also rather close to the view of FGD, where discourse-related factors interact with surface-level regularities in determining word order and prosody. Thus, my claim is that by adopting a constraint-based formalism for FGD, word order (and prosodical) phenomena (at least) in Czech can be solved more easily than in previous approaches. Although this may be better achieved by a system built with different parameter settings, I intend to show that even with the present proposal there are obvious advantages to be appreciated.

1.2 Scope

Because the task involves marrying two different linguistic traditions, a substantial amount of introductory reading is provided, with the aim to build solid foundations for the subsequent grammar writing exercise. The latter task consists of specifying how the deep word order is reflected in the surface string and how it interacts with surface-level constraints. This task can be solved on a rather abstract level, given the wealth of FGD research results concerning the issues of functional sentence perspective and related topics. It is more difficult to find a firm empirical ground for specifying surface-level constraints.

In addition to surface ordering effects of the underlying representation, I have attempted to cover to a various degree of detail several other phenomena governed by surface ordering constraints: besides the regular continuous cases also discontinuous comparison constructions, long-distance dependencies, split prepositional phrases, and the position of function words. The difficult class of clitics has received a more detailed treatment.

1.3 Organization

The thesis consists of 8 chapters and an appendix. The following two chapters provide theoretical and formal background for the core of the thesis, which is treated in chapters 4–6. Chapter 7 evaluates the framework against a range of empirical facts and the final chapter summarizes achievements and failures. In the following, the content of the chapters will be presented in more detail.

In Chapter 2, the background assumptions about the choice of linguistic theory, formal framework and software tool are discussed. Also in this chapter, some previous dependency-based approaches to word order issues are mentioned.

Chapter 3 describes foundations of the system: its theoretical background, formalism and implementation. In this chapter, the main features of FGD as a linguistic theory are presented and its embedding into a constraint-based formalism considered. Next, the syntax and semantics of basics of the formalism are presented, together with examples. Implementation issues will be addressed elsewhere, unfortunately outside the scope of this thesis.

Chapter 4 is concerned with linguistic rather than formal issues: features of FGD relevant in the following parts are specified in detail, and a few notes concerning some linguistic concepts are made. One of them deals with word order: starting from the word order principles of Vilém Mathesius, a classification of word order phenomena is developed, which is employed later in the proposed formal account.

In Chapter 5, the key components of the formal description are presented. After specifying the roles of tectogrammatical representation and derivation structure, I present the formal objects for representing the tectogrammatical and the morphemic levels. Finally, I briefly explain the role and the organization of lexicon.

Chapter 6 is rather formal, but probably the most important one. It starts with the definition of general principles governing the setup of linguistic objects, the representations of tectogrammatical and morphemic levels and the relation between them. Next, a number of surface-level constraints specific to Czech is suggested.

In Chapter 7, Czech clausal clitics have been selected as a set of items with very complex ordering regularities, with the intention of testing how robust the proposed framework is. A number of clitics-related ordering phenomena are covered by more general statements, some of them receive specific treatment, but there are a few for which an adequate description has not been found within the proposed framework.

Finally in Chapter 8, experience from the whole enterprise is evaluated and some perspectives in this line of research are sketched.

In the Appendix, the signature and the formal descriptions are presented again for reference, together with definitions of all relations used and a list of tectogrammatical functions.

Chapter 2

Preliminary Considerations

2.1 Criteria for an optimal linguistic theory

In the 20th century, theoretical linguistics has in many respects come closer to other scientific disciplines. Perhaps the most revolutionary methodological change was on the formal side: today, the universe of language – the language itself and the events of producing and understanding language expressions – can be represented as a formal model, which is constrained by statements of the linguistic theory. In §2.2 more will be said about formal requirements which a theory is supposed to fulfil and which make the long-established and sometimes misused term *generative grammar* an important characteristic of the theory.

Consequences of a theory satisfying the formal criteria have to be compatible with observable facts from the 'real' universe, and with assumptions about what constitutes a plausible shape of the model and the theory itself. Since the publication of Noam Chomsky's seminal work on syntax (Chomsky, 1957), it has been commonly accepted that a linguistic theory should satisfy requirements corresponding to the three levels of adequacy: observational, descriptive, and explanatory.

It is rather straightforward to test *observational* (or empirical) *adequacy* by comparing language expressions with the theory's predictions. Although there is hardly a theory which can claim 100% coverage for a standardised contemporary dialect of any language, different theories can be compared by using commonly available test suites or corpora, at least for some languages.

Some of these resources of constructed or natural data even include 'correct' analyses of the data – annotated parse trees (see, e.g., Lehmann and Oepen (1996) or Hajič (1998)). However, *descriptive adequacy* of a theory can be evaluated against such data only if the description scheme corresponds to the model used by the theory, i.e., if individual analyses of the annotated corpus or test suite can be converted into objects of the model. Furthermore, the objects of the model must be adequately structured and informative in a way which is sufficiently relevant and explicit.[1]

What exactly the requirement of adequate structure for the model objects means, is difficult to precise. On the one hand, there are a handful of 'common-sense' criteria, most notably the economy of representation (Occam's razor) and intuitiveness (or compatibility with linguistic tradition). On the other hand, there are more 'theory-internal' or formal criteria: the shape of the objects of the model should allow for another type of economy, the economy of description, or, more generally, for a theory satisfying the requirement of explanatory adequacy (see below). It is the relative weight of these two criteria (economy of representation vs. economy of description) which seems to lie behind some differences between theoretical frameworks.

Given a consensus on the shape of a description, even descriptive adequacy can be measured and compared. It is much more difficult to measure and compare *explanatory adequacy*. Intuitively, a theory is explanatory adequate if it is able to explain observable phenomena in the same way as physics explains why lightning comes about. Accordingly, a linguistic theory should provide reasons for ways in which languages are used, in which they differ and develop – i.e., for observable linguistic phenomena, phenomena whose existence is theory-independent. However, the term explanatory adequacy is mostly used in another, 'theory-internal' sense. In this sense, a theoretical construct is supposed to substantiate the existence of another theoretical construct. To put it in a more appealing way, there has to be something more to a theory beyond mere observational and descriptive adequacy: the theory should be a compact and coherent system, non-redundant and logically consequent, in Chomsky's words, it should not "miss possible generalizations". One of the clues to explanatory adequacy in this sense may be the extent to which the theory is able to derive descriptions of individual phenomena (across languages) from general principles ('universal grammar'). And it is precisely this search for general principles 'explaining' those seemingly disparate ways in which meaning is expressed both within a single language and across more languages that characterises today's research in theoretical linguistics.

At this point, it is important to realise that in searching for general principles linguists should not lose the empirical foundations of their discipline. The process of generalising from observations must be 'data-driven' even if the results are vulnerable to the criticism of being ad-hoc because they do not follow from any deep principle. After all, any theory is based on axioms, which cannot follow from something 'deeper'. As Pollard (1996) puts it in his 'Methodological Principle of Empirical Adequacy':

[1] According to my understanding, the term *descriptive adequacy* actually covers two requirements: a representation is descriptively adequate if it corresponds to standards given by the theory *and* if it satisfies some theory-external criteria on the shape and content of a proper representation. Clearly, for a reasonable theory the former requirement is subsumed by the latter one.

a. There are no "deep principles", since any theory can be reaxiomatized. In any case, science can only tell how things are, not why. Therefore:

b. first write constraints that get the facts right, and worry later about which constraints are axioms and which are theorems.

Another heritage of Chomsky is the division of labour between a theory of linguistic competence and a theory of linguistic performance, which corresponds to the distinction between the knowledge of language and the various ways such knowledge is used by human beings. While competence theories are an established discipline, a reasonably comprehensive description of at least some processes of linguistic performance is a task for the future. Moreover, there is no proven close correspondence between any of the currently available competence theories on the one hand and the actual competence grammars in human minds on the other.

Nevertheless, some conclusions can be drawn if a competence theory is actually interpreted within a processing regime. Firstly, the amount of processing required to fulfil common linguistic tasks like parsing or generating a sentence should not exceed an assumed capacity of a human language user. If competence theory A scores significantly better in this respect than competence theory B, there is a reasonable chance that theory A is closer to the language user's 'mental grammar'. Secondly, if competence theory A can be coupled with a reasonable theory of performance in a more natural way than competence theory B, it can again be argued that theory B is closer to reality than theory B. Indeed, in the absence of a full-fledged performance theory, such comparison must be approximated in some way, e.g. by assuming facts which such a theory has to take into account. Some of these facts are presented below in the form of further requirements which an optimal competence theory should meet.

Assuming that the statements of the theory of competence are interpreted within a processing regime, some additional requirements which a theory of linguistic competence should meet (see Pollard and Sag (1994)) are as follows.

The first requirement is that the theory should be stated in *declarative* terms, i.e., avoid procedurality in its description of language phenomena. Procedural descriptions represent bias for a specific processing regime, preventing correct interpretation of the theory and blurring the competence/performance distinction.[2]

[2]A formally specified grammar consisting of rewrite rules, which is said to *generate* a representation of an utterance, is often called *generative grammar*, although there may be no procedurality involved. In this sense, the use of arrows in rewrite rules and of the terms *generative* and *generate* is misleading: a generative (context-free) grammar may represent a static, *declarative* description of possible utterances and their representations. See §2.2 for more details on *generativity* in the sense of formal rigour in linguistic theory. Elsewhere, the terms *generative* and *generation* will be used to indicate a specific direction of processing, inverse to parsing.

Next, since the knowledge resources needed to produce/understand an utterance (grammar, contextual knowledge, world knowledge) are not consulted in any fixed order, but rather as required by the phenomena present in the utterance, the theory should be able to offer available resources from all levels of description simultaneously, i.e., avoid any fixed ordering of levels and treat them as *parallel* representations/descriptions of a single object.

Finally, since in the real world speakers/hearers are able to assign meanings to incomplete utterances, the theory should be able to describe *partial expressions*, including their interpretation at all levels of description.

2.2 Generative grammar

If a linguistic theory is to have any empirical consequences, it should be able to decide whether an abstract object, such as a well-formed structure representing a string of items of a language, belongs to the set of possible objects modelling that language. This is what Chomsky originally meant by the notion of *generative grammar*. There is an often quoted piece of text in Chomsky (1957, page 5), where the utility of formal rigour in linguistic theory is explained.[3] Pullum (1989) elaborates the notion of generativity into the following three criteria:[4]

1. It must be possible to decide whether an object (representation of a syntactic structure) is an object of the modelling domain.

   Typically, objects of the modelling domain are graphs of some sort: trees with labelled nodes (series of phrase markers as in GB), trees with labelled nodes and edges (sequences of multidimensional projective trees and strings labelled by complex symbols as in FGD), typed feature structures (i.e., directed graphs with labelled nodes and edges as in HPSG) or a combination of the above (phrase structure trees with feature structure annotations on nodes as in LFG).

2. It must be possible to decide whether an object (usually a statement in a formal logic) counts as a constraint of the grammar.

   Some theories employ context-free grammar with various extensions, other use notational variants of first order logic (Arc-pair grammar and

[3]"Precisely constructed models for linguistic structure can play an important role, both negative and positive, in the process of discovery itself. By pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of inadequacy and, consequently, gain a deeper understanding of the linguistic data. More positively, a formalized theory may automatically provide solutions for many problems other than those for which it was explicitly designed. Obscure and intuition-bound notions can neither lead to absurd conclusions nor provide new and correct ones, and hence they fail to be useful in two important respects." Unfortunately, judging by his more recent work, Chomsky seems to have changed his views.

[4]The criteria are presented here in a paraphrased version, adapted from Pollard (1996).

GB in one of its formalizations), feature logic with some extensions (relations – HPSG), a sequence of push-down automata (FGD in its 'classical' formalization) or a combination of standard and specialised tools (LFG: context-free grammar, quantifier-free theory of equality, and 'glue language' – a linear logic). Although a formal language is usually preferable, plain natural language can serve the purpose as well, provided that the statements can be interpreted only as intended.

It is difficult to prove or falsify statements of a theory which does not meet this requirement, precisely in the spirit of the above quotation from Chomsky (1957). It is therefore a highly desirable step to proceed from the level of intuitive and loosely delimited concepts to a thoroughly defined framework.[5]

3. It must be possible to decide whether an object of the modelling domain satisfies constraints of the grammar.

   Unlike model objects in other sciences, objects in linguistic modelling domains are finite (although there can be infinitely many such objects, each has only a finite number of nodes, edges, labels, etc.). Also, a theory consists of a finite number of statements (constraints). Therefore, as long as the above two criteria are met and the theory is stated clearly and explicitly, we should be able to check whether an object in the model satisfies the theory.[6]

In fact, the first two requirements are met whenever a theory is properly formalised and a proper formalization is actually the best means to test and meet the three requirements of adequacy: When observations about language are to be structured as a general and coherent theory in order to satisfy the requirement of explanatory adequacy, the move towards a formal framework (or 'formalism') is only natural, since it allows for linguistic facts to be stated explicitly, concisely and at an appropriate level of generality.

Then it is merely a technical problem to verify the other two Chomsky's requirements of observatory and descriptive adequacy by drawing conclusions from the theoretical statements.

The formal framework – a metalanguage for representing and describing language expressions – makes the theory readily falsifiable in the presence of appropriate facts.

However, not every type of formalism is suitable for every type of theory. The requirements put on the theory actually severely restrict the choice of a formalism, which should have enough expressive power to embody constraints of the theory. According to some views, a formalism should have

[5] An example of a theory which does not satisfy this requirement is Chomsky's minimalist program, see Johnson and Lappin (1997).

[6] This does not imply that it is possible to decide whether an arbitrary *string* is a string of a specific language.

only as much expressive power as is required to describe possible languages, i.e., a formalism should constrain the set of possible grammars – the linguistic theory. The arguments in favour of this position are as follows: (i) by assuming a highly constrained formalism the problem of language acquisition is easier to solve, and (ii) by using a constrained formalism it may be possible to decide whether a string belongs to a language, or to characterize a language or human languages in general by the power of the formalism. While the former argument loses ground in the face of infinite choice of options for a language learner and the latter is put in doubt when we realise the fact that some linguistic issues are undecidable, it seems reasonable to assume that a theory should not be constrained by a formalism, but rather that constraints should be imposed by the theory.

2.3 Implementation and testing

For descriptions of systems of such complexity as human languages, it is very difficult to maintain consistency even within a formal framework without the benefit of testing all aspects of the description computationally. Ideally, the formal framework should be directly encodable into an electronic form and interpretable by a corresponding software tool. In practice, it is often necessary to translate from the 'linguistic' formalism into a 'computational' formalism – of course, the shorter is the distance between the two, the better.

Another issue is whether the software tool offers the functionality needed to interpret the description adequately and in all desired aspects. For example, available linguistic software often includes a parser, but not a generator. Then a possibly reversible description can be tested only in one direction. In order to verify a linguistic theory by testing a hand-crafted grammar, issues concerning computational efficiency of the framework, such as speed, are of secondary importance.

Unfortunately, the formal description of a fragment of Czech presented in this thesis has not yet been implemented. It is previewed as the next step.

2.4 A long-term perspective

The decisions which have been made along the way were guided by the following long-term objectives, which can roughly be classified as 'engineering' goals, i.e., those concerning the overall design from the formal and computational viewpoint, or as 'linguistic' goals, which are related to the theoretical requirements and phenomena present in individual languages. The actual implementation of a multilingual environment is considered as a step which should follow the implementation of a monolingual module.

'Engineering' goals:

- The resulting system should serve as a testbed for a theory of linguistic competence in a declarative multilingual setting, enabling verification of theoretical hypotheses.
- The system should verify the possibility to build a declarative description of a language and of its relation to another language.
- The system should support linguistically motivated modularity and allow for the employment of existing resources (e.g., lexical resources, morphological component).

'Linguistic' goals:

- The description should as much as possible satisfy the requirements of observational,[7] descriptive and explanatory adequacy.
- The theoretical and formal frameworks should allow for declarative description of linguistic data, in order to support multiple modes of interpretation of the description, especially in the analysis and generation.[8]
- The framework should also allow for expressing data from all linguistic levels in parallel in order to make them simultaneously accessible, irrespective of any predetermined order of the linguistic levels of descriptions.
- Incomplete expressions should receive proper (underspecified) interpretation at all levels of description.

2.5 Parameters of the design and their settings

In order to achieve the above long-term goals, the system should be built from suitable components. There are a number of choices to be made for these components across several dimensions, in other words, the basic parameters of the system can be set in many ways.

In this section, these parameters are listed together with samples of options available as their settings. The actual choices made are presented together with some arguments.

The parameters are: linguistic theory and formalism, implementation formalism and processing tool, and the way monolingual descriptions are related to each other, i.e., the contrastive (transfer or interlingual) component.

Not all of the basic parameters of the system are equally easy to modify and they are strongly inter-related, too, e.g. the choice of a constraint-based formalism restricts all the other choices. Consequently, a linguistic

[7] Of course, it is not realistic to assume that all language phenomena can be covered. Therefore, a specific fragment of the language is delimited, for which the requirement of observational adequacy could be met.

[8] In other words, the description should be reversible.

theory which does not subscribe to the view that all linguistic hypotheses can be expressed and verified in a constraint-based setting requires a different formalism and processing tools.

2.5.1 Linguistic theory

There are many aspects in which theories may differ, concerning mainly their views on what descriptive and explanatory adequacy means, but also in other methodological aspects. A given theory may be:

- dependency or constituency-based,
- specified generatively or by constraints,
- with different relative importance of lexicon or grammar,
- with different number and content of description levels,
- with description levels organised as parallel interpretations within a single formal structure or as formally separate strata,
- with different views on where the borderline is between language and the rest of the world, etc.

In order to meet the goals presented earlier, an eclectic approach seems to be justified. Taking the requirement of descriptive adequacy as the starting point, there are a number of arguments in favour of dependency-based underlying syntax as the level which is (i) sufficiently abstract and thus devoid of surface syntax phenomena such as the form and position of function words, and (ii) still firmly belonging to the language system. A representation of this sort is proposed by the theory descended from the Prague School tradition, namely FGD (*Functional Generative Description*, see Sgall, Hajičová, and Panevová (1986). In addition to the elementary relations of dependency between content words in a sentence, for every dependent the representation also specifies an underlying function (comparable to θ-roles). Furthermore, the representation expresses the information structure of the sentence: all nodes in the dependency tree are distinguished as being either *contextually bound* or *contextually non-bound*.[9] The nodes are ordered with respect to the hierarchy of communicative dynamism, i.e., deep word order, where the most dynamic items (usually carrying the new information) come last.

The level of underlying syntax is one of the levels in FGD. In FGD, the relation between this level and the string of letters or sounds is usually described in a procedural way by means of transducing components and movement rules operating successively from a higher to a lower level of description.

[9] Contextually bound nodes represent those elements of the relevant discourse which are supposed by the speaker/writer to be easily accessible by the hearer/reader. Contextually non-bound nodes represent those elements which are supposed by the speaker/writer to be new for the hearer/reader.

This approach does not support easy interpretation of the description during analysis of the surface string and makes descriptive parallelism and partial interpretation difficult.[10]

For these reasons, the relation between the level of underlying syntax and the surface string is described in this work in a declarative way. Syntactic units (words, syntagms or phrases, clauses, and sentences) are modelled as typed feature structures (see §3.2 below), whose form and content is defined by a system of constraints – the grammar and the lexicon. In this setup, every syntactic unit (word, syntagm, clause, or sentence) is modelled and described as a single object with several dimensions, corresponding to description levels.[11] One of these levels is the level of underlying syntax, another is the level corresponding to the surface string, with a (possibly optional) level of surface syntax in between.[12] The levels are related to each other by value sharing and by constraints on representations of the multidimensional objects. Furthermore, all of the levels are accessible in parallel for an arbitrary syntactic unit. In this way, the theoretical framework meets the goals presented earlier.

[10]Crystal (1997, page 83) notes that it was Leonard Bloomfield who recommended a particular order of levels (bottom-up) in the analysis: according to this scheme, it is not possible to 'see' the next higher level before all work is done at a lower level. The author continues (ibid.):

> "Similarly, when we study grammatical patterns, such as sentence structure, we need to be aware of both semantic factors (such as the relationships of meaning that bring the patterns together) and phonological factors (such as the features of intonation that help to identify sentence units in speech). In a sense, when we work with levels, we need to be able to move in all directions at once. The British linguist J. R. Firth (1890–1960) once likened the business to a lift that moves freely from one level to another, in either direction, without giving priority to any one level. The simile makes its point, but the two-dimensional analogy is still misleading. To capture the notion of levels, multidimensional geometries are required."

[11]Not all words are represented at all levels as equal citizens. Function words do not appear at the underlying level as independent nodes, but rather modify some features of a relevant content word.

[12]The number and content of levels is not a purely linguistic-theoretical issue. There are other considerations due to formal and computational aspects. Seen from this perspective, a framework without the level of surface syntax is possible when the requirement that subparts of a syntactic unit have to form an adjacent surface string is abandoned – when there is no condition to the effect that the description is equivalent to a context-free grammar. For some constraint-based proposals in this direction see Reape (1994), Kathol (1995) and their discussion in §5.3.3 below.

The relaxed relation between syntactic tree and surface string, unconstrained by a context-free grammar, concerns the issue of function words occurring in a position not adjacent to the corresponding content word. Although from the viewpoint of (some) linguistic traditions the treatment of analytic forms belongs to the domain of morphology, their parts exhibit behaviour shared by regular syntactic units, such as clitics. And since the *syntactic* behaviour of analytic forms often introduces discontinuity in the surface string, the context-free requirement has to be relaxed.

The integration of the underlying level into a declarative framework can be compared – disregarding conceptual differences – to the position of f-structure in LFG or to the position of *content*-type objects in HPSG. While in LFG the relation between surface string and f-structure is mediated by c-structure, a level based on context-free grammar, in HPSG there is a possibility to employ a general relation between PHONOLOGY string and syntactic structure.

This option will be pursued in the present approach, although for reasons of processing efficiency a context-free backbone may be more desirable.

2.5.2 Linguistic formalism

Linguistic formalism is a metalanguage for expressing statements of the theory and for representing the objects being described, the universe of linguistic objects. The formalism should have enough expressive power and should preferably be defined as a logical system.[13] A given linguistic formalism might employ descriptive tools such as inheritance hierarchies of types, feature structures, value sharing, lexical rules and relational constraints (in constraint-based systems), movement rules or transformations (in derivation-based systems), context-free rules, general well-formedness conditions, linearization rules, etc. The choice of these tools is to a large extent given by linguistic theory. In fact, most theories are formulated within a specific formalism.

In our case, once we have agreed upon a declaratively specified theory, allowing for representation and description of any syntactic unit at all levels in parallel, the choice is rather restricted. The option which suggests itself is that adopted in 'constraint-based' frameworks such as HPSG: objects of the model are fully instantiated descriptions of the theory, exhibiting the same structural and conceptual properties. The desirable characteristics are a consequence of this isomorphism: an object of the model, having the shape of a multidimensional representation, can be defined by a system of constraints upon such an object.

Within FGD, representation of a sentence at the level of underlying syntax is a tree graph whose nodes correspond to content words, labelled by complex categories and ordered from left to right.[14] There is also a canonical linear version of the underlying syntactic structure. In either case, translation into a different formalism supporting complex ordered hierarchical structures is possible.

[13] See some remarks in 2.2 above on the issue of whether linguistic theory should be restricted by expressive power of the formalism.

[14] There is an added complexity in the treatment of coordination, which is represented as the third dimension of the tree.

2.5.3 Implementation formalism and processing tool

If a description is to be interpreted by a processing tool, it must be encoded within an implementation formalism. The distance between a linguistic and an implementation formalism may be very small or very large.

There are a few ready-made options available, especially for constraint-based linguistic formalisms, which were influenced by developments in computer science, computational linguistics and linguistic theories. Software tools such as ALE (Penn, 1993), ConTroll (Götz and Meurers, 1997) and LKB (Copestake, 1999) (among a number of other options) are all possible candidates for the implementation of a constraint-based system and the choice allows to stipulate particular requirements, such as a specific approach to processing (by constraint solving or by parser/generator). A detailed survey of systems for implementing HPSG-like grammars is presented in (L. Bolc and K. Czuba and A. Kupść, 1996).

2.6 Linearization of dependency trees

2.6.1 Dependency in linguistic theories

Ingredients of dependency grammar have been present more or less explicitly in most syntactic frameworks, including those of the monostratal constraint-based type. A common manifestation of dependency within constituency-based tradition is the distinction of one of the daughters in a constituent as the syntactic head, as attested by the usage of names for phrase types indicating their head (*noun phrase, verb phrase*). Chomsky's X-bar theory generalized the status of syntactic head across phrasal projections and phrasal types, giving the phrase structure a kind of 'dependency backbone'.

Valency can be viewed as another important dependency-based feature, because it specifies which items can be headed by a specific head, rather than which items co-occur with the head in a phrase. Valency is expressed in various terms as a set of deep cases, theta roles, or – on a more surface-oriented level – as a list or set of complements of a head.

In some approaches, description of phenomena such as binding of pronominals is based on an ordering of complements (HPSG) rather than on phrase-structure trees (GB). According to more recent HPSG-based proposals (e.g. Bouma, Malouf, and Sag (1998)), a list of all immediate dependents (including adjuncts) is used to solve other phenomena.

In addition to the inspiration drawn from the tradition rooted in constituency, a number of ideas which are included in this work have been presented in various flavours elsewhere in the literature explicitly advocating dependency.

In the following, I will compare FGD with a few other dependency-based approaches. I will be mainly interested in the way they describe the relation between the surface string and its dependency representation.

2.6.2 Functional Generative Description

Plátek, Sgall, and Sgall (1984) propose a multistratal formalization of Functional Generative Description (FGD), where the individual strata correspond to linguistic levels. The grammar consists of four subgrammars – see Fig. 2.1) below. The first subgrammar 'generates' (in the declarative sense) underlying (tectogrammatic) representations and the other three grammars provide successive mappings between the levels with the level of tectogrammatics at one end and the level of phonemics (or, for practical purposes, graphemic) at the other end. As the authors say, this description can be interpreted bidirectionally in generation and recognition. In both cases, the subgrammars are applied sequentially.

	start symbol
	$\Uparrow$
SUBGRAMMAR	*generation*
	$\Downarrow$
LEVEL	**tectogrammatics**[15]
	$\Uparrow$
SUBGRAMMAR	*transduction 1*
	$\Downarrow$
LEVEL	**surface syntax**
	$\Uparrow$
SUBGRAMMAR	*transduction 2*
	$\Downarrow$
LEVEL	**morphemics**
	$\Uparrow$
SUBGRAMMAR	*transduction 3*
	$\Downarrow$
LEVEL	**phonemics/graphemics**
	$\Updownarrow$
LEVEL	**phonetics**[16]

Figure 2.1: Multistratal formalization of levels in FGD

The generation grammar is a context-free grammar generating a language called *basic dependency structure* – BDS.[17] BDS is in fact a (bracketed) linearized version of dependency structure. At the level of tectogrammatics, BDS reflects the underlying functions, contextual boundness and communicative dynamism. An important difference is made between *dependency tree* (sometimes identified with *syntactic tree*), which corresponds to the BDS, and *derivation tree*, which corresponds to phrase marker – a phrase-

[15] Also called *underlying structure* or *level of (linguistic) meaning.*

[16] The mapping between the level of phonemics and phonetics is not described in the quoted article.

[17] A more elaborate version of this language – *complex dependency structure* – is used for representing structures combining dependency with coordination.

structure tree (a record of CF rules applications), i.e., to what is called syntactic tree outside the dependency-based linguistic tradition.

Transduction subgrammars use translation schemes (see Aho and Ullman (1973, page 215 ff)) to map dependency subtrees in the BDS notation (at the levels of tectogrammatics and surface syntax) or strings of symbols (at the levels of morphemics and phonemics/graphemics) between neighbouring levels. In these transductions, various phenomena are accounted for: control of embedded predicates, grammatical ellipsis (e.g., in some cases of coordinated structures), 'transfer' of relevant categories from the governing noun to the relative pronoun,[18] passivization, nominalization, agreement, and movement.

Although the authors do not deal with the topic of mapping between the levels of surface syntax and morphemics, they mention movement rules, which operate between the two levels.[19] If movement rules are to be formalized by means of translation schemes, it is not obvious how this could be done without the introduction of an extensive conceptual and formal apparatus used in the theory of Government and Binding (see, e.g. Haegeman (1994, page 293-670)) and/or a more powerful formalism for implementing the movement rules.

More recently, **Sgall (1992)** argues for a simpler scheme without the level of surface syntax. Even before this move, the level of surface syntax was nearer to the underlying level than to the surface string in that function words were not represented as nodes in the tree and adverbials were classified in a way parallel to the typology of free modifiers at the underlying level. Arguments for the existence of this level were based on general criteria of grammatical ambiguity and synonymy, which are shown to lose their force as examples of strictly synonymous syntactic constructions are difficult to find.[20]

[18] Here the bias towards one direction (generation) is only in the formulation. The translation schemes are reversible.

[19] In previous writings (Panevová, 1979; Panevová, 1980, page 144-171) the mapping between the levels of surface syntax and morphemics is described mainly with respect to the differences between the two levels in the values of grammatical categories and the percolation of agreement information. The issues of surface-syntactically determined word order (such as the placement of clitics and function words) have not been in the centre of focus.

[20] This includes cases of nominalization and passivization, see Sgall (1992, page 278). Arguments against the disposal of the level of surface syntax, based on the mismatch between the surface and deep syntax, have been offered by Panevová (p.c.). They include such issues as control in nominalizations (*podezřívá komornou z krádeže* 'she suspects the chambermaid of theft'), counted mass nouns (*dvojí víno* 'two sorts of wine'), reciprocal arguments (*Honza s Marií se líbají*, lit.: 'John with Mary kiss themselves').

However, the discussed level of surface syntax has not been adopted as the theoretical basis for the tagging scheme employed in the Prague Dependency Treebank. As an intermediate level between the underlying level and the morphemic level, a new *analytic level* is used. The level differs from the surface syntax level in that function words are treated

One of the merits of the immediate mapping between the levels of underlying syntax and morphemics is supposed to be the ease of handling the issues of surface word order: linearized terminal nodes of syntactic trees at the level of surface syntax often have the form of discontinuous syntactic units (syntagms), which often corresponds to the fact that the corresponding trees are non-projective (have crossing branches). The issues of linearization are argued to be better solved by applying movement rules between the underlying (projective) trees and the strings at the level of morphemics.

While the level of surface syntax was not used for solving all cases of surface word order anyway (due to its indifference to the issue of whether, e.g., a category of tense is realized as a function word or an inflectional ending or both), the absence of the level of surface syntax does not involve any principal new complications in addition to those already present under the original scheme in the mapping between surface syntax and morphemics.

Assuming the framework without the level of surface syntax, **Hajičová (1998)** takes up the issue of movement rules in more detail and provides illustration of movement rules of three types: (i) rules which move a rightmost (most dynamic) dependent at the underlying level leftwards (as in *My sister was visited by a painter in PARIS last week*, where *PARIS*, as the 'information centre' of the sentence and its most dynamic part, has moved), (ii) rules which transform underlying categories (grammatemes) of number, tense, definiteness etc. into 'morphemic items' such as endings or function words, which are in fact expanded rather than moved, and (iii) rules which reorder underlying items in cases where surface order is constrained by rules of (surface) syntax (as in *a larger town than Boston*, whose underlying order corresponds to *a larger than Boston town*).

The movement rules operate on bracketed strings – linearized dependency trees, terminal yields of the generative component. Similarly as Chomsky's transformations, which transform d-structure trees into s-structure trees, the movement rules transform underlying dependency trees into their surface counterparts in the same bracketed format, except for the fact that as the last step all brackets are deleted and the linear sequence of items is made explicit. The result is a morphemic representation in the form of a string of (complex) symbols, a linearization of the transformed tree with function words expanded.

However, the disposal of an explicit level of surface syntax does not resolve the main weakness of stratificational approaches with transformations or movement rules as additional mechanisms for relating formally distinct levels of description, namely the bias towards a certain processing mode. In order to apply the framework to the task of recognition or parsing, movement

as nodes in dependency tree. Although this level is claimed to be motivated by technical needs, it may also indicate a gap in the way FGD treats the relation between the syntactic tree and the surface string, with or without the level of surface syntax.

rules have to be applied 'backwards'. Since movement rules in generation map a tree onto another tree, whose terminal yields are trivially linearized as a string, a naive reverse of the generation procedure during the process of parsing would involve making assumptions about the tree structure (bracketing) of the parsed string before any movement or grammar rules are applied. The assumptions about the actual shape of the tree and the movement rules applications could then be verified only later by rules of the generative grammar.

This inefficiency during the process of parsing is avoided by treating the transition between the level of morphemics and the syntactic tree in a different way. According to one of possible proposals, a tree is built before movement rules are applied.[21]

Petkevič (1993) and **Petkevič (in press)** follows the multistratal model of FGD, concentrating on the level of underlying syntax (tectogrammatical level), one of the formally distinct strata. As in Plátek, Sgall, and Sgall (1984), a context-free grammar (alternatively, a push-down store automaton) generates an underlying dependency structure, obeying the constraints given by principles of contextual boundness and communicative dynamism. Nodes of the dependency tree are represented in a constraint-based formalism as feature structures, while dependency relations are represented as in the previous work by brackets of *basic dependency structures* or *complex dependency structures* (which include coordination). Thus, the formalisms used for representing the structure and the nodes of the structure are different.[22]

The nodes of the underlying dependency tree are unified with entries in the underlying lexicon to yield a complete underlying representation. Also in this step, generated dependents are checked against valency frames. Mappings to the other strata are considered to be a separate issue, treated along the lines presented in the previous FGD work.

Further extensions in this direction, namely in the use of a constraint-based formalism within the FGD framework, seem to be a natural next step. Instead of using two different means for formalizing structural relationships on the one hand and nodes in the structure on the other, a constraint-based formalism could be used to model and describe both. The formalism itself could use types to constrain the universe of possible objects in the model world of underlying level, even to the extent of replacing the rules of generative grammar by a recursive definition of types corresponding to

[21] Approaching the issue from a more computational angle, it may be possible to compile a restricted class of movement rules and rules of the generative grammar into a single monostratal grammar.

[22] Some theories employ the same formalism for representing both structure and its elements. HPSG, especially in its more recent version, is a good example. LFG uses a single formalism for its f-structure. A single formalism is useful for expressing constraints and relations which involve various pieces of information from multiple elements.

syntagms or subtrees. In an even more radical move, the individual levels could be modelled and described in parallel together with the mappings between them within a formally uniform structure.

2.6.3 Other dependency-based frameworks

There a few other major linguistic theories based explicitly on dependency structures. Extensive work has been done within the multistratal *Meaning-Text Model* **(Mel'čuk and Žolkovskij, 1970)**, which is similar to FGD in using a series of formally independent levels (strata), each for every level of description.

Other important theory is the monostratal surface-oriented *Word Grammar* **(Hudson, 1990; Hudson, 1998)**. Its most recognizable superficial characteristic is the form in which dependency relations are represented: by arrows linking words written sequentially as in a plain text. Orthographic words are treated as elementary units. Syntactic structure may be more complex than a tree, as in cases of long-distance dependencies, where the non-locally dependent word forms a 'visitor' relation to the nearest suitable governor. Thus, in a sentence *What do you think?*, the *wh*-pronoun is dependent both on *think* (as its object) and on *do* (as its visitor). Word Grammar uses default inheritance to formalize general patterns and exceptions: by default, English words follow the word they depend on, but exceptionally subjects precede it. The theory also has a more computationally-oriented side: it comes with a parsing algorithm, the design of a method for generation is planned. A specific parsing regime has to be stipulated in addition to the grammar itself, which does not provide any syntactic rules in the usual (computational) sense.

Another monostratal dependency-based theory is *Lexicase* **(Starosta, 1988)**, which originated in opposition to the Chomskian system of multiple strata under the influence of Charles Fillmore's deep cases. Unlike Fillmore's deep cases, the Lexicase case roles are not based on situational but rather on syntactic criteria. Word order constraints are – similarly as any other constraints – formulated as lexical specifications of governors and are based on word class, syntactic function and the notions of Theme and Spotlight as the first (preverbal) and last (postverbal) constituents.

Similarly as Lexicase, the monostratal *Dependency Unification Grammar* **(Hellwig, 1986)** is very much a lexically oriented theory: all dependents are specified as valencies of their governors. Adjuncts are treated as dependents having a valency slot for their governor. The theory is very much computationally oriented and comes with a formal language which is directly interpretable by computational tools. The formal language employs non-recursive feature structures: attribute values can only be atomic. Dependency relations do not link a governing word and its dependent word, but rather a governing word (which is a terminal node) and its complement as a complete subtree. According to this view, dependency grammar does not imply the disposal of non-terminal nodes. Every item in the surface string corresponds to a node in the representation tree, which represents several aspects (levels) of an expression simultaneously: syntagmatic function, lexical meaning and morphosyntactic features. Word order is constrained by means of position attributes. They enable to handle, e.g., word order phenomena of German. Analytical verb forms are represented using the concept of *nucleus*: a nucleus inherits all categories and templates from its component function and content verbs.

The use of parallel description of multiple levels within a single formal stratum is a trait common among constraint-based approaches. As the author says (Hellwig, 1998, page 4-5):

> It turned out that f-structures and c-structures, to use the terminology of LFG, can be processed completely in parallel in a dependency tree. In more than 25 years of experience with DUG I never came across any principal divergence between the three levels of analysis.

However, this claim is put in doubt later (ibid., p.7), where the author lists some open problems: discontinuous analytic forms, object raising, scope, topic-comment structure and ellipsis. Thus, the claim about 1:1 relationship between words in the surface string and the nodes in the dependency tree seems to stand in the way to proper treatment of a number of phenomena.

It is obvious why constituency-based approaches are favourite tools for the description of languages with a high degree of grammaticalized word order and why formal linguists dealing with languages where word order is less rigidly tied to the rules of (surface) syntax are more inclined to seek alternatives. It is not difficult to find such an alternative in the older tradition of dependency grammar, which represents syntactic relationships in such languages in a more intuitive way. However, it is not obvious how the mapping between the linear string of morphemes and the dependency tree should be described within a dependency framework.[23] Some of the approaches briefly introduced above solve this issue by means of non-trivial linearization components implementing reordering relations between the linear string and the dependency tree (FGD, Meaning-Text Model).

[23]Dependency is a syntactic relation, but not all dependency grammarians agree that it determines word order. Lucien Tesnière (1959) assumed syntactic structure in abstraction from surface string and a number of other dependency grammarians adopted a structure which does not represent word order.

On the other hand, in theories such as FGD and Word Grammar, word order is an integral part of dependency structure. However, the two theories differ in that FGD assumes projective syntactic trees, representing ‘deep word order’, and movement rules mapping the trees onto surface word order, while Word Grammar admits more general structures with crossing branches and multiple parents, whose terminal nodes correspond to surface word order.

Other frameworks model and describe the mapping by means of context-free grammar rules, each paralleled with a representation of dependency relationship(s) involved. Such an approach comes close to that of Lexical Functional Grammar (Dalrymple, 1999; Dalrymple et al., 1995), where context-free rules generate c(onstituent)-structure, providing at the same time mapping to f(unctional)-structure, or Head-driven Phrase Structure Grammar, where context-free rules – at least in some versions – combine complex categories, whose *content* values carry semantic interpretation of the relevant constituent.

Context-free grammars have been used to build dependency trees (and to generate from such trees) in two machine translation system prototypes: an English-to-Czech system **(Kirschner, 1982)** (see also Kirschner and Rosen (1989)) and a Czech-to-Russian system **(Oliva, 1989)**. In both systems, a single computational formalism (Q Systems, see Colmerauer (1970)) was used to implement descriptions of morphology, syntax and transfer (in a restricted form). Analytical morphology was formally treated as a syntactic phenomenon at the same level as complementation and adjunction by means of context-free rules: a typical rule corresponded to the description of a (partial) complex form or a dependency relationship. In fact, rules of the latter type could be compared to the rules used in Plátek, Sgall, and Sgall (1984) used for generating BDS – the basic (or complex) dependency structure (see above). However, the mapping to the string of morphemes is done directly: the sequence of terminal nodes of the *derivation tree* represents the surface word order.

Rosen (1996) represents another attempt in this direction, using a constraint-based formalism and making the disassociation between the abstract dependency structure and grammar rules explicit. In addition to the issues involved in the treatment of function words and multiple complementation/adjunction of a single head, the mismatch was necessitated by the need to cover many syntactic phenomena of Czech by rules of two variants: head-final and head-initial. The resulting (surface) syntactic tree is then necessarily different for each member of a set of constructions which consists of different word-order variants of otherwise identical elements and relations modulo functional sentence perspective facts.

This issue is addressed by **Avgustinova and Oliva (1990)** and Oliva (1992), who propose a uniform shape of (surface) syntactic trees, formally identical to lists. The trees are binary-branching with head daughters always on the right. The trees are projected from an empty functional head sitting at the rightmost node of every head domain. As the functional head inherits all categorial and valency information from the 'real' lexical head, all rules can be head-final. As a result, pre- and post-head word-order differences in otherwise identical constructions do not require pairs of rules and word order can be constrained by independent means.

Kuboň and Plátek (1993) propose another formalism based on context-free grammar with an integrated mechanism for deriving syntactic dependencies from applied rules. The framework also includes an important extension for parsing discontinuous constructions.

Oliva and Petkevič (1998) deal explicitly with the issue of mismatch between a processing-independent and descriptionally adequate dependency tree and the need to compositionally describe its relation to the surface string. The idea of syntax conceived as a network of phenomena, which are described independently of each other and of the issues of processing, is married with a dependency framework in the spirit of FGD. The merit of this approach is in providing for a natural, linguistically-motivated declarative formalization of independently viewed phenomena. Classifying generalizations about language into an array of separate phenomena can be viewed as a separate issue.

However, another problem is pointed out by the authors: although the proposed framework is able to describe the relation between a string of words and its dependency representation, it does not lend itself easily to processing, namely to parsing a surface string into such representation. The reason is that it lacks an explicit description of how dependency relations relate to the surface substrings, i.e., (in parsing) how a dependency tree can be constructed compositionally. Traditionally, such description is provided by context-free grammar, for which there are efficient implementation techniques. So far, a parallel technique for dependency-based grammars is yet to be found.

Bröker (1998) – similarly as FGD – proposes to disassociate the level of dependency representation from the surface ordering. However, the relation between the two levels is not defined by means of transformations or movement rules, but rather in the spirit of constraint-based theories, namely Lexical Functional Grammar, by providing a dependency-like functional interpretation (equivalent to LFG's f-structure) of surface context-free rules (equivalent to LFG's c-structure). Categories used in the context-free grammar rules are redefined as 'order domains', whose terminal yields correspond to surface order. Order domains corresponding to individual words recursively concatenate into larger order domains, constituting 'order domain structure', which corresponds to a phrase structure tree.

As in LFG, nonlocal phenomena can be treated by functional uncertainty. Thus the German sentence *Den Mann hat der Junge gesehen* is represented as a projective dependency tree at the f-structure level (with *den Mann* being dependent on *gesehen*), while the initial NP is interpreted by functional annotation in a c-structure rule, which can be paralleled to the LFG rule in (1).

$$(1) \quad \mathrm{S} \Rightarrow \begin{array}{cc} \mathrm{NP} & \mathrm{VP} \\ (\uparrow \{\ (\textsc{vcomp}^*)\ \textsc{obj} \mid \textsc{subj}\ \}\ ) = \downarrow & \uparrow = \downarrow \end{array}$$

The system is implemented using the Xerox Linguistic Environment package (`http://www.parc.xerox.com/istl/groups/nltt/xle`). The author notes a potential problem in that the resulting amount of functional uncertainty may lead to unbearable inefficiency. Another problem is the unfulfilled desideratum for the possibility to express binary precedence predicates over dependency relations.

In the employment of an existing constraint-based formalism, Bröker's approach is similar to the work presented in this thesis. It is differs, though, in that it relies on a context-free backbone in its description of surface order.

As a natural extension of a large rule-based tagging project,[24] **Järvinen and Tapanainen (1998)** describe a parsing engine called Functional Dependency Grammar (FDG), which produces syntactic analyses inspired by work of dependency-oriented theoreticians and consistent with the practical goal to provide syntactic annotation for unrestricted text.

Similarly as in most other dependency-based frameworks, the issue of defining a descriptively adequate representation scheme assumes the central role, while the grammar is defined in a more ad-hoc fashion. In the case of FDG, representation (or rather annotation) is constructed by means of a large number of construction-specific rules with no ambitions of arriving at theoretically interesting generalizations. Perhaps due to this practical orientation, an implementation of this system is one of the hottest candidates for the most successful parser of English presently available.

Similarly as the work described by Järvinen and Tapanainen (1998), the framework of **Sleator and Temperley (1993)** represents a practical system, designed for parsing unrestricted texts, producing concise structural and functional annotations. The system is based on lexical entries with a range of valency-like lexical attributes, which allow words to be combined into relations, interpretable as dependencies.

FGD has also been formalized as a multimodal categorial grammar **(Kruijff, 1999)**, together with the specification of the deep/surface relation for a number of phenomena. This is another example showing that it is useful to view linguistic theory independently of the formalism in which its statements are expressed.

2.7 Conclusions

In this chapter, I was setting the stage by listing a number of requirements which linguistic theory and formal language should fulfil, both from the general viewpoint of the adequacy criteria, and from the more specific perspective of the present work. I also presented a brief overview of how various

[24] See Karlsson et al. (1995) and/or a review thereof by Rosen (1998).

dependency-based frameworks deal with the issue of relating dependency tree with its surface realization.

Not all requirements could be satisfied within the scope of this thesis. Implementation of the description of a fragment of Czech is previewed as the next step, together with the necessary rephrasing of some of the descriptions into a more computationally tractable form.

However, as a result of comparing the various dependency approaches, it is considered important to keep the theoretically adequate description separate from its computationally tractable version, if the two cannot be the same.

I have also presented arguments in favour of a specific combination of the theory of FGD and the constraint-based formalism as used in HPSG. The arguments were based mostly on the difficulties of a multistratal description employing movement rules for linearization of dependency tree nodes. I will assume the combination of FGD and a version of the constraint-based formal language throughout the rest of this thesis and I will provide more arguments as this approach will be applied to specific phenomena, related mainly to word order in Czech.

Chapter 3

Foundations

3.1 Theoretical background

Before sketching the shape of the linguistic theory behind the present work, I would like to make some general comments on the assumed status of linguistic theory and its domain.

3.1.1 Ontological issues

Linguistics is usually defined as the science of language. Similarly as in other sciences, its goal is to formulate (theoretical) statements of appropriate generality to account for existing data. In linguistics, the data is a human language, or – more ambitiously – human language in general.

Theoretical statements in linguistics (or at least in *theoretical* linguistics) have the form of a grammar (and a lexicon), which delimits (describes, generates, licenses, constrains) the set of possible language expressions. In addition to delimiting such a set, the grammar usually assigns one or more representations to the expressions in the set. Here, a few ontological questions may be raised: what is the status of such representations and what is their relation to the represented expressions and the grammar? More generally, in what kinds of objects (both real and theory-internal) is linguistics interested and in what way are they linked?

Within the constraint-based grammar tradition such issues were opened by Gazdar et al. (1985) and the discussion has continued in more recent work within the HPSG grammar theory (Pollard and Sag, 1994; Pollard, 1999). The main concepts are the same as in natural sciences: a certain domain or aspect of the theory-external world is idealized by means of a model with some desirable mathematical properties. The theory then describes properties of such a model.

In linguistics, the theory-external domain consists of a specific type of events, namely tokens of linguistic expressions. Those among them which are considered to be parts of (a specific) language are idealized as objects of a formal model of (a specific) language, i.e. as mathematical structures of a certain type. However, multiple tokens of linguistic expressions sharing identical linguistically relevant properties (e.g., more occurrences of an utterance) are modelled by structures which are undistinguishable – isomorphic. What exactly is the ontological status of such structures as abstractions over individual entities is a matter of debate reducible to the argument between the two philosophical schools of realism and nominalism. Without taking sides, a subset of isomorphic structures can be viewed as an isomorphism class, of which the structure is a representative.

Which kinds of structures are possible as objects of the model and therefore correspond to (a token of) a theory-external linguistic expression is determined by more or less general statements of a formal grammar (and lexicon). The well-defined formal properties of the model allow the theory to make concise and falsifiable statements at the cost of not talking directly about the theory-external world.[1]

When objects of the model are idealized parallels of theory-external linguistic expressions, is there something in the 'real world', for which formal grammar is the idealized parallel? Indeed, this 'something' is often called *mental grammar*, the linguistic competence of a language user, the static source of knowledge about forms and meanings of elementary linguistic units and the possible ways they can be combined.

This source of linguistic competence is consulted in the various activities involving language: comprehension, production, translation, language games. All these processing regimes can share the single source. What must be added and what seems to be to a large extent specific to the various activities, is a strategy of using the source – a mental *performance* grammar. Unlike competence grammars, formal parallels of mental performance grammars is a topic whose research is in its beginnings.

Once the competence vs. performance distinction is accepted, linguistic theory can be expected to meet certain requirements, otherwise it would not fit into the scheme of a single static competence grammar being consulted by a processing module. A few such requirements have been presented in §2.5.1: declarativity, parallelism and the possibility to describe partial expressions. These requirements can be satisfied when linguistic entities are modelled (represented) as structured objects, whose parts correspond to distinctions made along several dimensions: nodes in a syntactic tree, levels of description, or various features of individual words or morphemes. In this way, the same object can model several aspects or dimensions of its 'real' counterpart at the same time.

More will be said about the shape of the structured objects from the for-

[1] Generative linguistics has a term for the set of structures *generated* by a grammar, provided that there are no isomorphic structures in the set, namely *strong generative capacity*.

mal viewpoint in §3.2. Here, the important point is to realize that the multidimensional character of the objects can be reflected in the formal grammar, which can be based on a theory advocating several levels of descriptions.

3.1.2 Linguistic sign

One of the approaches to the 'ontology' of linguistic data which lends itself easily to being recast within the model of multidimensional objects, is Ferdinand de Saussure's (de Saussure, 1916) view of language as unifying in the notion of *linguistic sign* the two opposing aspects: 'the signified' or 'the form' (such as the acoustic image of the word *tree*), and 'the signifying' or 'the meaning' (such as the concept of a tree).

The two aspects of linguistic objects have been adopted by a number of theories, including stratificational ones. In FGD, there are several objects in the model corresponding to a single theory-external entity, each at a particular level (stratum) of description and each exhibiting the opposition between the signified and the signifying. These oppositions (signs) are 'partial' in the sense of representing the form and its function ('meaning') as interfaces of a specific level: the function of a unit (its 'meaning') at a lower level (nearer to the acoustic image) equals a form of the corresponding unit at the neighbouring higher level. In order to arrive at the opposition (sign) spanning all levels between the acoustic image and the related concept, the 'partial' opposition (signs) are traversed in succession.

Another theory which makes explicit reference to de Saussure's *sign* is HPSG. Indeed, its model world is populated with multidimensional objects called *signs*. However, an HPSG sign has more than two aspects. In addition to the aspects representing the intermediate levels of description, there is an important aspect corresponding to the combinatory properties of the object.[2]

Given the performance-based and computational arguments in favour of multidimensional objects and a single stratum, it seems worthwhile to make an attempt to recast a theory expressed in multiple strata within a monostratal framework. This may especially be the case if the notion of linguistic sign is an important ingredient in the theory and if objects in the model and the theory itself can be expressed without the employment of multiple strata and transformations relating those strata.

In the following, main concepts of the linguistic theory of FGD will be presented with a stress on its potential to underlie a monostratal constraint-based system.

[2]De Saussure's binary signs have been extended already by Roman Jakobson (Jakobson, 1990).

3.1.3 Functional Generative Description

The main points of the theory[3] can be summarized as follows:

3.1.3.1 Multiple levels of description

More than one level of description is necessary in order to express substantial linguistic facts about human language. While the acoustic image of an expression is represented and described at the lowest level of phonetics, its *meaning* is represented and described at the highest level of *tectogrammatics* (also called *underlying* or *deep syntax*).[4]

In earlier versions of the theory, the following 5 levels were assumed: phonetics, phonemics, morphemics, surface syntax, tectogrammatics. More recently, arguments have been put forward in favour of dropping the level of surface syntax (Sgall, 1992) (see §2.6).

Among the levels mentioned above it is the level of tectogrammatics which receives most attention. It is here where linguistic phenomena are represented in an abstract and yet explicit fashion: even at this abstract level, any linguistic information relevant to semantic interpretation and thus constituting a part of *linguistic meaning* is included, no matter whether it corresponds to a morphological, syntactic, or lexical category. At the same time, the level of tectogrammatics (also called the level of linguistic meaning) serves as interface between the system of language and the extralinguistic *content*.

Given that a constraint-based formalism can accommodate any number of levels, multiple levels as such are not an issue. If, however, multiple levels are represented in the theory as distinct strata with complex transformations or movements specifying relationships between them, a reformulation of the theory is necessary. In fact, this step can lead to a more compact and transparent theory.

3.1.3.2 Language proper as the domain

Only *linguistic* aspects of language are considered to be the domain of the theory. Distinction is made between *meaning* and *content* of an expression: the latter term concerns conscious images present in human mind, not necessarily fitting into patterns of a specific language, while the term *meaning*

[3] The theory was presented as a whole in Sgall, Hajičová, and Panevová (1986), more recent work was published e.g. in Sgall (1992), different versions of its formal specification in Plátek, Sgall, and Sgall (1984), Petkevič (1987) and Petkevič (in press).

The theory has also been used in the design of the system for syntactic tagging in the Prague Dependency Tree Bank (Hajič, 1998), (Alena Böhmová and Eva Hajičová, 1999).

[4] Concerning the terms *represented* and *described*: An expression is *represented* by a formal object in the model, which in turn is *described* by the grammar. Although FGD does not make explicit reference to the ontological assumptions made in §3.1.1, they can be adopted easily.

is reserved for the *form* of content, for content as structured by language, in other words for literal or descriptive meaning, *Bedeutung*. This distinction delimits the domain of linguistics from the domains of other disciplines (such as that of cognitive science).[5]

3.1.3.3 Content words vs. function words

Morphemes – as functions of morphs, the elementary morphological units of language – are classified as autosemantic (content morphemes) and synsemantic (function morphemes). Function morphemes include affixes and endings as well as auxiliary verbs, conjunctions, prepositions and function words in general. Content words consist of a single content morpheme or – in the case of complex words – of several content morphemes. Unlike content words, function morphemes (including function words) do not enter syntactic relations directly, but rather are represented as annotations of a content word.[6]

3.1.3.4 Dependency syntax

Syntactic relations are treated as dependencies between a governor and its modifier (complement or adjunct) and represented at the level of tectogrammatics (and surface syntax) as edges connecting nodes in a (projective) tree graph. The nodes correspond 1:1 to actually present or elided content words of the represented expression and are labelled by 'complex symbols' expressing facts about the corresponding item, as represented at the level of description. Edges connecting the nodes are labelled by names of syntactic functions of the dependents.[7]

---

[5]It is not clear whether *content* is neutral with respect to languages. This seems to be an empirical issue to be decided by psycholinguistic experiments. If, in this crucial aspect, the notion of content were to differ from the notion of meaning only in degree, then the distinction between linguistic and extra-linguistic domains would be blurred. As the framework assumed for the formalization of the theory can accommodate additional levels, decision in one way or the other will not undermine the approach.

[6]This approach fits well with the traditional view that analytic forms belong to the domain of morphology. However, if syntax is viewed as "a branch of grammar dealing with the organization of words into larger structures" (Trask, 1993, p. 94) or – as Crystal (1997) puts it: "Syntax is the way in which words are arranged to show the relationship of meaning [...] within sentences," noting that the original Greek word *syntaxis* means 'arrangement', then function words have their role in syntax as well. Indeed, description of word order is not complete without specifying the placement of function words, in some cases the position of content and function words exhibits the same regularities (clitics). As the theory will be formalized in a way which does not insist on making a sharp split between phenomena belonging to one level or the other, it is not necessary to decide the issue here.

[7]The presentation of FGD's view of syntactic relations by means of tree graphs should actually belong to a section concerned with the way the theory is formalized. However, at this point it seems easier to talk about the theory itself and its formalization at the same time.

Constraint-based formalisms are more expressive than tree graphs, so it is not difficult to model tree structures as a representation of expressions on a certain level.

3.1.3.5 Inner participants vs. free modifications

All syntactic relations are classified as syntactic functions (roles) of dependents with respect to their governors. At the underlying level (tectogrammatics), two types of functions are distinguished: inner participants (arguments, complements) and free modifications (adjuncts).[8] This distinction also corresponds to Tesnière's *actants* and *circonstants* (Tesnière, 1959), the main criteria being

(a) whether a dependent of the given type can occur with a governor at most once (→ inner participant) or more than once (→ free modification), and

(b) whether the type of dependent can occur with every verb (→ free modification) or only with a subset of verbs (→ inner participant).

3.1.3.6 Obligatory vs. optional dependents

The distinction between inner participants and free modifications is orthogonal to the distinction between obligatory and optional dependents. Thus, some inner participants can be optional and some free modifications can be obligatory. Since the opposition between optionality and obligatoriness concerns dependents at the underlying level (tectogrammatics), which do not necessarily correspond to their surface counterparts (typically in cases of ellipsis), a *dialogue test* is used to identify a missing item as an obligatory dependent. If a question asking about a missing item cannot be reasonably

[8] There have been arguments presented against this major distinction. See, e.g., the 'adjuncts as complements' proposal of Przepiórkowski (1999b). The issue is also discussed by Sgall, Hajičová, and Panevová (1986, p. 127–130). Although the distinction of complements and adjuncts is traditionally favoured by constraint-based grammar formalisms, the other path is possible as well. The usual solution is based on lexicon viewed as a dynamic device, providing lexical categories with valency requirements corresponding to the number of dependents, including adjuncts. In fact, valency in FGD is understood as including adjuncts (Sgall, 1993, pages 43–44):

> It is important to notice that data on valency can be specified in a similar way even in such cases where they are not restricted to relatively small groups of lexical items functioning as heads. Thus, not only *arguments* [...], but also adverbial complementations [...] can be specified by means of a list, common to all verbs (or nouns, adjectives, etc.) and determined by grammar; this list can be activated e.g. [...] by a parser, similarly as the data from the lexical entry are activated.

It seems that the above view is not far from the lexical 'adjuncts as complements' approach. More about valency in FGD will be said in §3.1.3.10.

answered by "I don't know", then the item missing on the surface has to be present at the underlying level:

(2) a. A: He has left already.
b. B: When?
c. A: I don't know.

(3) a. A: He has left already.
b. B: Where from?
c. A: I don't know.

In (2) A's answer makes sense: the knowledge that somebody has left does not imply the knowledge about the time of the event and therefore the temporal specification with the verb *leave* (free modification of Time) is an optional dependent. However, in (3) A's answer is incoherent – the knowledge that someone has left implies the knowledge about which place the person has left and therefore the local specification with the verb *leave* (free modification of location) is an obligatory dependent.

Although the distinction between obligatory and optional dependents cannot be expressed within a standard constraint-based formalism in a straightforward fashion, it can be handled lexically.

3.1.3.7 Deletable vs. non-deletable dependents

The fact that obligatory dependents need not be present on the surface results in yet another distinction, namely that between *deletable* vs. *non-deletable* items. This distinction makes sense only for obligatory dependents. A non-deletable dependent must be realized on the surface, a deletable dependent need not be realized. For example, the Czech verbs *znát* 'know *somebody/something*' and *vědět* 'know *that-clause*' differ only in that while the inner participant corresponding to object is obligatory with both, only *znát* requires its presence on the surface.

Indeed, the degree of deletability is widely different across languages due to their grammar constraints, with languages such as English near one extreme, where participants are almost always realized, and Japanese near the other, where it may be difficult to find a non-deletable dependent.

Similarly as in the case of obligatory/optional distinction, deletability can be handled lexically.

3.1.3.8 Syntax-based typology of inner participants

The first two argument functions (Actor and Objective) are defined by purely syntactic criteria, while the classification of adjuncts and the rest of the argument functions has an additional semantic motivation. In both cases, the repertoire of the roles is language-specific, although for related languages there may be no or very few differences.

The classification of argument functions is often presented against the background of Charles Fillmore's deep cases or Noam Chomsky's θ-roles, which are seen as belonging to the (extralinguistic) domain of content (Sgall, Hajičová, and Panevová, 1986, p. 123), while FGD functions represent these cases or roles as structured by a specific language. On the other hand, the classification differs from a more surface-oriented perspective. Thus in FGD, subject of a passive verb and object of an active verb receive the same underlying function.

FGD postulates five types of *inner participants* (underlying or argument functions): Actor, Objective (or Patient), Addressee, Origin and Effect. Two additional ones – Identity and Partitive – are restricted to dependents on nouns (Identity), and nouns and adjectives (Partitive). The types of inner participants have their prototypical counterparts among roles at the higher level of content and among items at the lower levels of surface syntax or morphemics. However, these primary correspondences are overridden in favour of the *shifting* principle based on a partial ordering of underlying functions, which must be assigned in that order. The order is partial because Addressee, Origin and Effect are not ordered with respect to each other. The metaphor of shifting has its origin in the correspondence between the layer of content and tectogrammatics: if – as a result of the prototypical correspondence between the content roles and the underlying functions – an underlying function higher up in the hierarchy would be left vacant while a lower underlying function would be filled, than the dependent is assigned the higher function in preference to the lower one. This is where the theory chooses a syntax-based solution, at least as far as the assignment of the first two underlying functions is concerned.

1. **Actor** – the primary function of subject in the active clause. If a verb[9] has a single inner participant, then it must be Actor:

   (4) Joe$_{Actor}$ sleeps upstairs.

   Otherwise (with other than intransitive verbs), Actor is the participant which 'behaves syntactically in the same way' as the single participant (with an intransitive verb):[10]

[9] Other word classes can also be syntactic governors, whose dependents are assigned functions to a large extent overlapping with those occurring with verbs. Distinctions among functions of dependents on other than verbal word classes are made on the basis of analogy with the classification of dependents on verbs and by independent syntactic criteria.

[10] Sgall, Hajičová, and Panevová (1986, p. 126) put it this way: "[...] Actor is that participant which exhibits the same linguistic structuring (surface representation, distribution) as does the prototypical one-participant verb Actor [...]" However, with some Czech verbs of attitude (*líbit se* 'like') and bodily states (*svrbět* 'itch') Actor may be a dependent in other than the nominative case, which is the prototypical case for subject (Panevová,

(6) Joe$_{Actor}$ understands the problem.

2. **Objective** or Patient – the primary function of direct object, the object affected by the action (often corresponding to the roles Undergoer or Goal). If a verb has two inner participants, Objective is the participant different from Actor.

   (7) Joe$_{Actor}$ understands the problem$_{Objective}$.

   If there are more than two participants, a reasoning similar to that used for determining Actor with transitive verbs is applied: Objective is the item which 'has the same linguistic structuring' as the item classed as Objective with monotransitive verbs (i.e. verbs with two participants):

   (8) Joe$_{Actor}$ gave his girlfriend a golden ring$_{Objective}$.

3. **Addressee** – the function corresponding to indirect object. The referred entity is prototypically animate.

4. **Origin** – the function corresponding to the source of the activity or state, e.g.: build something$_{Objective}$ from something$_{Origin}$, hand something$_{Objective}$ over from someone$_{Origin}$ to someone$_{Addressee}$

5. **Effect** – the function corresponding to object complement (in terms of Quirk et al. (1985)): they elected him *the vice-chair*, or to adverbials of result: he tore it *to pieces*.

6. **Identity** – only for dependents on nouns, expresses identity: the city *of London*, the notion *of God*.

7. **Partitive** – only for dependents on nouns and adjectives, expresses measured material or measured abstract notion: a bunch *of flowers*, full *of hope*.

1998):

(5) a. Líbí se mi$_{Actor}$ tady
like-3SG REFL I-DAT here
'I like it here'

b. Svrbí ho$_{Actor}$ dlaň$_{Objective}$
itch-3SG he-ACC palm-NOM
'His palm is itching'

In (5a) Actor is the dative personal pronoun while Objective is missing on the surface and is treated as a general participant (cf. the English translation with 'it'). In (5b) Actor is the accusative personal pronoun and Objective is the nominative noun 'palm'. Here semantic criteria prevail. Fortunately, the relationship between morphosyntactic aspects of valency (subcategorization requirements) and the underlying functions can and will be described in a redundancy-free lexical fashion (see §3.2), which can accommodate various theoretical assumptions about the classification of underlying functions.

Similarly as in other lexically oriented frameworks, a constraint-based grammar allows for lexical specification of deep cases or underlying functions. The mapping between morphosyntactic features of dependents and deep cases or underlying functions ('*linking*') can be expressed as constraints related to a specific word class, as proposed by Davis and Koenig (1999).

3.1.3.9 Semantically motivated types of free modifications

According to Sgall, Hajičová, and Panevová (1986, pages 159-161), the classification is based on structural properties of dependents, their position in *systemic ordering* (see below) and affinity of their meanings. The total number of types of free modifications is a matter of ongoing empirical research and varies across languages.[11] The types can be presented – following Sgall, Hajičová, and Panevová (1986) in groups:

1. **temporal**: *when, since when, till when, how long* (duration), *for how long* (intention)

2. **manner**, **regard**, **extent** (to the last penny), **norm** (in accordance with), **criterion** (according to), **substitution** (instead of), **accompaniment** (with his dog)

3. **means** (instrument), **difference**, **benefit**, **comparison**

4. **location** (where), **direction**: *from where, which way, where to*

5. **condition**, **cause**, **aim**, **concession**, **result**

6. modifying nouns only: **appurtenance** (a leg *of the table*, *Paul's* brother), **general relationship** (restrictive: *three black* cats, those *without a ticket*), **descriptive property** (non-restrictive: *damned* weather)

The way FGD treats adjuncts as being selected by valency requirements of their syntactic governors provides motivation for adopting a corresponding constraint-based approach. The one at hand is the lexical approach treating adjuncts similarly as complements. Nevertheless, the standard approach

[11] In English, verbs can govern as many as 27 types of free modifications (Hajičová and Panevová, 1984, pages 183-184). Sgall, Hajičová, and Panevová (1986, pages 198-199) and Petkevič (1993, pages 42-44) give 28 types for Czech, with three additional types modifying nouns only. According to Sgall (1993), the necessary number of all types of dependency relations (obviously including inner participants and relations involving non-verbal governors) is at least 40 for most European languages. The total number of types of free modifications would then be at least 34. For Czech, Petkevič (in press) presents altogether 38 types of dependency relations, including 6 types of inner participants. From the remaining 32 types of free modifications, 6 types are subdivided each into 2 subtypes. Thus, the total number of free modifications for Czech is at least 38. In §C all these types are listed according to *systemic ordering*.

of constraint-based grammars with adjuncts selecting their governors is a possible alternative.

For any practical purpose, the determination of a type of free modification is a very difficult task, because – in addition to the governor's word class and morphosyntactic setting – also semantic class of the modification is involved (usually of the noun in prepositional case).

3.1.3.10 Lexicalism

FGD assumes that lexicon is the central descriptive device. Following the tradition of European classical linguistics, the content word is viewed as the principal building block of sentence structure and the lexicon as the adequate location for storing information related to it. According to the usual understanding of valency in FGD, a lexical item which can stand as a governor at the underlying level is assigned a valency frame in the lexicon. This valency frame lists all possible dependents, including free modifications, together with the information about their optionality or deletability. As a result, grammar specifications ('rules of grammar') can be very general. In order to resolve the redundancy due to the presence of the identical list of free modifications with every lexical item of a given part of speech, only a list of inner participants and obligatory free modifications is assumed to occur in a lexical entry (see footnote 8 on p. 44).[12]

A lexical entry in FGD (Sgall, 1993) contains the following parts:

1. Underlying representation of the lexical unit itself, i.e. its lexical meaning. Ambiguous (homonymous) items (*round*) receive multiple entries, unlike vague items (*we*) or items differing only in their valency frames (*swarm, load*).

2. Specification of the values of relevant grammatical categories – *grammatemes*, belonging to a given word class: number and definiteness with nouns, tense and modalities with verbs, degree with adjectives.

3. Valency frame, i.e. the list of possible dependents, ordered in accordance with *systemic ordering* (see below in §3.1.3.11). The individual types of possible dependents bear indications of obligatoriness, deletability, control (obligatory or optional),[13] or other syntactic be-

[12]Petkevič (in press) distinguishes *case frame* and *extended case frame*. Only the latter includes optional free modifications.

[13]At this point, FGD differs in strategy from constraint-based frameworks: whereas in FGD an immediate dependent *cannot* control a deeply embedded dependent *unless explicitly specified*, in a constraint-based framework an immediate dependent *can* 'control' (= be co-indexed with) a deeply embedded dependent *unless specified otherwise*. (Obligatory control is expressed by co-indexing, which means in effect the exclusion of any other possibility.) Therefore, the concept of *optional* control makes sense in FGD (with the intended interpretation that any entity or an immediate dependent X, but not any other

haviour (possibility of standing as Subject, a *wh*-element, a barrier for movement).

4. Subcategorization conditions corresponding to the individual items in the valency frame, i.e. their possible morphosyntactic features.

The lexicalist character of FGD fits well with a constraint-based architecture. However, there is one point in which the approaches to lexicon differ: in FGD, function words are treated on a par with bound morphemes, i.e., during the transition between the level of morphemics and the level of surface syntax (or tectogrammatics). Function morphemes (no matter whether they have the form of affixes, endings or function words) are not treated as lexical elements: they are not represented as nodes in the syntactic tree and they have no place in the lexicon. The principal argument for this treatment is based on the observation that individual languages typically employ very different means for expressing grammatical categories such as types of dependency relations, tense or definiteness. Another argument concerns combinatory possibilities function words – unlike content words – have only restricted combinatory properties.

From a certain viewpoint it certainly is not relevant whether the categories mentioned above are expressed by function words, word order, endings or affixes. However, various ways of expressing the same categories do exist and the division between bound and free morphemes is syntactically relevant (at least from the view of surface syntax). In some respects, function words form a natural class with content words as free morphemes. For example, there are certain word-order regularities concerning clitics, no matter whether they are function or content words. Also, subject-verb agreement works in the same fashion for both function verbs and content verbs.

In a constraint-based framework, a function word can be represented as an item whose contribution to the meaning of a larger expression is indirect, through the meaning of a content word whose *grammateme*(s) the function words specifies. Similarly as other lexical items, it can be represented (modelled) as a multidimensional object, with the dimensions corresponding to different levels of description. Of course, at the levels where FGD represents function morphemes uniformly as annotations of content words, the corresponding dimensions of the function word model could either be vacuous or could refer to the content word. As a result, in a larger expression including the function word, the function word would be represented in the corresponding dimension (= level) as annotation (grammateme(s)) of a content word.

As such, the function word has a place in the lexicon. In a constraint-based framework, this allows for making it subject to any general word-order

dependent, can control an embedded dependent), but not in the other constraint-based approaches (where it must be transformed to a negative statement).

or agreement regularities.

3.1.3.11 Functional sentence perspective as an aspect of grammar

Functional sentence perspective (also information structure, topic-focus or theme-rheme articulation) is generally accepted as a factor governing pragmatic plausibility of an utterance within a given context and situation. However, more importantly for syntactic research, it is reflected in the grammatical structure of the language. It is also relevant for semantic interpretation.

FGD argues that the distinction (or a range) between 'given' and 'new' information, as perceived by the speaker to be present in the hearer's mind, is immediately relevant for (although not always univocally expressed by) various linguistic phenomena. Such phenomena are in fact means of expression. Functional sentence perspective is thus a legitimate domain of linguistic studies and its manifestations must be described within grammar.

The approach is rooted in the assumption that the speaker has a notion of the hearer's stock of available information, and of its partial order – *salience hierarchy*, which is determined to a large extent by preceding context, but also by the discourse situation, items of general knowledge and elements present in the utterance itself (pronominals, spacial and temporal indices).

In a specific utterance, an item which is sufficiently activated (sufficiently high in this ordering) can be considered as *contextually bound* (CB). Typically and in many languages, such an item tends to be placed at the beginning of the utterance. Other items which are not activated enough or not activated at all can be treated as *contextually non-bound* (NB). Again, in many languages such an item tends to be placed at the end of an utterance and can become the *intonation centre* of the utterance.[14] A sufficiently activated item can also become contextually non-bound if it is contrasted with some other available items.

The dichotomy of CB and NB items can be subjected to testable criteria. Two such tests have been proposed: a question test, where only an item absent from the question can be understood as NB (see example (9)), and a 'natural' negative response test (10). The NB item is in italics.[15]

(9) a. Boris travelled from Tbilisi to Grozny *by bus*.
 b. *How* did Boris travel from Tbilisi to Grozny?
 c. Boris travelled by bus to Grozny *from Tbilisi* .
 d. *From where* did Boris travel by bus to Grozny?

(10) a. Boris travelled from Tbilisi to Grozny *by bus*.
 b. No, he travelled from Tbilisi to Grozny *by plane*.

[14] See Preinhaelterová (1999) for some doubts about the status of *intonation centre*, based on an empirical study.

[15] It is assumed that the sentences are pronounced with an unmarked intonation, with the 'intonation centre' at the final item.

c. Boris travelled by bus to Grozny *from Tbilisi* .
d. No, he travelled by bus to Grozny *from Gori*.

Other sentences can be ambiguous in this respect. As (11) and (12) show, in the declarative sentence (11a) the range of non-bound items can span any final segment of the sentence, including the whole sentence.

(11) a. Boris travelled by bus from Tbilisi to Grozny.
b. Where did Boris travel by bus from Tbilisi?
c. From where to where did Boris travel by bus?
d. From where to where and how did Boris travel?
e. What did Boris do?
f. What happened?

(12) a. Boris travelled by bus from Tbilisi to Grozny.
b. No, he travelled by bus from Tbilisi to Gori.
c. No, he travelled by bus from Kutaisi to Gori.
d. No, he travelled by train from Kutaisi to Gori.
e. No, he called by mobile phone from Kutaisi to Gori.
f. No, Nina called by mobile phone from Kutaisi to Gori.

Topic (also *theme* or T) of the sentence can be defined as its contextually bound part and *focus* (also *rheme* or F) as its contextually non-bound part, where dependent items belong to the topic or the focus part according to their governors.

In addition to the topic/focus dichotomy, the content words of an utterance (nodes of the underlying tree representing the utterance) are ordered, constituting the hierarchy of *communicative dynamism* (CD) or *deep word order*, where the topic proper is the least dynamic item and the focus proper the most dynamic item of the sentence. The verb is assumed to be either the most dynamic item of the topic or the least dynamic item of the focus[16] and in the typical case, a dependent is more dynamic than its governor.

In the topic part, CD corresponds to the salience hierarchy of the given items, while in the focus part CD is determined by *systemic ordering* (SO). According to FGD, SO can be stated for a given language as a total ordering of all types of dependents (both inner participants and free modifications), irrespective of the governor. To determine the ordering it is necessary to perform empirical tests to see the types of dependents in an 'unmarked' order, i.e., in the focus part. Once determined, SO can be used to distinguish

[16] In examples such as (13b) the item intervening between the verb and the focus is assumed to precede the verb in the deep verb order. Its position in the surface string is mediated by a movement rule.

(13) a. *What* did Mary give John?
b. Mary gave John a *book*.

topic and focus in a given sentence: the boundary lies after the item which violates it.

The hierarchy of communicative dynamism is closely related to the surface word order, especially in the so-called free word-order languages, where the surface word order is not used for expressing other grammatical categories (such as syntactic relations) to the extent typical for languages like Chinese or English. However, word order is not the only means for expressing CD and – more generally – functional sentence perspective, especially in those 'less word-order-free' languages: prosody is an equally important vehicle. Additionally, some languages employ specific morphemes to mark topic or focus, pronouns can have different forms depending on whether they are CB or NB, and various syntactic constructions can be used to mark specific items as more or less dynamic.

In FGD, functional sentence perspective is described and modelled at the level of tectogrammatics. The scale of communicative dynamism is expressed by means of the left-to-right ordering of nodes of the dependency tree: less dynamic items precede more dynamic ones. The distinction of contextual boundness or topic/focus articulation is expressed by marking the nodes of the tree.

Several attempts have already been made to formalize functional sentence perspective (information structure) within constraint-based frameworks (Elisabet Engdahl and Enric Vallduvi, 1994). Within HPSG and LFG, hierarchies similar to systemic ordering (although mostly not postulated as universal for a language) have been used as a basis for solutions to various phenomena (word order, binding). This shows that a constraint-based formalization of the FGD concepts related to functional sentence perspective and its reflection on the surface could be feasible. Indeed, in cases where the surface and deep word orders correspond the solutions may not be difficult. A more challenging task is to relate the scale of communicative dynamism, topic/focus distinction and systemic ordering with other syntactic constraints.

3.1.3.12 Options for formalization

FGD has already been recast in different formal frameworks. Various tools have been used: push-down store automata, context-free grammars, transducing components, movement rules, unification, the formalism of categorial grammar. Here I am going to explore the hypothesis the foundational concepts of the theory can be expressed within a constraint-based setting.

3.2 Formalism

This section provides a formal ground for the subsequent linguistic enterprise. This involves:

- providing definitions of the following two kinds of metalanguage:
  - structures used as representations of language expressions, as a *model of language*;
  - statements of a grammar used as a *description of language*, constraining the model;
- explicating what kinds of relations hold:
  - between the two sets of formal objects: the description and the model;
  - between the model and objects of 'real' language: expression tokens.[17]

Specific linguistic analyses will be provided as illustrations for the use of a formal metalanguage. However, linguistic analysis will not be the primary issue.

An important aspect of the relationship between linguistic theory and formalism will be pointed out in §3.2.1, namely the fact that at least in some cases a theory can be spelled out in more than one formal language. It will be argued that this is the case for the core of FGD as the theory and the constraint-based formalism underlying HPSG. In the following, foundations of the formalism will be defined and illustrated by examples. In §3.2.2, an example simple sentence represented in an FGD-like dependency tree notation will be translated into an equivalent feature structure. Next, in §3.2.3, Speciate Re-entrant Logic (SRL) will be presented as a formal language for representing and describing feature structures, with examples of its use in grammars, i.e., in feature structure descriptions. After listing some cases where SRL falls short of its intended role as a formalism for HSPG, an extension of SRL featuring relations and quantifiers, Relational Speciate Re-entrant Logic, will be informally presented in §3.2.4. In §3.2.5, attribute-value matrices and feature declarations with will be introduced as a more convenient equivalent of the original (R)SRL notation. The simple dependency tree used as the example in §3.2.2 will be replaced by a slightly more complex dependency tree with more realistic FGD-like analysis, again with an equivalent feature structure in §3.2.6. In the same section, abbreviatory conventions will be listed for translating from the full RSRL notation into the AVM notation as used in standard literature, and a few examples of RSRL principles and relations will be given. The final two short sections 3.2.7 and 3.2.8 will relate the formalism to linguistic practice. This will concern especially the individual components of the formal language and the possibility to treat levels of descriptions as properties of a single formal object.

[17] I will not have much to say about psycholinguistic issues, which is a good excuse for ignoring the relation between *formal grammar* and *mental grammar* (a hypothesized entity inside a language user's head).

3.2.1 Theories and formalisms

Linguistic theories differ in the degree of formal rigour. Their metalanguage is often presented without making semantics or even syntax of the metalanguage explicit. Although both may be obvious, understanding the metalanguage is often possible only due to a linguist's (rather than language user's) intuition, which is conditioned by a specific theoretical tradition and viewpoint. Sometimes, a specific usage of metalanguage in the model or the description is presented as the formalism (e.g., principles of grammar are treated as a part of the formalism). Sometimes, the metalanguage is just a carefully used natural language.[18]

An influential branch of computationally oriented linguistic formalisms was based on augmented versions of context-free grammar or an equivalent framework.[19] On the other hand, specific requirements of linguistic theories gave rise to description tools such as transformations or movement rules. Dependency-oriented theories have developed formalisms suitable for representing and describing language without the employment of nonterminal categories, at least not in the representation metalanguage.

The computationally oriented formalisms mentioned above were quite influential in laying foundations for theories based on grammars of the constraint-based type.[20] While taking the issue of proper formalization seriously, these theories have been able to propose successful accounts of non-trivial linguistic phenomena relying on the constraint-based architecture, where:

- language expressions are represented by means of sets of attributes and their values, i.e., feature structures,[21] which have additional power due to the possibility of

- recursive embedding,

[18]This is the solution adopted by Pollard and Sag (1994, page 7):

> "[...] Moreover, we require that the theory itself actually count as a theory in the technical sense of precisely characterizing those modelling structures that are regarded as admissible or well-formed [...]. This does not mean that the empirical hypotheses must be rendered in formal logic as long as their content can be made clear and unambiguous in natural language (the same holds true in mathematical physics), but in principle they must be capable of being so rendered."

[19]Augmented Transition Networks – ATN of Woods (1970), Systèmes Q of Colmerauer (1970), Metamorphosis Grammars of Colmerauer (1978) or Definite Clause Grammars – DCG of Pereira and Warren (1980), Functional Unification Grammar – FUG of Kay (1983), and PATR-II of Shieber et al. (1983) can all be quoted as examples.

[20]In addition to LFG and HPSG mentioned below, GPSG (Gazdar et al., 1985) and some versions of categorial grammar (Uszkoreit, 1986) can be given as examples.

[21]See (16) and (17) below for examples of two ways of depicting a feature structure.

- structure sharing, and
- imposing a partial order for such structures, resulting in a subsumption hierarchy, which corresponds to the amount of information contained in the structures.[22]

The formalism of Lexical Functional Grammar evolved during the 1970s from the Augmented Transition Networks (Woods, 1970) and found its first comprehensive presentation in Kaplan and Bresnan (1982), for a more up-to-date overview see Dalrymple et al. (1995). The LFG formalism is closely tied to LFG as a linguistic theory. There are two levels of syntactic representation: constituent structure and functional structure. The levels are related by links that permit the properties of the abstract functional structure to be defined in terms of phrase structure configurations. LFG distinguishes between (i) structures as linguistic representations (constituent and functional structures), (ii) descriptions as the language that characterizes the structures, and (iii) structural correspondences, functions which map nodes of a constituent structure into the items of a functional structure. This simple conceptual foundation proved to be very successful, both linguistically and computationally.

Compared to Lexical Functional Grammar, the theory of Head-driven Phrase Structure Grammar – following Kay (1983) – employs a more uniform formalism. Variations of a single type of metalanguage (feature structures) are used not only to constrain collocations of morphological, syntactic and semantic information, but also to specify lexical entries, grammar rules and principles, and to represent syntactic tree and its semantic interpretation. Where other linguistic theories explicitly employ trees (rooted directed graphs which satisfy the condition of a single mother) with labelled nodes (sometimes also edges), HPSG makes use of feature structures. However, the interpretation of the role of formal structures used as representations and those used as description language has developed: they are notationally similar, but are now viewed as completely distinct entities. Still, their notational uniformity is attractive for both theoretical and implementational reasons and the formal foundations assumed in the more recent versions of HPSG will be adopted also here.

In fact, HPSG can be viewed both as a linguistic theory and a linguistic formalism. Uszkoreit (1996) used the metaphor of toolbox for HPSG as a formalism and Przepiórkowski (2000) argues for distinguishing the two aspects. Of course, the theory of HPSG is mostly expressed by the formalism

[22] In recent versions of HPSG (starting with Pollard and Sag (1994)), where the term *feature structure* is reserved for formal objects in the model, while objects used in the description are called *feature descriptions*, there is a notable difference between the two in that only the latter can be partially ordered in a subsumption hierarchy. Feature structures are always – in the terminology of Carpenter (1992) – totally well-typed and sort-resolved, i.e., they correspond to maximally specific items of a subsumption hierarchy.

of HPSG, but examples of HPSG-based analyses can be found within other frameworks[23] and the formalism of HPSG can be found as the formal framework for other theories (an example of this can be found in Kay (1998)). The recent proposals for an HPSG formalism (such as RSRL of Richter (2000)) are actually well disposed for application in other theoretical contexts: they are based on the view that the source of constraints must be in the content of grammar, not in its form, i.e., in the formal language used to express the constraints. In other words, restrictions of the formalism should not prevent linguists from expressing their generalizations in a natural way. While a less restrictive formalism makes implementation more difficult and human processing based on such grammars less straightforward to imagine, it has better prospects for being used as metalanguage in the context of other linguistic tradition. Although there are restrictions inherent in any formalism, which are eventually determined by theoretical motivations, a less restrictive version of HPSG formalism will be shown to provide a suitable framework for formalising grammars based on the theoretical insights of FGD. Arguably, the actual restrictions, represented mainly by strong typing, attribute-value notation and monostratality, do not contradict the theoretical stance of FGD. It will be demonstrated that the formalism actually provides formally and theoretically superior solutions to some issues in FGD, such as the relation between the level of underlying syntax and the string of morphemes.

In the earlier version of HPSG (Pollard and Sag, 1987), the same formal objects are used both (i) as models for *partial information* about language entities and (ii) as their descriptions (in the grammar, including the lexicon). Feature structures are used to describe and model a language entity at the same time. The grammar does not describe tokens or types of entities of a language directly. The grammar merely describes *information* about those entities, and objects (feature structures) in the model, representing the various collections of pieces of information about an entity, are partially ordered according the amount (specificity) of information in a subsumption hierarchy.

In more recent versions of the theory (Pollard and Sag, 1994), there is a sharp distinction between the model and the description, i.e., between **feature structures** and **feature descriptions**. A feature structure now serves as the canonical representative for a set of language entity tokens which are undistinguishable from a certain viewpoint, typically from the viewpoint of a grammarian interested in linguistic competence.[24] A feature structure includes complete information about the entity (is 'maximally specific') and the subsumption ordering holds only among feature descriptions. Feature descriptions generalize over feature structures, in other words, constrain them

[23]Przepiórkowski (2000) provides as an example the HPSG-inspired but (formally) Minimalistic analysis of raising and control in Hornstein (1999).

[24]The set of language entity tokens can be exemplified by a set of tokens of the utterance *my guinea-pig likes carrot* pronounced by various speakers at various times.

(or 'generate', in the sense of generative grammar) to eliminate those which do not correspond to any well-formed language expression.

Feature descriptions (*constraints*) represent one part of a specific grammar. This part is also called *theory*. The other part, called *signature*, defines elements of the formal language, which are used in the theory. These elements are called *types* and correspond to sets of potential language objects, such as words, constructions, morphological categories of a given type, etc. In the signature they are ordered in a *type hierarchy*. The hierarchy is partially ordered according to the specificity of types. At the same type, the set of potential linguistic objects represented by a type is exhaustively partitioned by subtypes representing disjoint subsets of the set. Properties of the objects are formally expressed by attributes, which are defined as appropriate for a type and therefore for all its subtypes. Values of these attributes are again types, and if an attribute of a type has a certain value, this attribute in a subtype must have a value which is at least as specific.

3.2.2 An example: representing dependency tree as a feature structure

Let us consider a simple dependency tree (Fig. 3.1), corresponding to a simplified version of underlying or tectogrammatic structure, as proposed by the theory of Functional Generative Description (FGD). The tree represents a Czech sentence in 15).[25]

(15) Máňa šla tancovat
Máňa-NOM go-PAST dance-INF
'Máňa went to dance'

The nodes of the tree are labelled by complex symbols, consisting of lexeme, word class and a grammateme.[26] The edges are labelled by underlying functions performed by dependent nodes. The symbol *COR* (standing for

[25] An equivalent canonical representation in FGD uses linearized notation. According to conventions introduced in Sgall (1995), the syntactic tree (Fig. 3.1) would be linearised as follows (the expression is broken into indented lines for readability):

(14) (
(*Máňa*.Noun.Singular $)_{Actor}$
jít.Verb.Anterior
$(_{Patient}$ (*COR* $)_{Actor}$ tancovat.Verb.Posterior)
)

The brackets indicate levels of embedding within the tree. Every pair of brackets except for the top pair carries a subscript, denoting underlying syntactic function of the embedded subtree, or – in the strictly dependency-based view – of the top node of the embedded subtree. The order of bracketed strings corresponds to the horizontal order of the respective subtrees in the tree graph. A subscript annotates the opening bracket if the (top node of the) subtree follows its governor.

[26] The complex labels represent much richer annotation in 'real' FGD.

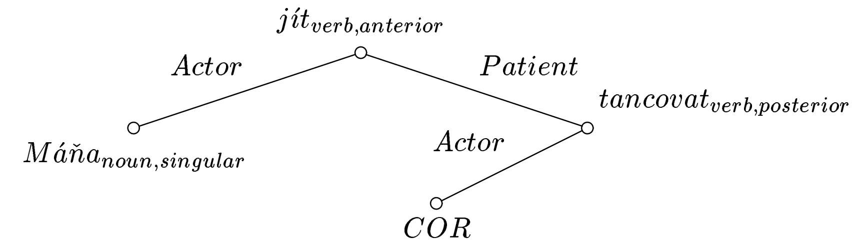


Figure 3.1: A dependency tree

coreference) labels a node which is involved in a syntactic control relation as the controllee. The controller is determined configurationally. In our example, the controller is the main verb's Actor.

The vertical dimension of the tree imposes a partial order among nodes. This dimension represents dependency relations. In addition to the vertical dimension, the underlying tree has another, horizontal dimension, whereby a total order for nodes is expressed, representing *deep word order*. According to the standard version of the theory (Sgall, Hajičová, and Panevová, 1986, p. 152), this order has to obey the condition of projectivity, equivalent to the prohibition of discontinuous constituents in a phrase structure tree. For the moment, the horizontal dimension of the tree will be ignored.[27]

The coreference resulting from the relation of grammatical control, which is expressed in Fig. 3.1 by labelling the controllee node with the symbol *COR*, can be alternatively represented by an index: Actor of the main verb is co-indexed with Actor of the embedded verb, as in Fig. 3.2. In the following, the coindexed version will be assumed.

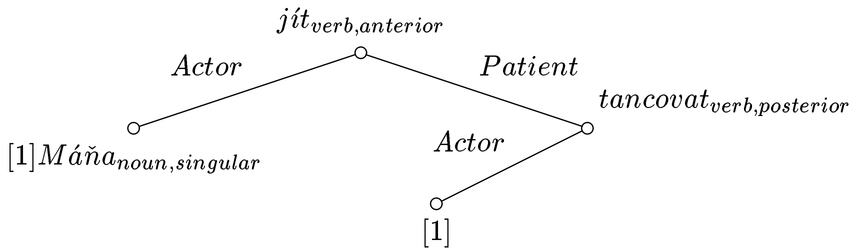


Figure 3.2: A dependency tree with coindexing

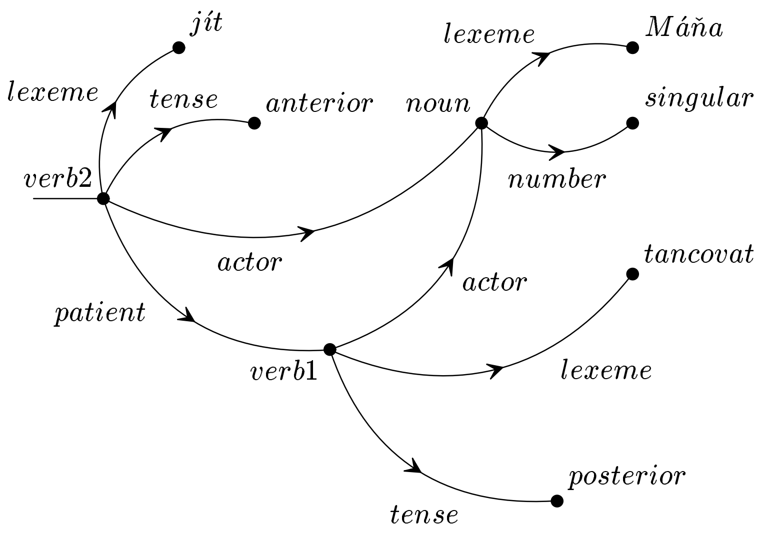
The information encoded in the tree can be represented as a feature structure. Feature structure is a rooted, connected and directed graph. Each edge is labelled by a feature name and each node is labelled by a name denoting a kind of linguistic object, in the terminology of King (1989) a *species* name.[28] The condition of a single mother does not apply to feature

[27] Another issue, which will go unnoticed here, is the treatment of coordinated elements.

[28] A species corresponds to a maximal (maximally specific) *type* or *sort*. In the earlier proposals for formalizing representations and grammar by means of feature structures (e.g., in Kay (1983)) only the most deeply embedded objects – the leaves of the graph,

structures, which allows for expressing identity of feature values by nodes which are pointed to by more than one edge. The feature structure in (16) is a rather naive transcription of the dependency tree in Fig. 3.2.

(16)



The tree in Fig. 3.2 has been transformed into a feature structure in (16) by the following recipe:

1. For every edge of the tree, treat its label as a feature name corresponding to the edge.

2. Treat all nodes which are co-indexed by the same index as a single node.

3. For every node of the tree, select a part of the complex symbol labelling the node as the species name of the node.

4. For every node of the tree and every other part of the complex symbol labelling that node, construct an edge leading from the node to a new node; use the part of the complex symbol to label the new node by a species name; label the edge by an appropriate feature name.

5. If a species name occurs on multiple nodes, check that they all have the same number of edges leading from them and that the edges have the same labels. If this is not the case, select a different species name for

atomic symbols – are named. In order to distinguish a formalism with named objects, the term *typed feature structure* is often used.

the node(s) which differ(s) in the number of edges leading from them or in a label for an edge.[29]

Instead of graphs, feature structures are commonly represented in a boxed notation with attributes and their values, where the attributes correspond to feature names and values of the attributes to species. Every box thus represents a species and includes the appropriate attributes (17).

$$
(17)\quad \left[\begin{array}{ll}
\textit{verb2} & \\
\text{LEXEME} & \textit{jít} \\
\text{TENSE} & \textit{anterior} \\
\text{ACTOR} & \boxed{1}\left[\begin{array}{ll} \textit{noun} & \\ \text{LEXEME} & \textit{Máňa} \\ \text{NUMBER} & \textit{singular} \end{array}\right] \\
\text{PATIENT} & \left[\begin{array}{ll} \textit{verb1} & \\ \text{LEXEME} & \textit{tancovat} \\ \text{TENSE} & \textit{posterior} \\ \text{ACTOR} & \boxed{1} \end{array}\right]
\end{array}\right]
$$

In the transition from the dependency tree to the feature structure certain assumptions have been made:

1. The whole feature structure is a species, consisting of other species as its subparts, identified by attributes – features. Each species represents a set of entities of a certain kind. The subparts of a species can be interpreted as constituent parts of the expression being represented (in our example Actor, Patient), characteristics of a given constituent part (lexeme, tense), or as a specific level of representation of the expression or its constituent part.[30]

   In fact, each constituent part of an expression can be structurally identical to the whole expression. Thus, syntactic units of any size, from sentences down to words, can have subparts corresponding to different levels of representation.

2. There is a certain fixed set of features defined as appropriate for every species. Every entity of the given species has all and only those features. In addition, a certain fixed set of species is defined for the subpart identified by a feature. Every entity which becomes the subpart can only belong to the fixed set of species. Feature structures satisfying this condition are **totally well-typed**.

[29] This is meant to guarantee that the structure meets the conditions of closed world and total well-typedness (see below).

[30] The structure in our example represents a single level. Examples where multiple levels are represented in a single structure will be presented later.

It is because of this condition that the two verbal nodes in the dependency tree are represented as distinct species in the feature structure: two occurrences of a single species could not have different sets of features, one set corresponding to the verb *jít* with two complementations, the other to the verb *tancovat* with a single complementation.[31]

3. Among the subsets of entities denoted by the species, there are no two subsets which have non-empty intersection. In other words, each entity belongs to at most one species. This is the so-called **species disjointness** assumption.

   For example, it cannot be the case that there is an entity of the species *verb2* which is also of the species *verb1*.

4. The universe of linguistic entities is partitioned by a fixed set of species. In other words, each entity belongs to at least one species, there are no entities outside the fixed set of species. (Taken together with the species disjointness condition, each entity belongs to exactly one species.) This is the so-called **closed world** assumption.

In addition to these basic ontological assumptions, objects in the model have to satisfy certain other conditions. Collections of statements used for this purpose are traditionally called *grammars*. According to this view, grammars are sets of constraints which objects in the model have to satisfy in order to be well formed, i.e., in order to correspond to linguistic entities. These constraints can be expressed as axioms of feature logic.

In the following section, the syntax and semantics of a relatively simple formal language will be presented. The formal language embodies the basic ontological assumptions made above and additionally requires explicit specification of types and their properties for objects in the model as a more refined grammar-specific 'ontology'. In fact, only after this specification is built, the formal language can refer to objects in the model within constraints imposed on them and their configurations. Examples will show how the formal language can be used for depicting linguistic entities as such objects, i.e., feature structures, and for writing grammars constraining them.

3.2.3 Speciate Re-entrant Logic

One of the formal languages which can be used for representing and constraining possible feature structures is SRL (Speciate Re-entrant Logic) of King (1989).[32] Other candidates include the attribute-value logic of Johnson (1988) and the feature logic of Smolka (1988). SRL can also be compared

[31] This is an area where the naiveté of the example feature structure shows up: for each verb with a certain number and types of optional free modifications (adjuncts), another verb species would have to be introduced. This might lead to an infinite number of species.

[32] A more recent presentation can be found in King (1999).

to the typed feature logic of Carpenter (1992). The main difference between SRL and the other frameworks is in that only SRL incorporates the assumptions of species disjointness, closed world and total well-typedness. These assumptions hold in all interpretations for SRL expressions.

SRL has been developed specifically with the goal to serve as the underlying formalism for HPSG. More recently, SRL has been extended by Richter (1999; Richter (2000). The extended formalism – Relational SRL (RSRL) – bridges the remaining gap between the expressive power of the formalism and of the semi-formal metalanguage as it is currently employed in the HPSG literature. I will provide some formal background of SRL and show how it can be used to describe tectogrammatical objects. The extensions of SRL will be presented only informally, together with the more intuitive and easier-to-use AVM notation. Formal definition of RSRL and its translation into the AVM notation is available, e.g., in Richter (1999).

A statement in a conventional logic such as first-order predicate calculus is interpreted to denote a truth value. A statement in SRL is interpreted as being either true or false of an object in the domain of possible interpretations, i.e., it denotes a subset of such objects – feature structures as representations of entities.[33]

SRL can be viewed as a tool for building formal languages of a certain class. There are two components of each such language: a syntactic component – its *signature*, and a semantic component – a class of *interpretations*.

The signature provides two disjoint sets of nonlogical symbols: *species* and *features*. There is a relation of *appropriateness* between the two sets: a feature is or is not appropriate to a species. Formulas of the language – *descriptions* – can be constructed as finite and well-formed strings consisting of symbols from the signature and of logical symbols. A set of descriptions is called a *theory*.

Each interpretation provides a set of possible objects and assigns meaning for each nonlogical symbol in the signature. Meaning as a set of objects in the interpretation is assigned also for each description and for each theory via a *description denotation function* and a *theory denotation function*. These functions are determined by each interpretation and select objects of which the description is true or the theory is true. A theory is true of objects just in case each description of the theory is true of them.

Definition 1 *A* **signature** *is a triple* $\Sigma = \langle \mathsf{S}, \mathsf{F}, \mathsf{A} \rangle$, *where*

[33]King (1999) departs from the concept of representations of entities and assumes that the formal language denotes sets of entities, not sets of feature structure representations of entities. This move remedies some methodological weaknesses resulting from the assumptions about the existence of *types* (objects generalizing over species, see below). This issue will be only marginally relevant for the present work. In fact, all occurrences of the word 'object' in this section can be substituted either by 'feature structure' or 'entity', unless stipulated otherwise.

S *is a set, its members are called* **species** *in* Σ*;*
F *is a set of features, its members are called* **features** *in* Σ*;*
A *is a total function from* $\mathsf{S} \times \mathsf{F}$ *to subsets of* S*, called* **appropriateness function** *in* Σ*.*

The following **logical symbols** are used:[34] $\approx, \sim, \neg, \wedge, \vee, \rightarrow, [,]$.

Definition 2 *An* **interpretation** *of* Σ *is a triple* $I = \langle U, S, F \rangle$*, where*

U *is a set, called* **universe** *in* I*, each member of* U *being an* **entity** *in* I*;*
S *is a total function from* U *to* S*, called* **species interpretation function** *in* I*;*
F *is a total function from* F *to the set of partial functions from* U *to* U*, called* **feature interpretation function** *in* I*, and*

for each $\varphi \in \mathsf{F}$*, for each* $v \in U$*,*

$F(\varphi)(v)$ *is defined iff* $\mathsf{A}(S(v), \varphi) \neq \emptyset$*, and*
if $F(\varphi)(v)$ *is defined then* $S(F(\varphi)(v)) \in \mathsf{A}(S(v), \varphi)$*.*

Thus, each species denotes a set of objects in U and the denotations of species partition U. S assigns to each object the unique species to whose denotation the object belongs: species σ denotes the set $\{v \in U | S(v) = \sigma\}$. Saying that entity v is **in** species σ means that $S(v) = \sigma$. Each feature denotes a partial function from objects to objects (from U to U) and F assigns each feature the partial function it denotes. Saying that feature φ is **defined** on entity v and that it **maps** v to entity v' means that $F(\varphi)(v)$ is defined and $F(\varphi)(v) = v'$.

The appropriateness function takes care of the relation between the denotations of species and features. If $\mathsf{A}(\sigma, \varphi) = \emptyset$ then feature φ is not defined on any object in species σ. If, however, $\mathsf{A}(\sigma, \varphi) \neq \emptyset$ then feature φ is defined on each object in species σ and maps each object in σ to some object in $\mathsf{A}(\sigma, \varphi)$.

Taking (16) as an example feature structure again, an SRL language can be defined to constrain the domain of possible feature structures. In the spirit of (16), these feature structures represent a trivial approximation of FGD dependency trees for a couple of trivial expressions.

Let us first define a signature $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$ (see Fig. 3.3). The signature provides species and features and restricts possible interpretations. For example, the species *verb1* cannot have a feature PATIENT, *noun* cannot have TENSE, but it must have NUMBER, which is either *singular* or *plural*.[35]

[34] None of the symbols should be a species or a feature.

[35] Lexical values are treated as individual species, values of the feature LEXEME, which means that the signature puts rather strong constraints on possible interpretations. Another solution would be to introduce a species *string* whose subparts would be species for characters of the alphabet. The actual lexeme could then be specified outside the signa-

$\mathsf{S} = \{verb1, verb2, noun, anterior, posterior, singular, plural,$
$\quad j\acute{\imath}t, tancovat, M\acute{a}\check{n}a\},$
$\mathsf{F} = \{\text{LEXEME, TENSE, NUMBER, ACTOR, PATIENT}\},$
$\mathsf{A}(verb1, \text{LEXEME}) = \{tancovat\},$
$\mathsf{A}(verb1, \text{TENSE}) = \{posterior, anterior\},$
$\mathsf{A}(verb1, \text{NUMBER}) = \emptyset,$
$\mathsf{A}(verb1, \text{ACTOR}) = \{noun, verb1, verb2\},$
$\mathsf{A}(verb1, \text{PATIENT}) = \emptyset,$
$\mathsf{A}(verb2, \text{LEXEME}) = \{j\acute{\imath}t\},$
$\mathsf{A}(verb2, \text{TENSE}) = \{posterior, anterior\},$
$\mathsf{A}(verb2, \text{NUMBER}) = \emptyset,$
$\mathsf{A}(verb2, \text{ACTOR}) = \{noun, verb1, verb2\},$
$\mathsf{A}(verb2, \text{PATIENT}) = \{noun, verb1, verb2\},$
$\mathsf{A}(noun, \text{LEXEME}) = \{M\acute{a}\check{n}a\},$
$\mathsf{A}(noun, \text{TENSE}) = \emptyset,$
$\mathsf{A}(noun, \text{NUMBER}) = \{singular, plural\},$
$\mathsf{A}(noun, \text{ACTOR}) = \emptyset,$
$\mathsf{A}(noun, \text{PATIENT}) = \emptyset.$

For all other species σ (*anterior, posterior, singular, plural, jít, tancovat, Máňa*), the value of $\mathsf{A}(\sigma, \varphi)$ is $\emptyset$, no matter what value φ is.

Figure 3.3: An example SRL signature

However, we do not know yet to which objects the species with their appropriate features refer. This information is provided by the interpretation $I = (U, S, F)$ in Fig. 3.2.3 on p. 66. The objects in U are assumed to be tectogrammatical FGD-like representations of the relevant expressions, i.e., objects of species $verb1, verb2, noun$, linguistic categories such as $anterior$, and lexical values such as $M\acute{a}\check{n}a$. Tectogrammatical representation of an expression will be written in single quotes.[36]

The objects and their configurations (such as the representation of the expression *Máňa šla tancovat*) must conform to the signature (3.3) both in the repertoire of objects (each object must belong to a single species) and in the features and their values appropriate for the objects. In our simplistic domain of a single sentence, the number of objects belonging to a species is rather scarce. Nevertheless, the interpretation does provide a lot of additional information, e.g., objects such as 'Máňa' can be interpreted naturally as singular nouns.

---

ture. However, if lexical values – as carriers of meaning in our simplistic example – are assumed to participate in partitioning the universe of linguistic entities, they do belong in the definition of signature.

[36]The fact that an expression can have more than one representation is ignored in this example. In principle, the objects can also be viewed as the (tokens of) expressions and categories themselves. It must be admitted that intuitively this view would not be unsatisfactory.

U = {'Máňa šla tancovat', 'Máňa', 'tancovat', anterior, posterior, singular, plural, jít, tancovat, Máňa}

Let S be the total function from U to S such that

S('Máňa šla tancovat') = *verb2*,
S('Máňa') = *noun*,
S('tancovat') = *verb1*,
S(anterior) = *anterior*,
S(posterior) = *posterior*,
S(singular) = *singular*,
S(plural) = *plural*,
S(jít) = *jít*,
S(tancovat) = *tancovat*, and
S(Máňa) = *Máňa*.

Let F be the total function from F to the set of partial functions from U to U such that

F(LEXEME) is the partial function from U to U such that

F(LEXEME)('Máňa šla tancovat') = jít,
F(LEXEME)('Máňa') = Máňa,
F(LEXEME)('tancovat') = tancovat,
F(LEXEME) is undefined at {anterior, posterior, singular, plural, jít, tancovat, Máňa}, and

F(TENSE) is the partial function from U to U such that

F(TENSE)('Máňa šla tancovat') = anterior,
F(TENSE)('tancovat') = posterior,
F(TENSE) is undefined at {'Máňa', anterior, posterior, singular, plural, jít, tancovat, Máňa}, and

F(NUMBER) is the partial function from U to U such that

F(NUMBER)('Máňa') = singular,
F(NUMBER) is undefined at {'Máňa šla tancovat', 'tancovat', anterior, posterior, singular, plural, jít, tancovat, Máňa}, and

F(ACTOR) is the partial function from U to U such that

F(ACTOR)('Máňa šla tancovat') = Máňa,
F(ACTOR)('tancovat') = Máňa,
F(ACTOR) is undefined at {'Máňa', anterior, posterior, singular, plural, jít, tancovat, Máňa}, and

F(PATIENT) is the partial function from U to U such that

F(PATIENT) ('Máňa šla tancovat') = tancovat, and
F(PATIENT) is undefined at {'Máňa', 'tancovat', anterior, posterior, singular, plural, jít, tancovat, Máňa}.

Figure 3.4: An interpretation of the SRL signature in Fig 3.3

We shall now define sets of **terms** T_Σ (also called *paths*) and **descriptions** D_Σ.

Definition 3 *For each signature* $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$, T_Σ *and* D_Σ *are the smallest sets such that*

$: \in \mathsf{T}_\Sigma$,
for each $\tau \in \mathsf{T}_\Sigma$, *for each* $\varphi \in \mathsf{F}, \tau\varphi \in \mathsf{T}_\Sigma$,
for each $\tau_1 \in \mathsf{T}_\Sigma$, *for each* $\tau_2 \in \mathsf{T}_\Sigma, \tau_1 \approx \tau_2 \in \mathsf{D}_\Sigma$,
for each $\tau \in \mathsf{T}_\Sigma$, *for each* $\sigma \in \mathsf{S}, \tau \sim \sigma \in \mathsf{D}_\Sigma$,
for each $\delta \in \mathsf{D}_\Sigma, \neg\delta \in \mathsf{D}_\Sigma$,
for each $\delta_1 \in \mathsf{D}_\Sigma$, *for each* $\delta_2 \in \mathsf{D}_\Sigma, [\delta_1 \wedge \delta_2] \in \mathsf{D}_\Sigma$,
for each $\delta_1 \in \mathsf{D}_\Sigma$, *for each* $\delta_2 \in \mathsf{D}_\Sigma, [\delta_1 \vee \delta_2] \in \mathsf{D}_\Sigma$, *and*
for each $\delta_1 \in \mathsf{D}_\Sigma$, *for each* $\delta_2 \in \mathsf{D}_\Sigma, [\delta_1 \rightarrow \delta_2] \in \mathsf{D}_\Sigma$.

Let us now turn to the interpretation of terms and descriptions.

Definition 4 *For each signature* $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$ *and for each interpretation* $I = (U, S, F)$ *of* Σ, *the* **term interpretation function** T_I *maps terms to partial functions from* U *to* U*:*

for each $\upsilon \in U$,

$T_I(:)(\upsilon)$ *is defined and* $T_I(:)(\upsilon) = \upsilon$,

for each $\tau \in \mathsf{T}_\Sigma$, *for each* $\varphi \in \mathsf{F}$, *for each* $\upsilon \in U$

$T_I(\tau\varphi)(\upsilon)$ *is defined iff* $T_I(\tau)(\upsilon)$ *is defined and* $F(\varphi)T_I(\tau)(\upsilon)$ *is defined, and*

if $T_I(\tau\varphi)(\upsilon)$ *is defined then* $T_I(\tau\varphi)(\upsilon) = F(\varphi)T_I(\tau)(\upsilon)$.

The symbol : denotes the identity function on U and the terms consisting of strings of features denote the function composition of the functions denoted by the features.

Definition 5 *For each signature* $\Sigma = (\mathsf{S}, \mathsf{F}, \mathsf{A})$ *and for each interpretation* $I = (U, S, F)$ *of* Σ, *the* **description interpretation function** D_I *maps descriptions to subsets of* U*:*

for each $\tau \in \mathsf{T}_\Sigma$, *for each* $\sigma \in \mathsf{S}$,

$D_I(\tau \sim \sigma) = \{\upsilon \in U \mid T_I(\tau)(\upsilon)$ *is defined and* $S(T_I(\tau)(\upsilon)) = \sigma\}$,

for each $\tau_1 \in \mathsf{T}_\Sigma$, *for each* $\tau_2 \in \mathsf{T}_\Sigma$

$D_I(\tau_1 \approx \tau_2) = \{\upsilon \in U \mid T_I(\tau_1)(\upsilon)$ *is defined*, $T_I(\tau_2)(\upsilon)$ *is defined, and* $T_I(\tau_1)(\upsilon) = T_I(\tau_2)(\upsilon)\}$,

for each $\delta \in \mathsf{D}_\Sigma, D_I(\neg\delta) = U \setminus D_I(\delta)$,
for each $\delta_1 \in \mathsf{D}_\Sigma$, *for each* $\delta_2 \in \mathsf{D}_\Sigma, D_I([\delta_1 \wedge \delta_2]) = D_I(\delta_1) \cap D_I(\delta_2)$,
for each $\delta_1 \in \mathsf{D}_\Sigma$, *for each* $\delta_2 \in \mathsf{D}_\Sigma, D_I([\delta_1 \vee \delta_2]) = D_I(\delta_1) \cup D_I(\delta_2)$,

for each $\delta_1 \in \mathsf{D}_{\boldsymbol{\Sigma}}$, *for each* $\delta_2 \in \mathsf{D}_{\boldsymbol{\Sigma}}$,
$D_I([\delta_1 \rightarrow \delta_2]) = (U \setminus D_I(\delta)) \cup D_I(\delta_2)$.

D_I assigns each description the set of objects of which the description is true. Description $\tau \sim \sigma$ is true of υ iff τ maps υ to an object in σ, and description $\tau_1 \approx \tau_2$ is true of υ iff τ_1 and τ_2 map υ to the same object. The other symbols for negation, conjunction, disjunction and implication are used as in classical propositional logic. Taking up our example again, we can write descriptions such as those in (18).

(18) a. $: \sim$ *verb1* $\vee : \sim$ *verb2*
b. :PATIENT ACTOR $\approx$:ACTOR
c. :TENSE $\sim$ *posterior*
d. :LEXEME $\sim$ *Máňa* $\wedge$:NUMBER $\sim$ *plural*
e. $: \sim$ *noun* $\wedge$:TENSE $\sim$ *posterior*
f. :LEXEME $\sim$ *Máňa* $\wedge$:NUMBER $\sim \neg$*plural*
g. :LEXEME $\sim$ *Máňa* $\rightarrow$:NUMBER $\sim \neg$*plural*

Description (18a) is true of 'Máňa šla tancovat' and 'tancovat', while description (18b) is true only of 'Máňa šla tancovat'. Description (18c) is true of 'tancovat'. Descriptions (18d) and (18e) are both false of any object in the domain, but for different reasons. The former description is false because there is no object in the domain on which the features LEXEME and NUMBER yield the species *Máňa* and *plural*, respectively – courtesy the species interpretation function $S(T_I(\tau)(\upsilon))$. The latter description (18e) is false because the feature TENSE is not defined on species *noun* due to the fact that term interpretation function $T_I(\tau_1)(\upsilon)$ is not defined for the term :TENSE and the object noun. Description (18f) is true of 'Máňa' (and no other object in the domain), unlike (18g), which is true of any object in the domain.[37] Since principles of grammar are assumed to apply to all objects in the domain, they are most naturally expressed as descriptions involving implication in order to pick the objects which the principle is concerned with, while ignoring (being trivially true of) all other objects. Thus, there is universal quantification over objects in the domain implicit in descriptions acting as principles.

An example of a more complex description is (19). The description is satisfied, e.g., by the feature structure in (16).[38]

[37] The description (18g) could only be false of 'Máňa' whose number is plural, which is not possible due to the definition of the interpretation function F in Fig. 3.2.3.

[38] Given the restricted signature, the description is quite redundant. As seen from ((18b)), a much simpler description is sufficient to pick up precisely the object in (16).

(19) $: \sim \textit{verb2} \wedge$
$:\text{LEXEME} \sim \textit{jít} \wedge$
$:\text{TENSE} \sim \textit{anterior} \wedge$
$:\text{ACTOR} \sim \textit{noun} \wedge$
$:\text{ACTOR LEXEME} \sim \textit{Máňa} \wedge$
$:\text{ACTOR NUMBER} \sim \textit{singular} \wedge$
$:\text{PATIENT} \sim \textit{verb1} \wedge$
$:\text{PATIENT ACTOR} \approx :\text{ACTOR} \wedge$
$:\text{PATIENT LEXEME} \sim \textit{tancovat} \wedge$
$:\text{PATIENT TENSE} \sim \textit{posterior}$

Definition 6 *A* **theory** *is a set of descriptions such that the denotation of a theory θ, $\Theta_I(\theta)$, is the intersection of the denotations of the descriptions in θ:*
for each $\theta \subseteq \mathsf{D}_{\boldsymbol{\Sigma}}, \Theta_I(\theta) = \{\upsilon \in U \mid \textit{for each } \delta \in \theta, \upsilon \in D_I(\delta)\}$.

The complex single description in (19) can be reformulated as a theory when all its conjuncts are treated as members of a set, constituting the theory.

A theory is true at an object iff each of its descriptions is. A theory of an object is the set of all descriptions true at the object. An interpretation is called a **model** of a theory iff its denotation relative to that interpretation is the whole U, in other words, a model of a theory is an interpretation such that the theory is true at every object. Finally, a **grammar** is a pair $\langle \boldsymbol{\Sigma}, \theta \rangle$, where $\boldsymbol{\Sigma}$ is a signature and $\theta \subseteq \mathsf{D}_{\boldsymbol{\Sigma}}$.

3.2.4 Relational Speciate Re-entrant Logic

Although King's SRL was created with the goal of providing an adequate formalism for HPSG grammars, there are still some points in which it falls short of this task. These shortcomings can be seen, e.g., when the grammar principles of Pollard and Sag (1994) are to be formalized in SRL. Richter (2000, p. 134) characterizes the problems as those of 'missing concepts'. The two most prominent ones are relations and quantification.

Throughout HPSG literature, **relations** are heavily used as a convenient and indispensible way of formalizing complex relationships, often with recursive structures. However, SRL does not offer such a concept and the suggested encoding of relations by means of junk slots suffers from serious defects. Due to many additional entities and attributes without any linguistic significance, the junk slot method is cumbersome and conceptually inadequate. Furthermore, it has been shown as unable to replicate the intended interpretation of relations under negations (see ibid.). Since relations are often used in such environments, not only junk slots, but also definite clause extensions are ruled out as a way of encoding relations.

A number of prominent principles of Pollard and Sag (1994) employ both existential and universal **quantification** over components of a certain entity. A good example is the principles of their Control Theory, p. 401. In order to express such principles in a natural way, bounded quantification must be available in the formalism.

With relations and quantification, **variables** represent another necessary extension.

The fact that standard HPSG literature employs certain concepts which have defied adequate and parsimonious formalization may signify that perhaps HPSG grammarians are misguided in some foundational theoretical issues. An alternative explanation may be that the nature of linguistic phenomena does not lend itself to generalizations which are at the same time theoretically meaningful and expressible in a formally restricted way. By this reasoning, any linguistic theory must eventually pay for greater expressivity by a less restrictive formalism. This is the position reflected in the adoption of RSRL as the formalism for the present enterprise. In fact, it will be shown that the new constructs introduced in RSRL are at least very useful, if not indispensable, for building a grammar with a different theoretical background

Richter (2000, p. 152) views bounded quantification over components of linguistic entities (rather than over all entities in the entire linguistic universe) as the most important innovation of RSRL. It is due to this extension that principles of Pollard and Sag (1994) can be expressed in a natural way. Due to the introduction of quantification, symbols of relations stand for relations between components of linguistic entities. Negation of relations can therefore be interpreted as in classical logic.

In spite of these important additions, the meaning of RSRL grammars can be characterized in the same ways as that of SRL grammars. Therefore, I will provide only two definitions concerning RSRL signature and then only briefly mention other differences between SRL and RSRL.

Definition 7 *A* **signature** *is a septuple* $\Sigma = \langle \mathsf{G}, \sqsubseteq, \mathsf{S}, \mathsf{F}, \mathsf{A}, \mathsf{R}, \mathsf{AR} \rangle$*, where*

$\langle \mathsf{G}, \sqsubseteq \rangle$ *is a partial order, where each element of* G *is a* **sort** *and* $\langle \mathsf{G}, \sqsubseteq \rangle$ *is the* **sort hierarchy***;*

$\mathsf{S} = \{ \sigma \in \mathsf{G} \mid$ *for each* $\sigma' \in \mathsf{G}$*, if* $\sigma' \sqsubseteq \sigma$ *then* $\sigma = \sigma' \}$ *is a set, its members are* **species** *or* **maximally specific sorts** *in* Σ*;*

F *is a set of features, its members are called* **features** *or* **attributes** *in* Σ*;*

A *is a partial function from* $\mathsf{G} \times \mathsf{F}$ *to* G*, called* **appropriateness function** *in* Σ*,*

where for each $\sigma_1 \in \mathsf{G}$*, for each* $\sigma_2 \in \mathsf{G}$ *and for each* $\varphi \in \mathsf{F}$*,*

if $\mathsf{A}(\sigma_1, \varphi)$ *is defined and* $\sigma_2 \sqsubseteq \sigma_1$

then $\mathsf{A}(\sigma_2, \varphi)$ *is defined and* $\mathsf{A}(\sigma_2, \varphi) \sqsubseteq \mathsf{A}(\sigma_1, \varphi)$*,*

R *is a finite set of* **relation symbols**

AR *is a total function from* R *to* N^+*, the* **arity function**

An RSRL signature includes an explicit sort (or type) hierarchy, which brings it closer to the linguistic practice. The definition of the appropriateness function reflects assumptions on feature inheritance: if a feature (attribute) φ is appropriate to a type (sort) σ_1, then it is appropriate to any type σ_2 which is more specific than σ_1, and $\mathsf{A}(\sigma_2, \varphi)$ is at least as specific as $\mathsf{A}(\sigma_1, \varphi)$. The arity function specifies the number of arguments for each relation symbol as a positive integer. Although nothing in the definition prevents infinite sets, actual grammars use only finite signatures.

In addition to the operator symbols of SRL, RSRL assumes other reserved symbols: $\exists, and\ \forall$ for quantifiers, and symbols related to *chains*. A chain is used instead of a list, whenever an argument of a relation standing for it does not correspond to a component of any of objects being described. This is an ontological problem for RSRL, which is solved by introducing chains as 'virtual lists' and 'quasi-sorts'. Only members of chains are components of the objects described, unlike the chains themselves. Variable symbols are also added as a countable infinite set.

In the following, interpretation of the signature is defined. Strong resemblance to the parallel definition for SRL is obvious.

Definition 8 *An* **interpretation** *of* Σ *is a quadruple* $I = \langle U, S, F, R \rangle$*, where*

U *is a set, called* **universe** *in* I*, each member of* U *being an* **entity** *in* I*;*

S *is a total function from* U *to* S*, called* **species assignment function** *in* I*;*

F *is a total function from* F *to the set of partial functions from* U *to* U*, called* **feature interpretation function** *in* I*, and*

for each $\varphi \in \mathsf{F}$*, for each* $v \in U$*,*

if $F(\varphi)(v)$ *is defined*
then $\mathsf{A}(S(v),\varphi)$ *is defined, and* $S(F(\varphi)(v)) \sqsubseteq \mathsf{A}(S(v),\varphi)$, *and*

if $\mathsf{A}(S(v),\varphi)$ *is defined*
then $F(\varphi)(v)$ *is defined;*

R *is a total function from* R *to the power set of* $\bigcup_{n \in N} \overline{U}^n$, *and*

for each $\rho \in \mathsf{R}$, $R(\rho) \subseteq \overline{U}^{\mathsf{AR}(\rho)}$

While the species assignment function S assigns every entity in the domain a species, a type σ denotes the union of the sets of entities denoted by the species of which σ is a supersort. The feature interpretation function F respects the appropriateness conditions stipulated by the appropriateness condition A. The relation interpretation function R interprets each relation symbol as a set of tuples of the form $\langle u_1, ..., u_n \rangle$, where n is the relation's arity and each u_i can be an entity or a sequence of entities in U.[39]

Terms in RSRL consist of either the reserved symbol ':' or a variable followed by a (possibly empty) string of attributes. Terms are the building blocks of RSRL formulae. There is a terminological difference between *descriptions* and *formulae*: a formula may contain free variables, unbound by quantifiers. Descriptions, a subset of formulae, may not contain free variables. A theory of grammar consists only of descriptions. In addition to all syntactic constructs of SRL descriptions, formulae allow for relations and quantification.

As in SRL, conjunction, disjunction, implication and equivalence are interpreted as in classical logic, including negation, which corresponds to set complement. Quantifiers are interpreted as applying to the set of components and chains of components of a member of U. The interpretation of type assignments $\tau \sim \sigma$ and path equations $\tau_1 \approx \tau_2$, where τ, τ_1 and τ_2 all start with the colon, does not differ from the way the expressions are interpreted in SRL. If, however, τ starts with a variable and the function denoted by the string of attributes following the variable is defined on the entity assigned to the variable and leads to an entity denoted by σ, then $\tau \sim \sigma$ denotes the entire universe. Similarly, $\tau_1 \approx \tau_2$ denotes the entire universe if the functions denoted by the strings of attributes are defined on the entities assigned to the variables and lead to the same entity. Otherwise, both expressions denote an empty set.

3.2.5 Attribute-value matrices and feature declarations

The definitions of Relational Speciate Re-entrant Logic make explicit all necessary assumptions about a formal language used for describing linguistic

[39]The symbol $\overline{U}$ stands for the set U unioned with the set of finite strings over U, including the empty set.

objects. However, to use the verbose (R)SRL notation for grammar writing is not very practical. Constraint-based grammars are usually spelled out in a closely related, yet more compact notation, namely in attribute-value matrices (AVMs). An example AVM has already been presented in (17) and it is related to the equivalent SRL description in (19) in the obvious way. In fact, using appropriate logical symbols, any description and therefore any theory in the (R)SRL sense can be expressed in the AVM notation.

In contrast to the linear notation of (R)SRL, the AVM notation is two-dimensional, based on matrices of attributes and types, enclosed in square brackets. Each matrix must be of a certain type. Matrices can be nested: an attribute value can be a matrix. The comparison of the SRL description (19) above with its AVM counterpart (17), repeated for convenience here as (20), shows the advantages of the AVM notation.

(20)
$$
\begin{bmatrix} \textit{verb2} & \\ \text{LEXEME} & \textit{jít} \\ \text{TENSE} & \textit{anterior} \\ \text{ACTOR} & \boxed{1} \begin{bmatrix} \textit{noun} & \\ \text{LEXEME} & \textit{Máňa} \\ \text{NUMBER} & \textit{singular} \end{bmatrix} \\ \text{PATIENT} & \begin{bmatrix} \textit{verb1} & \\ \text{LEXEME} & \textit{tancovat} \\ \text{TENSE} & \textit{posterior} \\ \text{ACTOR} & \boxed{1} \end{bmatrix} \end{bmatrix}
$$

While in the linear notation of (R)SRL there is a significant amount of redundancy in the repetition of attribute names, the two-dimensional AVM notation is more compact. This is because (i) there are no parallel stretches in the paths of attributes (cf. :ACTOR LEXEME and :ACTOR NUMBER in (19)) – more deeply embedded attributes are elements of an AVM, introduced as a whole by the name of a less deeply embedded attribute, and (ii) shared values of two or more paths are indicated within the specification of path values by indices, depicted as boxed numbers (tags). Additionally, logical connectives, quantifiers, relations and variables can be used in the AVM notation in the same way as in (R)SRL.[40]

Now what about signature? In HPSG, the signature part of grammar is expressed as a *sort hierarchy* together with *feature declarations*. Sort hierarchy is a finite partial order of *sorts* with a distinguished sort, which is a supersort of all other sorts and which is taken to denote all linguistic objects.[41] According to Pollard and Sag (1994), the denotations of immediate subsorts of a nonmaximal (nonterminal) sort partition the denotation of

[40]The boxed numbers (tags) are also variable symbols.

[41]The distinction between sorts and types is irrelevant here.

that sort. Therefore, each linguistic object is in the denotation of one and only one maximal (terminal) sort. Additionally, for each maximal sort and feature, the denotation of the feature can be defined only on all objects in the denotation of the sort, or on no objects in the denotation of the sort. As a result, each maximal sort is a (R)SRL species, and each nonmaximal sort is the disjunction of its maximal subsorts in SRL. An RSRL grammar includes sort hierarchy, so it is not necessary to make additional assumptions about the relation of non-maximal HPSG sorts to species. Each sort feature is an (R)SRL feature and feature declarations for maximal sorts can be expressed by the SRL appropriateness function A: for each maximal sort σ, for each feature φ, if σ bears feature declaration $[\sigma\varsigma]$, where ς is a sort, then $\mathsf{A}(\sigma,\varphi)$ is the set of maximal subsorts of ς else $\mathsf{A}(\sigma,\varphi)=\emptyset$. In SRL, feature declarations for the nonmaximal sorts are inferable from A. In RSRL, the appropriateness function is specified for non-maximal sorts as well.

In a sort hierarchy, as a partial subsumption order, (R)SRL species are assumed to be maximal sorts (maximally specific, most informative items) at the bottom. In the trivial case of a flat sort hierarchy, only one additional item must be present, namely the top element (written as $\top$ or *top*), the most general and least informative item. However, a common sort hierarchy usually abounds in other non-maximal sorts. The primary reason is that the non-maximal sorts express linguistically important intuitions about the categories of objects in the domain, where a mere inventory of species is not sufficient. The denotation of a non-maximal sort can be construed as a set of entities denoted by all maximal sorts subsumed by the non-maximal sort. The non-maximal sorts are typically interpreted as linguistic categories and used in grammar statements, constraining all their subsorts. Non-maximal sorts can thus be used to express generalizations across species and one type of such generalization is usually present in the very specification of a subsumption hierarchy: the (R)SRL appropriateness function is translated into "feature declarations"; a feature declaration is applied to the sort subsuming all other sorts for which a feature and its value is appropriate. Our example signature (Fig. 3.3) can be translated into a sort hierarchy with feature declarations as in Fig. 3.5 on page 75. Instead of the graphical notation, the sort hierarchy can be expressed in an indented text format, as in Fig. 3.6.

In the indented text format, sort *b* is indented below sort *a* just in case sort *b* is subsumed by sort *a*. Attributes and other sorts as values of the attributes follow the sorts for which their are appropriate. Subsorts are not followed by attributes and attribute values when those are inherited from a supersort. Thus, for *verb2* not only the attribute PATIENT is appropriate, but also the attributes ACTOR, TENSE and LEXEME.[42]

Given these equivalences (i) between (R)SRL descriptions/theories and

[42] Among other verb classes, we are ignoring verbs without subject (and actor), such as *pršet* 'to rain', which would not fit under the sort *verb*.

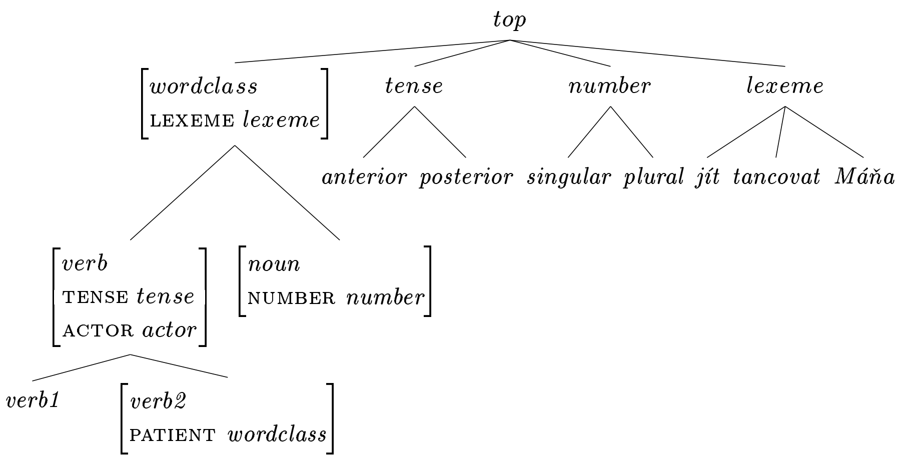


Figure 3.5: Example of a sort hierarchy in a graphical format

top
- *wordclass* LEXEME *lexeme*
  - *verb* TENSE *tense*, ACTOR *wordclass*
    - *verb1*
    - *verb2* PATIENT *wordclass*
  - *noun* NUMBER *number*
- *tense*
  - *anterior*
  - *posterior*
- *number*
  - *singular*
  - *plural*
- *lexeme*
  - *jít*
  - *tancovat*
  - *Máňa*

Figure 3.6: Example of a sort hierarchy in a text format

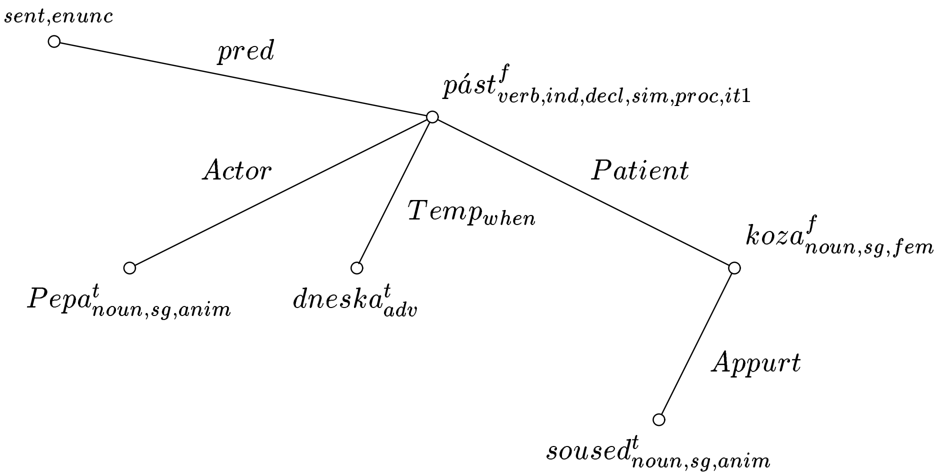


Figure 3.7: Another TR as a tree

HPSG-like principles expressed in the AVM notation, and (ii) between (R)SRL signature and sort hierarchy with feature declarations, an HPSG-like grammar can be expressed as an RSRL grammar.

3.2.6 A more complex example and abbreviatory conventions

The example dependency tree in Fig. 3.2 and its recast as a feature structure in (16) was a rather naive exercise serving mainly for expository purposes. The following example (21) adds an adverbial and one more level of syntactic embedding. Its tectogrammatical representation in Fig. 3.7 on p. 76 features a more complete inventory of grammatemes and the distinction between contextually bound and non-bound semantemes. A nonterminal node is added to represent the sentence as a whole.

(21) Pepa dneska pase sousedovu kozu
Pepa-NOM today graze-PRES-3RD-SG neighbour-POSS goat-ACC
'Today Pepa is grazing the neighbour's goat'

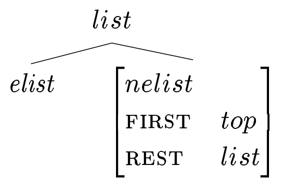
The mapping of the tree in Fig. 3.7 to a feature structure should retain all information, including the horizontal order of nodes. Additionally, it should allow for integration into a structure and grammar specifying the relation between the string of graphemes/phonemes and its tectogrammatical representation. A possible solution is presented in Fig. 3.8 on p. 77.

A number of symbols used in Fig. 3.8 have not yet been introduced. Firstly, there are two new pairs of symbols: angle brackets ⟨ ⟩ and curly brackets { }. Both may be viewed as a kind of 'syntactic sugar'. Angle brackets are used to enclose items in a list and are interpreted as a sort *list*, a supersort of species *elist* (for empty list) and *nelist* (for non-empty list).

$$
\left[\begin{array}{ll}
\textit{top} & \\
\text{ORDER} & \boxed{1} \\
\text{GOV} & \left[\begin{array}{ll} \textit{sent\_pred} & \\ \text{SENTMOD} & \textit{enunc} \end{array}\right] \\
\text{DEPS} & \left\{ \left[\begin{array}{ll}
\textit{pred} & \\
\text{ORDER} & \boxed{1}\ \texttt{concatenate}(\langle\boxed{3}\rangle, \langle\boxed{4}\rangle, \langle\boxed{2}\rangle, \boxed{5}) \\
\text{GOV} & \boxed{2} \left[\begin{array}{ll}
\textit{verb} & \\
\text{CB} & \textit{no} \\
\text{LEX} & \textit{pást} \\
\text{VERBMOD} & \textit{ind} \\
\text{DEONTMOD} & \textit{decl} \\
\text{TENSE} & \textit{sim} \\
\text{ASPECT} & \textit{proc} \\
\text{ITER} & \textit{it1}
\end{array}\right] \\
\text{DEPS} & \left\{ \begin{array}{l}
\left[\begin{array}{ll}
\textit{actor} & \\
\text{ORDER} & \langle\boxed{3}\rangle \\
\text{GOV} & \boxed{3} \left[\begin{array}{ll}
\textit{noun} & \\
\text{CB} & \textit{yes} \\
\text{LEX} & \textit{Pepa} \\
\text{NUMBER} & \textit{sg} \\
\text{GENDER} & \textit{anim}
\end{array}\right] \\
\text{DEPS} & \{\}
\end{array}\right], \\
\left[\begin{array}{ll}
\textit{temp\_when} & \\
\text{ORDER} & \langle\boxed{4}\rangle \\
\text{GOV} & \boxed{4} \left[\begin{array}{ll}
\textit{adv} & \\
\text{CB} & \textit{yes} \\
\text{LEX} & \textit{dneska}
\end{array}\right] \\
\text{DEPS} & \{\}
\end{array}\right], \\
\left[\begin{array}{ll}
\textit{patient} & \\
\text{ORDER} & \boxed{5}\ \texttt{concatenate}(\langle\boxed{7}\rangle, \langle\boxed{6}\rangle) \\
\text{GOV} & \boxed{6} \left[\begin{array}{ll}
\textit{noun} & \\
\text{CB} & \textit{no} \\
\text{LEX} & \textit{koza} \\
\text{NUMBER} & \textit{sg} \\
\text{GENDER} & \textit{fem}
\end{array}\right] \\
\text{DEPS} & \left\{ \left[\begin{array}{ll}
\textit{appurt} & \\
\text{ORDER} & \langle\boxed{7}\rangle \\
\text{GOV} & \boxed{7} \left[\begin{array}{ll}
\textit{noun} & \\
\text{CB} & \textit{yes} \\
\text{LEX} & \textit{soused} \\
\text{NUMBER} & \textit{sg} \\
\text{GENDER} & \textit{anim}
\end{array}\right] \\
\text{DEPS} & \{\}
\end{array}\right] \right\}
\end{array}\right]
\end{array} \right\}
\end{array}\right] \right\}
\end{array}\right]
$$

Figure 3.8: Another TR in the AVM format

For the latter species, two features are appropriate: FIRST (whose value may be any species) and REST (whose value is the sort *list*). The part of sort hierarchy defining the sort *list* and its two subsorts is shown in (22).

(22)


Curly brackets enclose a set. See Richter (2000, §4.4) for a detailed suggestion on how sets are formalized in RSRL.

These two additional data types are needed to deal with some subparts of the structure whose number cannot be fixed or whose status, otherwise encoded in a feature name, is not known. A list is used where the order of items matters. Otherwise, a set should be used. In the present example, the number and status of some types of dependents – namely optional free modifications (adjuncts) – is not determined by the lexical head. Therefore, the vertical dimension of the dependency tree is represented by feature GOV for the governor and a set-valued feature DEPS for its dependents. In order to represent the horizontal dimension, a list-valued feature ORDER is used for every subtree. This feature scopes both over the governor and its dependents, so it is appropriate for the same kinds of species as GOV and DEPS, namely for species encoding syntactic (tectogrammatical) function of the subtree (or the subtree's governor).

Secondly, values of some features are expressed as functions, which evaluate to the appropriate species. Such functions are in fact relations in functional notation and can be expressed by relations including an additional argument, which is identified with the value of the relevant feature. The relation `concatenate` (in its functional notation) is used to express the fact that its value is a list consisting of members of the argument lists, concatenated in the order of the arguments.

Richter (1999) or Richter (2000) provides a detailed formal description of the mapping between the AVM and RSRL notations, together with abbreviatory conventions. The mapping and conventions make explicit all steps from a full RSRL description to a compact AVM format, as commonly used in HPSG literature. The two are equivalent notational variants. After the translation of RSRL expressions into AVM matrices, the following steps can be taken in order to make the structure more compact:

1. The colon as the tag of a top matrix may be left out.

2. At the top of a matrix, the type symbol may be left out. If it is missing, the least specific symbol is assumed.

3. Brackets may be omitted in a matrix which is not the top matrix and contains only one attribute and no type. The brackets of any matrix may also be omitted if it contains only the type symbol.

4. If a tag (i.e., a variable) is not explicitly bound by a quantifier, it is implicitly bound by existential quantification scoping over the entire formula.

5. The string $\langle a|b\rangle$ is equivalent to the AVM $\begin{bmatrix} \textit{list} \\ \textsc{first}\ a \\ \textsc{rest}\ b \end{bmatrix}$.

6. A matrix may occur as an argument of a relation.

7. A relation may be defined in a way resembling Prolog using a new 'clause symbol' $\stackrel{\forall}{\Longleftarrow}$, where each disjunct in the consequent of the clause may give rise to one 'pseudo-clause'.[43] The universal quantifier which is part of the clause symbol represents universal quantification over the variables to its left, in the head of the definition. All variables following $\stackrel{\forall}{\Longleftarrow}$ in the consequent of the pseudo-clause are regarded as existentially bound with the quantifiers scoping over the consequent.

8. Sets can be represented as enclosed in curly brackets with the usual set operations symbols as notational variants of the corresponding relations.

The examples below show descriptions in the AVM format with the abbreviatory conventions applied, and in the RSRL format.

(23) TREE AS LISTS PRINCIPLE (TLP) in the AVM notation – see (109)

$$\begin{bmatrix} \textit{deep} & \\ \textsc{tree} & \boxed{1} \end{bmatrix} \rightarrow \texttt{d\_list}(\boxed{1})$$

(24) TLP in RSRL

$$\exists x\ [\ [\ :\sim \textit{deep} \wedge\ :\textsc{tree} \approx x\ ] \rightarrow\ \texttt{d\_list}(x)\ ]$$

[43]For some examples, see §A.3

(25) At least one NB node Principle (ONBP) in the AVM notation – see (110)

$$\begin{bmatrix} \textit{deep} & \\ \text{STATUS} & \textit{unemb} \\ \text{TREE} & \boxed{1} \end{bmatrix} \rightarrow \texttt{nested\_member}([\text{CB } \textit{no}], \boxed{1})$$

(26) ONBP in RSRL

$$\begin{array}{l} \exists x \; [\; [: \sim \textit{deep} \;\wedge\; : \text{STATUS} \sim \textit{unemb} \;\wedge\; : \text{TREE} \approx x \;] \\ \quad \rightarrow \; \exists y \; [\; y\text{CB} \sim \textit{no} \;\wedge\; \texttt{nested\_member}(y, x) \;] \;] \end{array}$$

(27) Governor Position Principle (GPP) in the AVM notation – see (111):

$$\begin{bmatrix} \textit{deep} & \\ \text{TREE} & \left\langle \boxed{1} \oplus \langle \textit{d-node} \,|\, \boxed{2} \rangle \right\rangle \end{bmatrix} \rightarrow$$

$$\begin{array}{l} \forall_{\boxed{3}} \Big(\texttt{dependent}\,(\boxed{3}\textit{d-node}, \boxed{1}) \;\; \rightarrow \;\; \boxed{3}[\text{CB } \textit{yes}] \Big) \\ \wedge \left(\begin{array}{l} \forall_{\boxed{4}} \Big(\texttt{dependent}\,(\boxed{4}\textit{d-node}, \boxed{2}) \;\; \rightarrow \;\; \boxed{4}[\text{CB } \textit{no}] \Big) \\ \vee \;\; \forall_{\boxed{5}} \exists_{\boxed{6}} \exists_{\boxed{7}} \left(\begin{array}{l} \Big(\texttt{member}\,(\boxed{5}, \boxed{2}) \wedge \texttt{member}\,(\boxed{6}[\text{CB } \textit{yes}], \boxed{5}) \Big) \\ \quad \rightarrow \;\; \texttt{nested\_member}\,(\boxed{7}[\text{CB } \textit{no}], \boxed{5}) \end{array} \right) \end{array} \right) \end{array}$$

(28) GPP in RSRL

$$\begin{array}{l} \exists x_1 \; \exists x_2 \; \exists y_1 \; \exists y_2 \; [\; [\\ \quad : \sim \textit{deep} \;\wedge\; :\text{TREE} \approx y_1 \\ \quad \wedge \; \texttt{append}(x_1, y_2, y_1) \;\wedge\; y_2\text{FIRST} \sim \textit{d-node} \;\wedge\; y_2\text{REST} \approx x_2 \;] \\ \quad \rightarrow \\ \quad [\\ \quad \forall x_3 \; [\; x_3 \sim \textit{d-node} \;\wedge\; \texttt{dependent}(x_3, x_1) \;\rightarrow\; x_3\text{CB} \sim \textit{yes} \;] \\ \quad \wedge \; [\; \forall x_4 \; [\; x_4 \sim \textit{d-node} \;\wedge\; \texttt{dependent}(x_4, x_2) \;\rightarrow\; x_4\text{CB} \sim \textit{yes} \;] \\ \qquad \vee \; \forall x_5 \; \exists x_6 \; \exists x_7 \; [\\ \qquad\quad \texttt{member}(x_5, x_2) \;\wedge\; x_6\text{CB} \sim \textit{yes} \;\wedge\; \texttt{member}(x_6, x_5) \\ \qquad\quad \rightarrow \; x_7\text{CB} \sim \textit{no} \;\wedge\; \texttt{nested\_member}(x_7, x_5) \;] \;] \;] \;] \end{array}$$

(29) member/2 in RSRL – see §A.3.17

$$\begin{array}{l} \forall x \, \forall y \, [\, \texttt{member}(x, y) \leftrightarrow \\ \quad [\; x \approx y\text{FIRST} \vee \\ \quad \exists z \, [\; z \approx y\text{REST} \wedge \texttt{member}(x, z) \;] \;] \;] \end{array}$$

(30) `member/2` in RSRL's Prolog-like notation

$$\texttt{member}(x,y) \overset{\forall}{\Longleftarrow}$$
$$y\langle x|\boxed{0}\rangle$$
$$\texttt{member}(x,y) \overset{\forall}{\Longleftarrow}$$
$$y\langle \boxed{0}|\boxed{1}\rangle \wedge \texttt{member}(x,\boxed{1})$$

(31) Equivalent of `member/2` in Prolog

```
member(X, [X|_]).
member(X, [_|Z]) :- member(X, Z).
```

3.2.7 Representation and grammar

Table 3.1 shows how the components of the formal language relate to the conventional picture of the domain of linguistics, consisting of grammars of various degrees of generality, lexicons, and the subject of linguistic enquiry: strings of words and their analyses (representations).

	Components of the formal language	**Linguistic interpretation**
A.	Model	Language expressions, their representations, linguistic categories
(1)	Feature structures of type *sign* (or equivalent), its subparts:[44]	Specific expressions with their representations
(2)	Lists of objects of type *phon-string* (or equivalent)	Specific expressions (terminal strings of phonemes/graphemes): morphemes, word forms, phrases / syntagms, sentences
(3)	Objects of type *deep* (or equivalent)	(Deep) syntactic representation of a specific expression
(4)	Any other object	Linguistic category (number, a node in a structure)
B.	Grammar	
(1)	Signature (sort hierarchy with appropriate features)	Definition of linguistic categories (including construction types), their properties and mutual relations, lexical items
(2)	Theory (a set of descriptions / constraints composed from terms)	Universal and specific principles, "grammar rules", inflection and derivation regularities

Table 3.1: Components of the formal language and their interpretation

[44] An entity of type *sign* is meant to be an entity which can be represented at all levels of the theory. Typical representatives would be words and higher syntactic units.

$$\begin{bmatrix} \textit{sign} & \\ \text{CONTENT} & \textit{content} \\ \text{DEEP} & \begin{bmatrix}\textit{deep}\\ \text{DEEP-ORDER } \boxed{1}\end{bmatrix} \\ \text{SURFACE} & \begin{bmatrix}\textit{surface}\\ \text{SURFACE-ORDER } \boxed{2}\end{bmatrix}\end{bmatrix} \wedge \texttt{relate-orders}\ (\boxed{1}, \boxed{2})$$

Figure 3.9: Relation between surface and deep orders.

3.2.8 Levels as properties of a single object

In a constraint-based framework, multiple levels of linguistic description are commonly represented in parallel as parts of a single object. This single object may model an entity of various kinds: morpheme, word, multi-word expression, phrase or syntagm, clause, sentence or even higher units of discourse. Any such entity can thus be represented along several dimensions corresponding to the levels of description. In this sense, the constraint-based approach is 'fractal'. In a schematic AVM in Fig. 3.9, a possible structure representing an entity of type *sign* is shown. The value of feature DEEP, type *deep*, is an object such as that in (17) or Fig. 3.8, i.e., its tectogrammatical representation. The value of feature SURFACE, type *surface*, includes morphological and possibly surface-syntactic characteristics of the object, including its syntactic combinatory potential expressed as valency requirements. There can be additional levels, such as a layer for semantic interpretation, which is treated as the value of feature CONTENT.

In multistratal frameworks, the often complex relationships between the levels are solved by transformations. In a constraint-based setting, where linguistic entities are represented as multidimensional objects with levels in parallel, this option is not available. The primary reason is that transformations involve nonmonotonicity, which is in conflict with the declarative nature of constraint-based architecture. However, the unavailability of transformations may well be an advantage.

The principal tools which provide an alternative to transformations are structure sharing and relations. The relation `relate-orders` in Fig. 3.9, relating the deep and surface orders, is an example of this. Of course, to define such a relation for any language is a huge task. In fact, the rest of this work will be concerned with such a task.

3.3 Implementation

The formal language of RSRL was created with the aim to reflect faithfully the formal properties of HPSG. Given that HPSG has been often presented as a theory which makes it relatively easy to write computationally tractable grammars, an RSRL grammar should not be difficult to implement. However, when Richter (2000, §3.3) examines the computational properties of RSRL, the conclusion is rather different. RSRL fails in all the three criteria: satisfiability, modelability and prediction.

A theory is satisfiable iff, given a signature, there exists an entity with possibly other entities as its components in an interpretation of the theory that satisfies every description in the theory. Satisfiability indicates whether a theory is consistent. If there is an algorithm which can find out whether the theory is satisfiable or not, satisfiability is decidable. Satisfiability of RSRL has been proven undecidable. This means that there is no sound and complete calculus which could check satisfiability of an RSRL theory.

A theory is modelable iff, given a signature, there is an interpretation of the theory in which every entity satisfies every description of the theory. Obviously, modelability is a stronger property than satisfiability and one which is again undecidable for RSRL.

Finally, there is the property of prediction, which is related to the problem of parsing. Generally speaking, a grammar predicts the existence of a description just in case there is an interpretation of the grammar in which the description is satisfied. In parsing, we are interested in the grammaticality of a string given a grammar. Because in constraint-based grammars the string is a component of an object, parsing is a problem of finding whether a description including the string is a description of an object according to the grammar. Again, prediction in RSRL is undecidable.

These results suggest that it is difficult, if not impossible, to formulate a linguistic theory which, at the same time, covers non-trivial phenomena, expresses generalizations in a natural way, uses a well-defined formal language, and is easy to implement. This difficulty may be due to the fact that human linguistic competence is in real life supplemented by processing strategies involving huge amounts of knowledge about the world and about the situation (discourse) being talked about, and also some measure of probabilistic guessing. In this light, the fact that it is difficult to process language using only competence grammar does not speak against such a grammar. On the contrary, it is important to investigate the formal properties of *langue* independently of the way it is consulted in language activities.

Such investigations may bear fruit which will be appreciated on the practical level. A grammar which is formulated in a well-defined way may be transformed by an algorithm into an equivalent grammar with the desirable computational properties. Such a grammar may also be supplemented by a processing strategy based on knowledge representation, heuristic or probabilistic techniques. Indeed, probabilistic techniques are a good solution not only for cases when humans also guess, but also when the necessary knowledge cannot be compiled and the human way of processing must be simulated.

The scepticism concerning computational properties of RSRL is justified in general. The situation may be different for specific grammars, which may be computationally more restrictive. Furthermore, in many cases, RSRL descriptions can be restated in a more restrictive way, even if it means some loss of correspondence with the theoretical background. Given a specific computational system, such as ConTroll, ALE, or LKB,[45] most, if not all, descriptions of a specific RSRL grammar can be rephrased as statements of a computational formalism, even if it means using techniques which may obscure their linguistic content or change some premises of the linguistic theory.

The impossibility to verify an RSRL grammar computationally in a mechanical fashion is to be regretted. Even though my aim here was different, at least some parts of the system could and should be implemented. I will leave that issue for further research.

3.4 Conclusions

In this lengthy chapter I have provided theoretical and formal foundations for the description of some real linguistic phenomena.

After making clear what ontological status of the description is being assumed, I have presented main characteristics of FGD, considering the possibility of embedding the theory in a constraint-based formalism. No case was found where such a possibility would be excluded.

Then I introduced the constraint-formalism by providing an example where a dependency tree was translated into a feature structure (Attribute-Value Matrix). Next, I presented the syntax and semantics of SRL, a formal language developed originally for HPSG, and its extension, RSRL, together with the AVM format and abbreviatory conventions.

Finally, some optimistic comments were made on the margin of unsatisfactory computational properties of RSRL.

[45]See L. Bolc and K. Czuba and A. Kupść (1996) for an overview of systems for implementing HPSG grammars.

Chapter 4

Focus on facts

This chapter is concerned with linguistic rather than formal issues: the features of FGD described formally in the subsequent parts are specified in detail, and a series of notes on specific linguistic issues is made. One of them is really an 'extended' note: starting from the word order principles of Vilém Mathesius, a classification of word-order phenomena is developed. This classification will be employed later in the proposed formal account of (some aspects of) word order in Czech.

4.1 A checklist of desiderata

Here is the summary of those features of the standard version of FGD which are addressed in the present work:

1. Language expressions are described simultaneously on several mutually interacting **levels**. There are two crucial levels in the system: the level of morphemics (where expressions are represented as strings of morphemes) and the tectogrammatical level (the interface level on the border between language proper and the domain of cognition).

   The following points are concerned with the tectogrammatical level.[1]

[1] In the examples below I will use a shorthand notation for TR, inspired by that used in Kruijff-Korbayová (1998), and abbreviatory conventions for values adopted in Hajičová, Panevová, and Sgall (2000). In (37), repeated here as (32), the underlined part of the sentence belongs to focus (with respect to the global TFA of the topmost tree), while SMALL CAPITALS denote its intonation centre:

(32) Tom admires her INTELLIGENCE.
[[ACT:Tom^{cb}] admire^{nb} [[APP:she^{cb}] PAT:intelligence^{nb}]]

In the linearized representation of the tree the performative node is omitted, the brackets delimit subtrees, the labels in small capitals preceding every node except the root stand for tectogrammatical functions, the superscripts mark the nodes as context-bound ^{cb} or non-bound ^{nb}. The underlining again denotes the global focus.

2. Language expressions are represented as **dependency trees**.[2] A dependency tree consists of edges and nodes. The tree has three dimensions along which the nodes can be ordered. The vertical dimension of ‘height’ expresses syntactic dependency, the horizontal dimension of ‘breadth’ expresses deep word order, and the dimension of ‘depth’ coordination and apposition.[3]

3. In order to accommodate information relevant to the utterance as a whole, the standard FGD notation employs a distinguished top node as the governor of the main clause (or the topmost syntagm), the only non-terminal node in the tectogrammatical tree. This **performative node** includes “the four chief referential indices (of speaker, hearer(s), place and time of utterance); further, illocutionary force, insofar as the latter is structured by the language system” – i.e., declarative, imperative, or interrogative (Sgall, Hajičová, and Panevová, 1986, p. 154).

4. Other nodes represent *semantemes* and correspond to **content words**. The edges correspond to various types of **dependency relations** – ‘functors’ (except for the edge leading from the top node). Equivalently, functors can label the dependent node instead of the corresponding edge.

Occasionally, contrastive stress is marked by *italics*, as in (51), repeated here as (33). In the linearized TR notation, a semanteme bearing contrastive stress is labelled by the superscript *c*.

(33) *Him* she TOLD.
[[ADDR:he^{c}] [ACT:she^{cb}] [PAT:_^{cb}] tell^{nb}]

An empty node has an underscore in place of a lexeme, possibly coindexed by a subscript number to express coreference with another semanteme, as in (34).

(34) She told him to STAY.
[tell^{cb} [ACT:she^{cb}] [ADDR:he$_{1}^{cb}$] [[ACT:_$_{1}^{cb}$] PAT:stay^{nb}]]

It is often the case that there is more than one possible TR corresponding to a given surface string, especially if TFA is taken into account. This is not explicitly mentioned in most examples when a single option is presented.

[2] I will use the term *tree* for the representation of the whole sentence or utterance and the term *local tree* for any subtree of depth 1.

Speaking about trees, I will use the terms *depending* and *governing* to identify relations between a governor and its immediate dependent(s), and the terms *subordinated* and *superordinated* to identify their respective transitive closure. The term *tree* covers the root of the whole structure and all of its *subordinated* nodes, *local tree* a node and all of its *depending* nodes. The term *subtree* covers a node and all of its *subordinated* nodes, unless its depth is specified, as in the explanation of the term *local tree* above.

[3] For technical reasons, the third dimension can be eliminated, coordination and apposition being handled as a kind of syntactic dependency, as in Hajičová, Panevová, and Sgall (2000).

5. There are nodes at TR which do not correspond to any content words present in the surface string. These are cases of **surface deletion**.

6. Information other than that expressed by the geometry of nodes and edges is represented as **features of nodes**: lemma, indication of contextual boundness, and morphological grammatemes (tectogrammatical equivalents of morphological categories). Functors can be further specified by syntactic grammatemes.[4]

7. Nodes of every local tree are totally ordered from left to right; the order corresponds to the hierarchy of **communicative dynamism** (CD).[5] Less dynamic nodes horizontally precede more dynamic nodes.[6]

8. If every local tree is ordered under CD and the condition of projectivity is applied,[7] an order of nodes can be defined for the whole tree as the transitive closure of CD orders in local trees. The total order of all nodes in a tree, derived in this way, is called **deep** (or *underlying*) **word order** (DWO) (Hajičová, Partee, and Sgall, 1998, p. 73–4).

9. Viewed as discourse-related, every node in a tree except the top node is either **context-bound** (CB) or **non-bound** (NB).

10. There is **at least one NB node or grammateme** in the tree.

11. In every local tree, the **root precedes all NB nodes**.

12. In every local tree, either the **root follows all CB nodes**, or there are CB nodes immediately following the root node which satisfy at least one of the following conditions:[8]

[4]For a recent and exhaustive list of grammatemes and functors see Hajičová, Panevová, and Sgall (2000).

[5] The theoretical background and development of this concept is presented in Sgall, Hajičová, and Panevová (1986, p. 175ff). It was Firbas (1957), who extended Vilém Mathesius' dichotomy of Topic and Focus, based on the comparison of Czech and English (Mathesius, 1939; Mathesius, 1975), by the hierarchy (or scale) of communicative dynamism. A more recent presentation of his work can be found in Firbas (1992).

[6] In the current standard version of the theory, CD is only defined for local trees. CD ordering of nodes in the whole tree is partial (Hajičová, Partee, and Sgall, 1998, p. 73). Alternatively, CD could be defined for the whole tree, with the possibility to order nodes irrespectively of their placement in local trees. Of course, either the condition of tree projectivity must be lifted, or CD must be viewed separately from the tree as a distinct aspect of tectogrammatical representation while the tree nodes are horizontally unordered. In the following, I will assume that CD is only defined for local trees, unless specified otherwise.

[7]The condition of projectivity licenses non-tangling trees only. It requires for every local tree and for every pair of adjacent sister nodes n_1 and n_2 in that tree with n_1 horizontally preceding n_2 the following: n_1 and all nodes subordinate to n_1 must horizontally precede n_2 and all nodes subordinate to n_2.

[8]See Plátek, Sgall, and Sgall (1984, p. 72-73), Petkevič (1995b, p. 274), and Petkevič (in press). In (35b) the node for 'teacher' is CB, its dependent node, corresponding to 'of

(a) the node has at least one NB grammateme, or

(b) the node has a subordinated node which is either NB, or has at least one NB grammateme.[9]

13. An ellipsis — a **restored node** (deleted at surface) — is CB. This applies to cases of both syntactic and discourse ellipsis.

14. In every local tree, the (horizontal) order of dependent NB nodes is determined by **systemic ordering** (SO) of their tectogrammatical functions.[10]

15. For every local tree, all its nodes and all nodes subordinated to them belong either to **topic** or to **focus** with respect to the local tree. This partitioning is referred to as topic-focus articulation (TFA).[11]

16. **TFA is derived** from the property of contextual boundness in the following way. For the root of every subtree:

English', is NB.

(35) a. Which teacher did you meet?

b. I met the teacher of English.
[[ACT:I^{cb}] meet^{cb} [PAT:teacher^{cb} [RSTR:English^{nb}]]]

According to Plátek, Sgall, and Sgall (1984) and Petkevič (1995b) only a single CB node was allowed to follow the root. However, as seen in example (36b), a CB node with an NB dependent can be followed by a sister NB node.

(36) a. What did you present to which teacher?

b. I gave the teacher of English a box of chocolates.
[[ACT:I^{cb}] give^{cb} [ADDR:teacher^{cb} [RSTR:English^{nb}]]
[PAT:box^{nb} [RSTR:chocolate^{nb}]]]

[9] Applied recursively to more deeply embedded local trees, point(11) above together with the current point guarantee that the NB/NB-grammateme node is on the end of a right-branching path from the CB node.

[10] Until recently, FGD postulated a single total systemic ordering of functions for every language, irrespective of what kind of governor in a local tree is involved. Since then, the notion of systemic ordering has shifted towards a more lexically specific approach, where the prevailing number of lexemes still require their NB dependents to be ordered in a uniform pattern, but where, at the same time, there are classes of lexemes (and possibly individual lexemes) which require a specific 'systemic' or unmarked order of their NB dependents.

I will continue using the term 'systemic ordering' (SO) irrespective of whether there is a single language-specific order or a set of lexically specific orders. Either approach is compatible with the present formalization, see §6.2.5 on p. 169 below.

[11] TFA was studied by Vilém Mathesius (see footnote 5 above) and later, i.a., by Firbas (1992), Daneš (1974), Uhlířová (1972; 1976), Dušková (1986), and esp. by Eva Hajičová (see Hajičová (1993) for a concise presentation). TFA was found to be a clue to various issues ranging from prosody to semantics.

(a) if the root is CB, it belongs to topic; if it is NB, it belongs to focus;

(b) if any NB nodes depend on the root, they belong to focus together with all their subordinated nodes;[12]

(c) if there is at least one NB in the topmost local tree of the subtree[13] then any CB nodes dependent on the root belong to topic together with all their subordinated nodes;[14]

(d) if the root is CB and all nodes dependent on it are CB, then the focus consists either of the highest NB node(s) together with all their subordinate nodes, or of one or more NB grammatemes of a CB node, while such NB node(s) or such a CB node depend on one or more CB nodes right-dependent on the root (see point 12 above).

Being a property derived from the information on contextual boundness, the identification of topic and focus need not be explicitly stipulated in the representation.

17. The least communicatively dynamic item in topic is referred to as **topic proper**. The most communicatively dynamic item in focus is referred to as **focus proper**. In surface word order, topic proper can be subject to specific ordering.

18. Topic proper can become **contrastive topic**. Contrastive topic is usually marked by a distinct intonation pattern.

---

[12]It follows that a CB node can belong to focus if its governing node differs from the root and also belongs to focus, as the node corresponding to *she* in (37):

(37) Tom admires her INTELLIGENCE.
[[ACT:Tom^{cb}] admire^{nb} [[APP:she^{cb}] PAT:intelligence^{nb}]]

[13]This condition excludes the case when there are only CB nodes in the topmost local tree, as in (35b).

[14]It follows that a NB node can belong to topic if its governing node differs from the root and also belongs to topic. The nodes corresponding to *fail* and *test* in (38) are NB, but belong to topic, because the node corresponding to *student* is NB and depends on the root.

(38) The student who failed the test is WAITING for you.
[[ACT:student$_1^{cb}$ [[ACT:_$_1^{cb}$] RSTR:fail^{nb} [PAT:test^{nb}]]] wait^{nb} [PAT:you^{cb}]]

Note also that the CB node corresponding to *you* belongs to focus, because its governor is NB and depends on the root.

In (36b), repeated below with focus explicitly marked as (39b), it is precisely this clause which makes only *a box of chocolates* belong to focus:

(39) a. What did you present to which teacher?
b. I gave the teacher of English a box of chocolates.
[[ACT:I^{cb}] give^{cb} [ADDR:teacher^{cb} [RSTR:English^{nb}]]
[PAT:box^{nb} [RSTR:chocolate^{nb}]]]

On the other hand, I will ignore the following issues:

1. **Focalizers** are treated as verbal dependents (adjuncts). **Negation** is translated into a distinguished type of focalizer and treated in the similar way.[15]

2. A CB node may have one or more NB grammatemes. Such a node is called a **split node**.[16] A node's lemma has the same context-boundness value as the node itself. In a NB node, all grammatemes are assumed to be NB.

3. TFA is recursive. For every local tree, there is a **local TFA** and a **global TFA**. Local TFA is defined for nodes of the local tree, global TFA is defined for all nodes subordinated to the root of the local tree, i.e., for the subtree with that root. Global TFA is identical to local TFA if the local tree equals the whole tree (modulo the performative node). A single node in an embedded tree can thus belong, e.g., to the topic of the global TFA of the whole tree and to the focus of the local TFA of the embedded tree.[17]

4. In the topmost global TFA, corresponding to the whole tree, there is **at least one node or grammateme in focus**. No such condition applies to other global and local TFAs, which do not correspond to the whole tree.

4.2 A note on linguistic meaning and semantics

Although the issues of semantic interpretation do not stand in focus of the present work, the minimum requirement for a grammar, a syntactic theory and a formalism is to make a suitable syntax/semantics interface possible. No attempt is made in this section to actually propose a specific approach to semantic interpretation. The modest goal is to show (by parallel with other constraint-based frameworks) that it is possible to interface non-derivationally the constraint-based FGD syntactic component with a semantic component.

---

[15] Lexical negation is an exception.

[16] See Plátek, Sgall, and Sgall (1984, p. 73), Petkevič (1995b, p. 277), and Petkevič (in press). In (40b) only the grammateme of tense and aspect corresponding to the auxiliary is NB.

(40) a. Is Nick going to say it?

b. He HAS said it.

[17] In (38) above the relative clause is a part of the global topic. At the same time, the 'failed the test' part of the relative clause is its local focus.

Similarly as other formal linguistic theories, FGD is concerned with constraining (determining) the set of well-formed expressions of a language and assigning a meaning to such expressions. Unlike some of these theories, FGD does not aspire to cover meaning in the sense of extralinguistic content.

FGD distinguishes between *linguistic* meaning (represented as annotated tree graph, the tectogrammatical representation – TR) and *content* (cognitive, ontological, factual knowledge outside the system of language as such). Following the Prague School tradition, linguistic meaning is described as 'the patterning (or structuring) of content by (a specific) language'. The function of TR can be compared to Chomsky's Logical Form as a level of interface between the language and the layer of cognition ('semantic interpretation').

FGD views the relation between two neighbouring levels of linguistic description as that of 'asymmetric dualism', where several synonymous items at a 'lower' level may correspond to a single item at a 'higher' level and an ambiguous item at a 'lower' level may correspond to several items at a 'higher' level. Sgall (1967, p. 53) presents a precisely defined condition for distinguishing additional levels based on a requirement that the synonymy and homonymy relations are appropriately represented.

Similar assumptions concerning the distinction between two levels seem to hold at the interface between syntax and semantics. At TR, synonymous expressions share identical representations while an ambiguous expression receives multiple representations according to the number of its meanings. However, it is ambiguity rather than indistinctness that is represented explicitly at TR: a sharp distinction is made between 'language-internal' cases of homonymy and synonymy and the corresponding 'language-external' phenomena: indistinctness and vagueness. Cases of the latter kind (including the specification of reference, systematic ambiguities such as "*book* as a physical object" and "*book* as a text", group vs. distributive reading, and some cases of scope indistinctness) are assumed to be distinguished only at the layer of content. A thorough discussion of semantic issues from the viewpoint of FGD can be found e.g. in Sgall, Hajičová, and Panevová (1986, p. 8–18, 35–99). A more recent account, contrasted with views stemming from the tradition of formal semantics, can be found in Hajičová, Partee, and Sgall (1998). The following extract (Hajičová, Partee, and Sgall, 1998, p. 82) concisely explains the distinction between TR and semantic representation:

> TRs are objects of empirical enquiry, supposed to be language specific and patterned in a complex way by (surface) means such as word order and morphemes corresponding to (underlying) syntactic relations (actor, addressee, objective, adverbials of different kinds, etc.), to TFA and to morphological categories such as tense, modality, definiteness, number, etc. On the other hand, the semantic representations are constructed so as to correspond in an overt and direct way to cognitive oppositions (which do not immediately depend on the structure of in-

> dividual languages), such as those that, in a systematically elaborated semantic analyses, are rendered by prenex operators (and parentheses explicitly indicating their scopes) or by the means used in lambda calculus, in type theory, and so on.

Although in FGD the actual relationship between meaning and content is usually described in procedural terms as interpretation or translation, the approach seems to allow for treating the relationship as a restructuring relation, definable in declarative terms. Taking the issue of tectogrammatical functions and cognitive (content) roles as an example, the tectogrammatical function of Actor (expressed primarily as the syntactic subject) can be translated into – or interpreted by – several content roles: Agentive, Experiencer, Theme, depending on context, which includes the lexical setting. Conversely, a specific content role may typically be translated into – or structured by – a specific tectogrammatical function, but a one-to-many mapping is also possible, again depending on context. Thus, the same relation can be viewed as structuring or interpretation, depending on the viewpoint.

A move towards declarativity would allow for representing information which does not belong to the system of language proper in parallel with all other properties of the relevant expression as a part of a single formal object representing that expression.[18] There may be one interesting aspect in such a move, namely the fact that it would allow for the possibility of using information from any level to constrain TR, including that originating behind the frontiers of the system of language proper. Declarative specification of the relation may also ease the dispute on the status of semantic interpretation and the type of information needed for its disambiguation, see Hajičová, Partee, and Sgall (1998, p. 81–82), where B. H. Partee claims that:

> [...] (disambiguated, truth-conditional) semantic is indeed a level of linguistic structure, and what non-linguistic inferences are needed for is not for 'building' the semantic interpretation but for resolving ambiguities among different semantic interpretations that are consistent

[18]Parallel description of multiple aspects of an expression may be viewed as a step towards modularity in terms of paradigmatic vs. syntagmatic relations. Distinctions between levels are of paradigmatic nature and occur across expressions of any degree of granularity within a formal object representing that expression. On the other hand, the individual expressions (and the objects representing them) enter into mutual syntagmatic relationships. In a parallel description, syntagmatic relation between two expressions is licensed by evoking all relevant constraints, in parallel for all levels (constraints themselves can be organized in a modular way, reflecting the paradigmatic distinctions, see Oliva and Petkevič (1998)). Thus, during analysis, all aspects of an expression are combined with all aspects of another expression in one move.

While in a monostratal, non-derivational framework the paradigmatic and syntagmatic aspects appear as orthogonal dimensions, in a stratificational approach the paradigmatic aspect seems to take priority: syntagmatic relations are at work successively on individual levels, combining only items relevant to a specific level.

> with a given syntactic structure. Disambiguating quantifier scopes, or group readings vs. distributive readings, for instance, may require neither more nor less inferencing than disambiguating among the different topic-focus articulations that may be compatible with a given (surface, or 'outer') syntactic structure.

The position of the two co-authors (E. Hajičová and P. Sgall) presenting arguments in favour of FGD is summarized in the following extract, again from Hajičová, Partee, and Sgall (1998, p. 82):

> [...] specification of the understanding of vague or indistinct structures belongs to the semantico-pragmatic interpretation. The difference between TRs (as linguistic structures) and semantic representations (as the output of the interpretation) can be characterized as follows: TRs are objects of empirical inquiry, supposed to be language-specific and patterned in a complex way by (surface) means such as word order and morphemes [...]. On the other hand, the semantic representations are constructed so as to correspond in an overt and direct way to cognitive oppositions (which do not immediately depend on the structure of individual languages) [...].[19]

With all levels/layers simultaneously accessible, a possibility opens up of treating borderline cases in a way more consistent with their theoretical status as issues of a specific level/layer, rather than according to the status of factors constraining them.

The 'output of the procedure mapping TRs into the semantic representations' can have various forms. As possible candidates, Hajičová and Sgall (1999) present examples of tripartite structures of B. H. Partee[20] and presuppositional and exhaustive predications of J. Peregrin (Peregrin, 1995). Although it remains to be seen if these types of semantic representation would fit into the framework with a declaratively specified syntax/semantics interface, theories employing such a framework often exhibit remarkable flexibility in this respect.

Let us take HPSG as an example. Semantic representation in HPSG has been influenced by the theory of situation semantic, but the formalism assumed for HPSG has been shown to allow for the integration of other semantic theories. For example, the typed feature structures can correspond to

[19]The authors provide examples of differences between the level of linguistic structures and the level of semantic representations: 'modes and tenses, which are not fully adequate in the sense of strict criteria of logic and are often connected with underspecification,' correspond to modalities in the sense of modal logic and to the time axis of temporal logic. 'Also the asymmetries and cases of indistinctness in the domain of operator scopes and classificatory patterns such as gender (discussed under headings such as "categorization" in cognitive linguistics) witness that natural language systems [...] have to be empirically discovered, whereas the formal semantic systems are consciously constructed [...].'

[20]For a brief introduction see Hajičová, Partee, and Sgall (1998, p. 13-20).

predicate-argument structures with provisions for quantifier scoping. This can be illustrated by Chapter 8 of Pollard and Sag (1994). More recent proposals can be found i.a. in Davis (1997) and Davis and Koenig (1999) (concerning the issues of linking types of syntactic dependents with their semantic roles), Pollard and Yoo (1998) and Przepiórkowski (1998) (concerning quantifier scoping), and Copestake, Flickinger, and Sag (1997) (on a 'flat' semantic structure allowing for underspecification of scope).

Some of the proposals above address the issue of ambiguity which occurs only in semantic interpretation of an expression unambiguous from the syntactic viewpoint. In a framework where semantic interpretation is only applied to complete syntactic structures, this is not an issue: semantic ambiguities need not be reflected in syntax and can arise during interpretation. However, even if semantic interpretation is paired with every rule of syntax – as in Montague-type grammar or in its re-incarnations in a constraint-based grammar – semantic ambiguities need not be carried over to syntax.

Here I will leave the issues of semantic interpretation.

4.3 An extended note on word order constraints

Surface word order is constrained by abstract linguistic information available at different levels. In a stratificational framework, this information is applied sequentially, in the order of levels. In the present approach, word order is constrained by information at all levels simultaneously, in a monotonous fashion. I will first present a brief summary of Vilém Mathesius' idea of interacting and mutually competing word order principles and then proceed to its modification according to the FGD view on functional sentence perspective (FSP), including communication dynamism (CD). The modified principles will be shown to be compatible with the constraint-based formalism.

4.3.1 Vilém Mathesius's word order principles

The idea of various types of 'principles' simultaneously determining surface word order by mutual interplay and competition dates back to the work of Vilém Mathesius, especially to his comparative studies of the role of FSP in the word order in Czech and English. The following summary is based mainly on Mathesius (1942b), Mathesius (1942a), and Mathesius (1939).

Mathesius claims that there is a common set of such principles (or factors – 'činitelé') at work in both languages, namely *grammatical function principle* ('princip gramatické platnosti'), *adjacency principle* ('princip členské sounáležitosti'), *FSP principle* ('princip aktuálního členění větného'), *emphasis principle* ('princip důraznosti'), and *rhythmical principle* ('rytmický princip'). These principles can be briefly characterized as follows:

Grammatical function principle determines the position of an item in virtue of its syntactic function. This principle yields effects of two kinds: (i) cases of *fixed word order*, with the item positioned always in the same way (e.g., apposition in Czech), or (ii) cases of *common word order*, with the item positioned in a certain way, unless another principle takes precedence (in Czech and English, adjectives usually precede the modified noun, unless the adjective itself is postmodified – 'heavy', or there is another reason for reversing the order).[21]

Adjacency principle requires that members of a syntagm are ordered as continuous strings. This principle may be stronger than the condition of projectivity, because it may require that two nodes in a subtree are positioned adjacently (this is the case of the English subject, which is *usually* followed by the verb, without the possibility of another clausal component intervening).

FSP principle requires that components of a clause are ordered according to functional sentence perspective. In Czech, this principle yields *objective word order*, with the most dynamic item at the end of a clause.

Emphasis principle applies in emphatic or otherwise excited speech, yielding *subjective word order*, in Czech with the most dynamic item at the beginning of a clause.

Rhythmical principle requires that the position of an item does not interfere with a typical rhythmical structure. This structure may be based on stress (governing the placement of unstressed clitics in Czech) or size (proportion – 'heaviness') of the items.

The degree of applicability of a specific principle may be different in each language, the principles being ordered in a partial hierarchy according to their relative strength.[22]

For a specific language Mathesius distinguishes *principles of general applicability*, which are part of linguistic competence, from *principles of occasional applicability*, which depend on 'stylistic sensitivity' of the speaker/writer. In Czech, the latter include cases involving heavy constituents (*principle of proportional rhythm* – 'rytmus rozměrový'), avoidance of ambiguous structures (*principle of stylistic clarity*), concerns of fluency and euphonia, as well as those aspects of intonation which are not related to meaning. However, it is the principles of general applicability which are

[21]In the context of FGD, this principle would also be responsible for some cases of *surface deletion*, as in cases of grammatical control.

[22]As Mathesius (1942a) put it (quoted from Mathesius (1982, p. 130)):

> Word order in Czech is the result of interplay and competition of a range of factors, for which a typical scale in degrees of strength and importance holds.

of main linguistic interest and which are subdivided into *primary principles* ('hlavní činitelé'), determining the position of main clausal components, and *secondary principles* ('vedlejší činitelé'), which do not have such immediate effect.

In Czech, the primary principles are *FSP principle* and *emphasis principle*. The other principles play only secondary role, the most prominent among them being *rhythmical principle*, determining the position of clitics.[23] On the other hand, the primary principles in English are *grammatical principle* and *adjacency principle*. Where the latter principles are in conflict with *FSP principle*, English tends to use a syntactic construction satisfying both, but quite often *FSP principle* gives way completely. Whereas in unmarked Czech indicative clauses a focussed subject can follow an active verb in topic, obeying *FSP principle* (41), similar English clauses with a corresponding order (42a) are much less frequent than those where a different syntactic construction is used in order to convey the same FSP effect (42b), (42c), (42d) or where the effect is conveyed only in speech (42e).[24]

(41) Ve dveřích se objevil nezvaný host.
in doors REFL appeared uninvited visitor
'An uninvited visitor appeared in the door'

(42) a. In Bamborough Castle once lived a king who had a fair wife and two children.

b. There once lived a king and a queen as many a one has been.

c. At that moment there came a knock at the door.

d. The door was opened by an unlikely visitor.

e. An unlikely VISITOR opened the door.

To sum up, Mathesius proposes a universal set of word-order principles. These principles have different roles in different languages, thus a partial order of the principles can be specified for every language. This order predicts

[23] *Rhythmical principle* alone does not suffice to constrain the position of clitics if more clitics meet in a *clitic cluster*. The order of clitics in a cluster cannot be constrained by *grammatical function principle* (at least if grammatical functions are understood as tectogrammatical), because the order of clitical content words is not determined by their functions, and also because there are clitical function words, which have no functions. Neither can the order of clitics be constrained by any other principle proposed by Mathesius. Therefore, an additional ordering principle for clitics seems to be required. See below in point 6 on p. 104 for another case where an additional principle seems to be needed.

[24] Example (42e) is meant as a continuation of (42c), i.e., the subject is in focus and carries the sentential stress. Examples (42a), (42b), and (42c) are borrowed from Mathesius (1942b), who quotes them from Jacobs (1907) and Curme.

which principles win in case several principles compete for different word orders. There can be multiple winners, resulting in synonymous expressions.[25]

However, if *FSP principle* loses in favour of a different principle, it is often the case that the result still bears some FSP-related information, albeit not expressed by a mere permutation of main clausal components. This information can be encoded by means of sentential stress (as in (42e) above), by a marked syntactic construction or otherwise. Yet it is only the permutation of word order for which Mathesius' *FSP principle* can be made responsible. Thus, in addition to *FSP principle* and *emphasis principle*, it seems reasonable to assume a more general mechanism for expressing FSP. After all, other tectogrammatical concepts (such as syntactic functions) are expressed by similar mechanisms, relating tectogrammatical information with its surface form. A specific FSP can thus be expressed in various ways mentioned above (i.e., by a word order, a stress pattern, a syntactic construction, or any combination of them), and can be constrained by *principle of adjacency* (restricting the possibility of word order permutations).

4.3.2 Word order principles in FGD and as constraints

It is precisely this more general view of the role of FSP, which is held by FGD. The FSP of a specific utterance is represented together with other aspects of linguistic meaning at the tectogrammatical level. Of course, the TR of a specific expression does not determine the corresponding surface string(s) in any straightforward way. If grammar is viewed as (i) a set of well-formedness constraints on possible TRs, (ii) a set of well-formedness constraints on possible surface strings, and (iii) a set of constraints on correspondences between the two, the latter represents a very crucial part and can be compared to the knowledge or information humans or computers use to render the meaning of expressions in one language (such as English or Prolog) by expressions of another language (such as Czech or Assembler). Thus, the specification of the 'translation' relation between TR and surface expressions provides information which is not present (or not explicitly present) at the level of tectogrammatics and which comes into play during transition between this level and the level of surface strings.

Languages may differ in the preferred means for expressing FSP. Given the FSP-related notion of deep word order at TR, if surface word order is free from the need to express syntactic relations and can serve as the primary vehicle for conveying FSP, the 'translation' between TR and the surface string is more straightforward. Indeed, the degree of interference of the relational (deep/surface) information with the tectogrammatical specification

[25] Uszkoreit (1986) proposes a similar idea of competing LP (linear precedence) rules, which order pairs of constituents in the 'middle field' of German clauses according to several criteria of graded prominence: morphological case, noun/pronoun, topic/focus, determination.

is what distinguishes a language such as English from Czech, the degree of interference in English being greater. It seems that FSP is manifested in the surface expression wherever possible, i.e. unless it would defeat another (usually syntactic) constraint. When FSP cannot be manifested, the expression is ambiguous, for example, if the sentence (42e) is used in writing, without the focus being marked by italics or similar means.

In the standard FGD formalism, the two types of information are separated not only by being located at different levels, but also by being used in a specific sequence (cf., e.g., Sgall (1997)). In a constraint-based framework, all types of information can be used in parallel. This feature brings the constraint-based framework closer to Mathesius' proposal of principles determining surface word order simultaneously by mutual interplay and competition. The question is, whether it is possible in such a framework to describe the different role of tectogrammatical information and its interplay with information coming from other sources. More specifically, whether the dominant role of FSP in determining surface word order, intonation and/or the choice of a construction is compatible with the declarative specification of the deep/surface relation and the monotonous, non-destructive way of combining and merging information during the transition between the corresponding levels.

The answer depends on the extent to which the different kinds of constraints can be formalized as separate dimensions, possibly exhibiting close covariance. Thus, a specific FSP at TR can be constrained as related to the surface expression in a number of ways as a disjunction of possibilities, each constraining the other surface aspects of the expression: the FSP constraints covariate with the 'lower-level' constraints. These possibilities may include cases where a specific FSP is not manifested at all. Thus, Mathesius' interplay of competing principles is translated into a hierarchy of constraints, where a constraint higher in the hierarchy triggers the application of possibly multiple branches of lower constraints, preserving monotonicity.

4.3.3 Types of mismatches between deep and surface word order with function words ignored

With the reservations made above, I will make the hypothesis that Mathesius' principles concerning word order are compatible (i) with a theory using a level of deep syntax expressing FSP, such as FGD, and (ii) with a constraint-based formalism. I will illustrate the interaction of *FSP principle* as a constraint with other constraints on the basic types of mismatches between deep and surface word order (DWO and SWO). At the same time, I will indicate which of Mathesius' principles presented above is responsible.

First, I will assume that there are no function words among the items involved. Two elementary cases exist:

A. DWO and SWO coincide (FSP is manifested straightforwardly by word order, *FSP principle* applies).

EXAMPLE: Such a situation often obtains in Czech sentences exhibiting *objective order*, see (43) below.[26]

(43) Vánoční stromek strhla naše kočka.
Christmas tree tore down our cat
'It was our cat that tore down the Christmas tree.'
[[PAT:Christmas tree^{cb}]
tear_down^{nb} [[APP:we^{cb}] ACT:cat^{nb}]]

See (51) below for an English example.

B. DWO and SWO are different (FSP is manifested by other means than word order, *FSP principle* does not apply).

In the latter case, there are a number of possibilities, depending on the factor responsible for the mismatch:

4.3.3.1 Left dislocation of topic proper

Topic proper is located in SWO at a position which makes the domain of its governor discontinuous.[27]

EXAMPLE: See (45), where topic proper of the embedded non-finite clause appears sentence-initially.[28]

(45) Okno šéf dovolil pootevřít.
window boss allowed to open slightly
'As for the window, the boss allowed to open it.'

[26] Note that *vánoční stromek* 'Christmas tree' is treated as a single node.

[27] A governor's domain is meant to be one or more surface substrings corresponding to the TR tree rooted in the governor.

[28] Example (44), where only *šéf* is in topic, exhibits a similar type of DWO/SWO mismatch.

(44) Okno dovolil pootevřít šéf.
window allowed to open slightly boss
'As for the window, it was the boss who allowed to open it slightly.'
[allow^{cb} [[PAT:window^{cb}] [ACT:*gen*^{cb}] PAT:open slightly^{cb}] [ACT:boss^{nb}]]

A significant differences between (45) and (44) is in the number of nodes intervening between the DWO and SWO positions of the item corresponding to *okno*, in other words the number of nodes which must be 'skipped' by a hypothetical movement operation: there are two such nodes (*šéf* and *dovolit*) in (45), but only one such node (*dovolit*) in (44). A systematic approach to measuring word order complexity based on counting the number of non-projective phenomena is presented in Holan et al. (1998) and Holan et al. (2000).

[[ACT:boss^{cb}] allow^{nb}

[[PAT:window^{cb}] [ACT:*gen*^{cb}] PAT:open_slightly^{nb}]]

4.3.3.2 Non-final placement of intonation centre

FSP is expressed by a non-final placement of the intonation centre rather than by word order (*emphasis principle* applies).

EXAMPLES: This subsumes cases of *subjective order*, as in (46) below.

(46) MÁMA už je tady
Mum already is here
'Mum is already here.'

[[TIME:already^{cb}] [LOC:here^{cb}] be^{cb} [ACT:Mum^{nb}]]

The non-final placement of the intonation center occurs more often in languages with the so-called fixed word order, as in (47).

(47) ANOTHER MAN is going to have to appear to take his place.

[[[[APP:he^{cb}] PAT:place^{cb}] [ACT:_$_1^{cb}$] AIM:take^{cb}]
appear^{cb} [ACT:man$_1^{nb}$ [GNR:another^{nb}]]]

4.3.3.3 Dislocation due to syntactic constraints

Syntactic function or other syntactic properties of an item require that it is located at a SWO position different from DWO (*grammatical function principle* applies).

EXAMPLES: In a number of languages, interrogative and relative items are required to occur clause- or even sentence-initially and adjectives tend to precede the modified noun. In English, contextually-bound local and temporal adverbials often follow their non-bound governor and even the intonation centre of the sentence, as in (48).

(48) I received a very kind NOTE from her yesterday.

[TIME:yesterday^{cb}] [ACT:I^{cb}] [SRC:she^{cb}]
receive^{nb} [PAT:note^{nb} [GNR:kind^{nb} [MNR:very^{nb}]]]]

In Czech, as in English, the verb often follows the subject even if both are NB, see (49).[29]

[29] As already pointed out in footnote 23 above, if more clitics meet in a cluster of clitics in a Czech sentence, they have to follow a specific order, see (50).

(49) Vlk roztrhal ovci.
wolf tore sheep
'A wolf tore a sheep.'
[tear^{nb} [ACT:wolf^{nb}] [PAT:sheep^{nb}]]

(50) Včera jsem se mu ji pokusil představit.
yesterday AUX REFL he-DAT she-ACC attempted to introduce
'Yesterday I made an attempt to introduce her to him.'

Both content words and function words can participate in a single cluster, obeying a common set of ordering rules. Accordingly, the solution proposed below will be based on the same treatment of SWO for both content and function words as far as possible.

4.3.3.4 Dislocation due to the constraint of adjacency

As in §4.3.3.3 above, this type of mismatch is caused by a syntactic constraint, however, here no non-projectivity can arise. An item must be located in SWO within a local syntactic domain, next to another specific item of that domain (*adjacency principle* applies).[30]

EXAMPLES: In many languages, adjectives have to be placed adjacently to their nominal governors, adverbs adjacently to their adjectival governors. In English declarative clauses, subject nominal groups tend to precede the verb, while the verb itself tends to precede the object.

This constraint can be defeated by constraints of a different type. From a formal viewpoint, several kinds of violation may be distinguished:

1. Depending on the distance between the SWO and DWO positions of an item:

   (a) An item is located within its local continuous syntactic domain, but at a position non-adjacent to another specified item.
   EXAMPLE: Left dislocation of object in English declarative clauses, see (51). This is a case where *FSP principle* takes over even in English, so the two orders coincide.

   (51) *Him* she TOLD.
   [[ADDR:he^{c}] [ADDR:she^{cb}] [PAT:_^{cb}] tell^{nb}]

[30] A sentence can be recursively subdivided into non-overlapping but potentially discontinuous syntactic domains. A minimal syntactic domain corresponds to a word or a fixed multiword expression – strings which cannot be further subdivided using (paradigmatically) syntactic criteria. Syntactic domains typically correspond to TR subtrees and nodes, but in some domains function words are also present. If an item is located within a *local* syntactic domain, it does not make component domains of that domain discontinuous. There may be other domains in a addition to syntactic ones, the most obvious example being phonological domains (clitic clusters or clitics with their host).

The object can be located in a way which makes the local syntactic domain discontinuous, as in (52).

(b) An item is located discontinuously from its local syntactic domain, see (52).

(52) *Him* Mary says Sue TOLD.
[[ACT:Mary^{cb}]
say^{nb} [[PAT:he^{c}] ACT:Sue^{cb}] PAT:tell^{nb}]]]

2. Depending on whether the DWO and SWO positions are different for an item together with all of its TR subordinates, or whether only the item and a subset of its subordinates are affected:

(a) An item is located at a different position together with all of its dependents.

(53) *Potato dumplings* she LIKES.
[[RSTR:dumplings^{c} [PAT:potato^{cb}]]
[ACT:she^{cb}] like^{nb}]

(b) An item is located at a different position, with some or all of its dependents staying at the position corresponding to DWO.

In (54) the object of comparison is not dislocated together with the adjective.

(54) menší vesnice než Lhota
smaller village than Lhota
'a smaller village than Lhota'
[village^{cb} [[COMP:Lhota^{cb}] EXT:small^{nb}]]]

4.3.3.5 Deletion due to syntactic constraints

Syntactic function or other syntactic properties of an item require or permit that a semanteme has no correlate in SWO (a case of surface deletion, *grammatical function principle* applies). For an example see (47).

4.3.3.6 Dislocation due to a stress pattern

Rhythmical structure requires an order different from DWO (*rhythmical principle* applies).

(55) Někdo ho nařídil sledovat.
somebody him ordered to watch
'Somebody ordered to watch him.'

[[ACT:somebody^{cb}]
order^{nb} [[PAT:he^{cb}] [ACT:*gen*^{cb}] PAT:watch^{nb}]]

4.3.4 Ordering function words

Now what about the situation where function words are involved? It seems that only some of the word order principles are relevant for function words. Before presenting a list parallel to the list of mismatches in DWO and SWO orders of content words, several remarks are due:

1. Based on their role at TR, function words can be divided into two types from the viewpoint of their role at TR, similarly as other function morphemes. In both cases a single TR semanteme node stands for the content word and the hosted function word(s).[31] The difference is in whether a function word contributes to the linguistic meaning (corresponds to a grammateme) or not:

   (a) The function word presents a specific contribution to the linguistic meaning of its host.[32] This contribution is reflected in a corresponding semanteme node at TR by an appropriate grammateme (or several grammatemes).

   Examples include the morphological grammatemes of tense (the future auxiliary in Czech), aspect (the perfect tense auxiliary in English), definiteness (article in English), types of coordination/subordination (conjunctions).

   (b) The function word does not contribute to linguistic meaning of its host. Its role is only to provide information necessary in order to establish syntagmatic relations among content words in a construction, or to be part of a complex lexeme. There is no grammateme corresponding to a function word of this kind at TR.

   Examples include function words expressing morphological categories of agreement (the past tense auxiliary in Czech), verbal particles (with phrasal verbs in English, with inherently reflexive verbs in Czech), prepositions as a part of subcategorization requirements.

   A more thorough investigation shows that rather than classifying function words, it might be more appropriate to classify their properties (morphological categories). A single word may bear several properties of different types: the Czech future auxiliary is the bearer of the value of tense and, at the same time, of the agreement features of person and number.

[31] I will use *host* as the term for a content word in relation to a function word which is needed to co-specify the content word's meaning or to make a syntagm involving the content word complete.

[32] According to Sgall (1997), '[such function words] are immediately semantically relevant'.

This distinction is important for TR and the way it relates to the set of morphemes at the level of morphemics. On the other hand, with respect to SWO there seems to be little advantage gained from making this distinction among function words.

2. No function word can be the cause of a real mismatch between DWO and SWO, because function words do not occur as nodes at TR, and therefore they are not subject to DWO. It is only through their hosts that they can participate in structural relations at TR.

3. Since a function word does not count as a TR node, its position in SWO cannot be determined straightforwardly by *grammatical function principle*, simply because a function word cannot have a (tectogrammatical) syntactic function. In this way, a function word's SWO properties are parasitic upon the properties of its host.

4. This is usually true also about *FSP principle* and *emphasis principle*, but there are exceptions. Although a function word is not subject to CD ordering, one or more grammatemes as the tectogrammatical representation of a function word can be subject to contextual boundness (CB) specification and topic-focus articulation (TFA), cf. *split node*, point 9 on p. 87. Such a function word cannot be viewed as parasitic upon the CB and TFA properties of its host, but rather is itself subject to *TFA principle* and *emphasis principles*.

5. It seems to be the interplay between *adjacency principle*[33] and *rhythmical principle* which is mainly responsible for determining the SWO position of a function word.

6. In some cases, the principles listed above still do not determine the proper position of a function word. This case obtains if more function words are hosted by a content word, are positioned adjacently to it, and their mutual order is not constrained by *emphasis principle* or *rhythmical principle*. Similarly as in the case of a clitic cluster (see footnote 23 on p. 96), an additional ordering principle seems to be required.

After these remarks, I will now present a list of types of SWO for function words. Function words can either be located adjacently to their host (*adjacency principle* applies, §4.3.4.1), or not (*adjacency principle* is violated). There are degrees of how severely *adjacency principle* is violated, which are parallel to adjacency violations concerning content words (see S4.3.4.2 and §4.3.4.3). Similarly as for content words, there are also various reasons why *adjacency principle* is violated (see §4.3.4.4 and further below).

[33] *Adjacency principle* requires the members of a syntagm to be ordered as continuous strings. In order for this principle to apply to function words, the notion of syntagm must be interpreted as including both content and function words.

4.3.4.1 Function words adjacent to their hosts

Function words are located adjacently to their host (*adjacency principle* applies).

EXAMPLES: Although this may seem to be a frequent case (prepositions, articles), a closer inspection reveals that in fact a function word can be separated from its host by the host's dependents (see §4.3.4.2 below). It is therefore not easy to find an example. One of the few cases in Czech is reflexive particle with deverbal noun (from inherently reflexive verbs), see (56) (however stylistically dubious).

(56) otcovo zachmuření se
father's frowning REFL
'father's frown'

The grading particles *more* and *most* with adjectives and adverbs are English examples of the same kind. In Japanese, a strictly head-final language, particles and auxiliary verbs immediately follow their hosts, see (57).[34]

(57) Aruki nagara hon o yomu no wa abunai desu
walking while book ACC read N-OR T-OR dangerous COP
'It is dangerous to read while walking'

4.3.4.2 Function words within local domain

One or more function words form a continuous string with their host *and* its (immediate) dependents, but they are not adjacent to the host itself and they do not split the domain of a dependent. In other words, they are located within the continuous local syntactic domain of the host, but at a position non-adjacent to the host (cf. point 1a on p. 101). It is the leftmost position of the local syntactic domain, which is usually occupied by the function word(s). This is the prototypical SWO placement of function words.

EXAMPLES: Prepositions, determiners and subordinating conjunctions all precede any content words dependent on the host. Coordinating conjunctions behave in a special way, depending on their type and the type of coordinating construction (syndetic, polysyndetic, asyndetic). In the commonest case of a serial conjunction used syndetically, the conjunction precedes the (domain of the) last conjunct.

In Czech, another frequent case is the past and conditional auxiliary, which in their finite clause tend to behave as second position clitics. This applies also to reflexive particles, but only if hosted by a finite verb. Then they are clause-bounded (58).

[34] The gloss ACC stands for 'accusative particle', N-OR for 'nominalization particle', T-OR for 'topicalization particle', and COP for the copula.

(58) Proč bychom se ze života netěšili
why COND REFL from life enjoy
'Why shouldn't we enjoy life'

4.3.4.3 Function words outside local domain

One or more function words are located discontinuously from the local syntactic domain of their host (cf. point 1b on p. 102).

EXAMPLES: This is the case of the Czech reflexive particles if hosted by a non-finite verb.[35] They can 'climb' from an embedded clause into the second position of a higher clause (in (59) the verb 'play' is reflexive: *hrát si*).

(59) Děti si šli hrát
children REFL went to-play
'The children have gone to play'

This phenomenon can result in haplology involving the reflexive particle. In (60) both *rozhodli se* and *projít se* are inherently reflexive verbs.

(60) Rodiče se rozhodli jít projít
parents REFL decided to-go to-take-a-walk
'The parents have decided to go for a walk'

Next, the various reasons why *adjacency principle* can be violated are listed.

4.3.4.4 Adjacency violation due to a stress pattern

Rhythmical structure requires a different order (*rhythmical principle* applies). This is the case of Czech clitics (see the examples involving clitics above) and prepositions.

It should be noted that all clitics in Czech, no matter whether function words or content words, are subject to the same SWO constraints.

4.3.4.5 Adjacency violation due to an ordering principle

A principle for ordering a cluster of clitics or function words applies.

EXAMPLE: See (50), repeated here as (61).

(61) Včera jsem se mu ji pokusil představit.
yesterday AUX REFL he-DAT she-ACC attempted to introduce
'Yesterday I made an attempt to introduce her to him.'

[35] Reflexive particles share this behaviour with pronominal second position clitics, see (55) above, in the section dealing with content words.

4.3.4.6 Adjacency violation due to split node

In the case of a 'split node', TFA involving a function word is manifested by word order (*TFA principle* applies).

EXAMPLE: In (63) it is only the grammateme of future tense which belongs to focus.[36]

(63) Chodit já tam <u>budu</u>.
to go I there AUX
'I WILL be going there.'

4.3.5 A list of constraints on surface word order

Content words and function words differ in how they are related to TR: content words correspond directly to semantemes, while function words either correspond to grammatemes within semantemes or have no straightforward representation at TR at all. In the latter case, they serve as markers of syntagmatic relations (in phenomena such as agreement), or as parts of complex lexemes.

On the other hand, it can be seen that as far as SWO is concerned, content words and function words share a number of regularities. If the issues of SWO on the one hand and of the correspondence between semantemes and surface words on the other are to a large extent orthogonal, the regularities concerning SWO should be stated independently of those governing the correspondence.

The following list is an attempt to modify Mathesius' word order principles in the light of the insights of FGD and the observations made above.

4.3.5.1 General constraints on word order

1. The relative SWO of every two content words corresponds to DWO of the corresponding semantemes, unless any of the cases in the list of Special SWO conditions apply (see below).

2. A function word is ordered adjacently to its host, their order being determined by a syntactic constraint, unless any of the cases in the list of Special SWO conditions apply (see below).

3. For the relative SWO of every two function words in a cluster the list of Special SWO conditions applies.

[36] Of course, in the case of a 'split node', TFA involving a function word can also be manifested only by emphasis (*emphasis principle* applies), as in 62.

(62) Já tam <u>BUDU</u> chodit.
I there AUX go
'I WILL be going there.'

4.3.5.2 Special SWO conditions

1. The word ordered first in SWO is the intonation center of the utterance and corresponds to focus proper.

2. A syntactic constraint requires otherwise.

3. A stress pattern requires otherwise.

4. A word is ordered first in a SWO domain, the domain is larger than that of its corresponding tectogrammatical local tree, and the word corresponds to topic proper or to contrastive topic.

If two or more conditions compete for different orders, language-specific priorities are applied with the possibility of multiple outcomes.

With all of the special SWO conditions, specific constraints on locality must be satisfied (parallel to the degrees in violation of *adjacency principle*).

4.3.5.3 Locality constraints on SWO

It is far from straightforward to specify constraints on locality. An adequate description of this topic has been a long-standing subject of inquiry in theoretical linguistics. In the GB/P&P tradition, terms such as barriers to movement, islands, and bounding nodes have been introduced to describe constructions which prohibit violations of locality relative to the type and functional status of the construction as well as of the candidate violator and the depth of its syntactic embedding.

Here the problem is slightly less general, because the task consists only in constraining special cases where DWO does not correspond to SWO and function words are not adjacent to their hosts, for specific reasons given by the Special SWO conditions. Still, the task is huge for the following reasons: (1) only some locality constraints are language-universal (or language-group-universal), and (2) only some locality constraints apply indiscriminately across the individual sets of word classes, construction types, syntactic functions and levels of embedding.

Some locality constraints are robust: prepositions are PP-bounded, Czech clitics are finite-clause-bounded, verbs are clause-bounded. Other locality constraints can be violated in some contexts: a Czech adjective can be extracted from its NP in a literary style or poetry.

The issue of interacting constraints on word order will be addressed in more detail in §5.3.

4.4 Governors or heads?

In the present work, the terms *governor* and *dependent* are used to denote semantemes – members of the tectogrammatical dependency relation, a special case of syntagmatic relation. Tectogrammatical dependency relations are used as the primary means for functional representation of language expressions.

However, in the description of syntagmatic relations in a language, semantemes may not be the only elements and governor-dependent relation may not be the only relation necessary. The reasoning behind the assumption that other elements (function words and non-terminal nodes) and other syntagmatic relations (licensing and representing them) may be needed is as follows:

1. In addition to stating well-formedness conditions on possible TRs, it is one of the primary tasks of grammar to describe the relationship between tectogrammatical (or other kind of) representation and the language expression as such.

2. Some expressions may include function words which do not correspond directly to a semanteme and which nevertheless enter into syntagmatic relations (syntagms) with other elements. Assuming that such relations are necessary, they have to be properly described.[37]

3. Paradigmatically, it seems most natural to treat function words on a par with bound synsemantic (= functional) morphemes, i.e., as expression of formal grammatical categories, represented by morphological grammatemes (features) appropriate to a relevant semanteme.[38] Syntagmatically, however, they participate in observable relations (ordering, prosodical) with various other elements in the clause or sentence. In this respect, they often form a natural class (e.g., second position clitics) with a subset of content words. If their paradigmatic status is preserved, their syntagmatic behaviour seems to be best described together with other elements of the same class, even if the class is paradigmatically heterogeneous.

4. A syntagmatic relation involving at least one function word cannot be a tectogrammatical relation. Therefore, additional ways of describing such a relation are needed. The standard approach in FGD is to handle function words by rules licensing and moving function words during the

[37] In the standard version of FGD, function words occur only as elements of morphemic strings and syntagms involving function words have no place in syntax. By Occam's razor, such syntagms are considered unnecessary, unless evidence about their usefulness is presented. (P. Sgall, p.c.)

[38] The very existence of terms *free morpheme* and *bound morpheme*, well-established across linguistic theories (Trask, 1993; Čermák, 1997), attests to this view.

transformation of the tree into a string and the string into a tree. Their application is conditioned (also) by syntactic structure (TR).

An alternative solution is to use the general combinatory mechanism already available in grammar for the treatment of content words, introducing additional types of syntagmatic relations for the treatment of function words.

5. Similarly as rules for licensing and moving function words in the standard version of FGD formalism, (constraints on) syntagmatic relations involving function words do not correspond to dependencies at TR. Nevertheless, actual occurrences of syntagmatic relations (syntagms *in presentia*, cf. Čermák (1997, p. 121)), as well as their description in grammar, constitute specific entities. These entities represent the occurrence of specific grammatemes as conditional upon the presence of a specific function word.

6. Once we allow for the existence of function words and syntagms involving function words in the grammar, we have to admit an additional structure – *derivation tree*, different from (but corresponding to) TR – the *representation tree*. Derivation tree immediately reflects the way formulas of grammar are applied to a specific expression.[39]

7. The primary means for constraining the deep/surface relationship is valency satisfaction. This mechanism is quite general and allows easy extension in order to handle function words. As a result, relations between elements in a derivation tree can be treated as dependencies of a special kind. These dependencies differ from tectogrammatical dependencies in several ways and this distinction is made explicit by using the terms *head* and *daughter* instead of *governor* and *dependent*.

8. In a grammar which requires the identification of head and daughter components in every syntagm, syntagms involving function words are obliged to make the head/daughter distinction as well. In constraint-based grammars, this distinction is expressed by sharing a distinguished subset of features ('head features'), grouped in a single object, by the head daughter and the mother categories. Also, the head usually makes valency requirements on non-head components, although it may

[39] Dependency tree can be read off a derivation tree by collapsing all nodes along the path from a terminal (lexical) node up to the maximal projection of that node into one. In the present framework, TR is contained within every node of the derivation tree as one of its aspects, representing the (sub)tree of which the node is the root. TR of terminal (lexical) nodes represents the lexical items. The geometrical properties of TR need not correspond to those of the dependency tree read off the derivation tree by collapsing nodes along the head paths, because TR lacks function words as nodes and its nodes are horizontally ordered according to deep word order. On the other hand, syntagmatic relations involving function words do not need to result in additional projections in the derivation tree.

not always be true.[40] This blurring of the notion of headedness can be viewed as necessary in the case of syntagms involving function words, where the governor/dependent and head/daughter relations need not coincide. Whenever it is the function word rather than the content word which makes specific requirements on its counterpart, it seems natural to assume that the head is the function word.[41]

However, there are ways of letting a function word make specific requirements on its host, while maintaining governors as heads. One of them means treating all function words as *markers* in the sense of Pollard and Sag (1994): a function word carries a specification of its host as well as of its contribution to the host's properties. The parallel with markers is not precise in at least the following points: (i) a function word needs more power than a marker in order to manipulate the host's properties, (ii) a function word can have another function word as its immediate host, and (iii) some hosts (such as inherent reflexives) require function words rather than the other way round.

Another way can be based on having the lexical component derive quasi-valency requirements on function words in the entries of the hosts. These issues will be discussed in more detail in §5.1.2.

9. Syntagmatic relations involving function words represent the most striking case in favour of additional entities in grammar and in derivation tree beyond those existing at TR. However, arguments for yet another type of entities can be provided.

   Language expressions exhibit a rather high degree of compositionality and this is precisely why it is possible to write grammars at all. Rather than licensing every possible (sub)tree, grammar rules (in conjunction with lexical specifications, especially valency requirements) describe syntagmatic relations as building blocks of which a (sub)tree may consist. Entities licensed by such grammars reflect the degree to

[40] In head-adjunct phrases of Pollard and Sag (1994) it is the non-head adjunct that selects properties of its *syntactic* head, the adjunct itself being the *semantic* head, which is due to sharing the distinguished subset of semantic features by the adjunct and the mother categories.

[41] Cf. Netter (1994, p. 310), who consistently analyses 'functional categories' (roughly equivalent to function words) as heads which subcategorize for 'substantive categories' (roughly equivalent to content words):

> One of the main reasons for this hypothesis is that we want to avoid the introduction of an additional selection mechanism, but rather assume one single homogeneous selection mechanism for all relations between a head and the phrases whose presence is required and licensed by that head. Therefore, it will be the functional category that requires the presence of a major category while the major category may be unspecified as to whether a functional category is to be present.

which valency requirements of their head is satisfied. These entities can correspond to a single lexical item (with any number of valency requirements including zero), to a fully saturated (sub)tree, or any partially saturated (incomplete) (sub)tree licensed by a sequence of applicable rules.

To sum up the points made above, given a non-derivational framework it seems useful to introduce objects and relations not present at TR: the head/daughter distinction in addition to the governor/dependent distinction, derivation tree in addition to tectogrammatical tree, and nonterminal objects in addition to semanteme nodes. At the same time, the objects and relations newly introduced do not seem to present any challenge to the privileged position of TR within the description of language, because they are merely used in place of a different mechanism already present in FGD for describing the transition from TR to the string of morphemes and back.

4.5 A note on coordination and apposition

In theoretically-oriented work in FGD, coordination and apposition are treated as the third dimension of the dependency tree: Plátek, Sgall, and Sgall (1984, p. 76–86), Sgall, Hajičová, and Panevová (1986, p. 140–141), Petkevič (1995a). The third dimension is viewed as multiple filling of a position corresponding to a node at TR: "a position occupied by a single node in a dependency tree may also be occupied by a coordinated construction – a sequence of nodes" (Plátek, Sgall, and Sgall, 1984, p. 76–86). A sentence such as (64) can be represented by the simplified TR in Fig. 4.1.[42]

(64) A duckling and several hens of our neighbour disappeared.

The curved line links coordinated edges and is labelled by the functor specifying the type of coordination: *Conj(unctive)* in the present case (to be distinguished from *Disjunctive, Adversative, Graded, Consequential,* etc. It should be noted that whenever a reference is made to the entire coordinated construction, rather than to an individual conjunct, the target of reference is not a node, but rather a set of nodes. The coordinated structure is neither a constituent nor a dependency structure with a single representative element except for the structure itself.

In preference to a graphical image, TRs including (coordination and apposition) receive a linear notation, an extended version of the notation exemplified by (69) below or used in examples on p. 99 and further. The extension consists in using a different kind of brackets or different subscript to the existing brackets. The following example (65) is a rather liberal paraphrase of the notation used in Petkevič (1995a).

[42] The intended meaning is that where the prepositional phrase modifies the whole coordination.

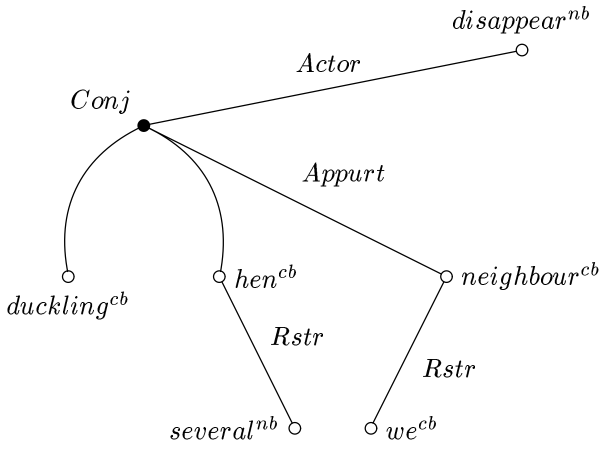


Figure 4.1: Coordination at TR as the 3rd dimension

(65) [[{ duckling^{cb} hen^{cb} $_{rstr}$[several^{nb}] }$_{conj}$
$_{appurt}$[[we^{cb}]$_{rstr}$ neighbour^{cb}]]$_{act}$ disappear^{nb}]

Braces enclose coordinated elements. Functors mark brackets (i.e., edges) rather than nodes, and they always stand between the governor and the dependent linked by the corresponding edge.

A more technically oriented solution has often been used, most recently in Hajičová, Panevová, and Sgall (2000): instead of introducing another dimension, or another set of brackets or subscripts thereof, coordinated structures are represented as regular dependency structures with conjunction as the governor and conjuncts as its dependents, see Fig. 4.2.[43]

As will be shown below in §5.2.5, the formalism proposed here allows for taking either direction. In that section, pros and cons of both approaches to representing coordination will also be discussed.

4.6 Conclusions

There are two important results: a list of features of FGD which will be addressed in the following sections, and a classification of word order phenomena based on word order principles of Vilém Mathesius. The latter will be immediately used in the formulation of specific word order constraints.

Hopefully, the notes on semantics, the head/governor distinction and coordination have made the linguistic picture of the intended system less fuzzy.

[43]It does not seem appropriate to assume a contextual boundness property for the conjunction node, unless it is adopted for grammatemes in general. The same conclusion is reflected in the assignment of NIL to the tfa attribute of conjunctions in Hajičová, Panevová, and Sgall (2000, p. 57).

Edges linking the conjunction and the conjuncts are marked by *Co* – they do not represent any tectogrammatical function.

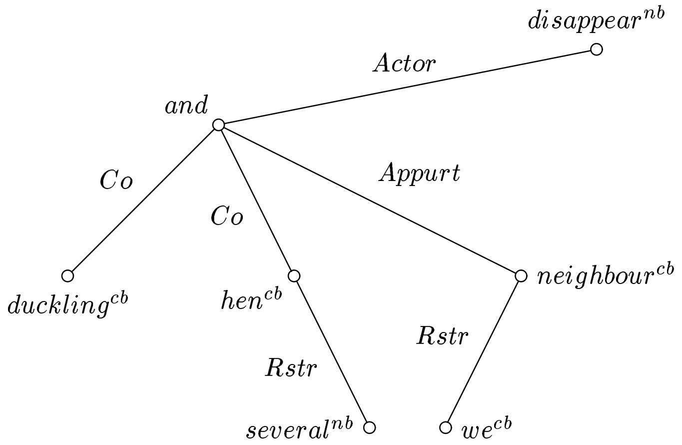


Figure 4.2: Coordination at TR with conjunction as the governor

Chapter 5

The architecture

In this chapter I will introduce the formal description first by specifying the role of tectogrammatical representation and derivation structure, and then by proposing formal objects for representing the tectogrammatical level and the morphemic level. Finally, I will briefly clarify a few lexical aspects necessary.

5.1 General assumptions

There are certain assumptions embodied in the formal framework – see §3.2. Additional assumptions are based on the possibilities offered by the formalism and useful in the context of FGD.[1]

5.1.1 The issue of representation

Each language expression is represented as a formal object, consisting of several uniform parts, modelled intentionally in a way immediately resembling the object *sign* in the theory of HPSG, in order to allow easy adoption of solutions available in that theory and adequate within the context of FGD. However, nothing substantial concerning the assumed theoretical background of FGD hinges on this similarity. There is no reason for not adopting a different, more adequate setup at the cost of losing some compatibility with the tradition loosely denoted as 'mainstream HPSG'.

The overall structure of the type *sign* can be seen from Fig. 5.1. The parts most relevant to the theoretical background are values of the attributes DEEP and SURFACE. The former correspond to the level of tectogrammatics (deep syntax, i.e., the dependency tree), the latter to the level morphemics (string of objects representing morphemes, ordered according to the surface

[1]Nevertheless, some features of the present approach are necessarily influenced by the adopted formalism, rather than by the theoretical background. An example of this can be seen in the distinction of mother and daughter categories – see footnote 2 below.

$$\left[\begin{array}{ll} \textit{sign} & \\ \text{PHON} & \textit{list(phonstring)} \\ \text{SYNSEM} & \left[\begin{array}{ll} \textit{synsem} & \\ \text{LOCAL} & \left[\begin{array}{ll} \textit{local} & \\ \text{CATEGORY} & \left[\begin{array}{ll} \textit{category} & \\ \text{HEAD} & \textit{head} \\ \text{VALENCY} & \textit{valency} \end{array}\right] \\ \text{DEEP} & \textit{d-list} \\ \text{CONTENT} & \textit{content} \end{array}\right] \\ \text{NONLOCAL} & \textit{nonlocal} \end{array}\right] \\ \text{SURFACE} & \textit{s-list} \end{array}\right]$$

Figure 5.1: Schematic picture of the subparts of *sign*

word order). They constitute the obvious targets for regularities concerning (i) the underlying syntactic structure and the representation of its elements, and (ii) the interplay of the deep and surface word order.

Additionally, there are parts corresponding to the syntactic combinatory potential of the expression: CATEGORY and – optionally – to its semantic interpretation (CONTENT).

Fig. 5.1 shows the structure of the type *sign*, which has two subtypes: *lexical* and *non-lex(ical)* (similar to the HPSG's sorts *word* and *phrase*). The *non-lexical* type has two additional attributes, which record immediate syntactic components of the expression, mimicking the local derivation tree: a *sign*-valued attribute HDTR (HEAD-DAUGHTER) and a *list(sign)*-valued attribute NHDTRS (NONHEAD-DAUGHTERS).[2] The actual string of phonemes (or – for the present purpose – graphemes) of the expression is represented as the value of the attribute PHON(OLOGY). Internal structure of these parts will not be discussed extensively in this section. Their structure and role can be paralleled to their corresponding counterparts in HPSG.

The placement of the part representing tectogrammatical information (value of DEEP) within the object *local* on the same level as *category* and *content* predicts the obvious fact that tectogrammatical representation of an

[2] It may seem that the type *non-lexical* with the additional two attributes reveals that phrase structure may be the primary descriptive component of the present approach, but this is not the case. The *non-lexical* type is merely a way of distinguishing representations of expressions corresponding to lexical items from those corresponding to syntactic trees. The reason why the attributes HDTR and NHDTRS are present is rather formal: rather than being an expression of the phrase-structure based view of syntax, their role is to allow for imposing constraints on local derivation trees. Kathol (1995, p. 149-151) proposes to get rid of these attributes by introducing relational constraints to license phrasal signs as a combination of other signs, without the requirement that the constraint talks about a single structure (the mother, including its daughters). However, the formalism adopted here does not allow to use constraints on formally separate objects.

expression is a structure recursively composed from a governor and its immediate dependents.[3] On the other hand, the part representing a morphemic string (value of SURFACE) does not necessarily correspond to a local tectogrammatical tree, i.e., to a permutation of strings corresponding to nodes of that tree. It is required to be a permutation of surface forms corresponding to all nodes of the tectogrammatical (sub)tree representing the expression. Being the result of a complex interplay of syntactic, discourse and prosodic phenomena, the value of SURFACE (representation of morphemic string) is freed from the requirement of locality by being placed on the same level as the list of phonemic strings and the objects representing daughters in the derivation tree.[4]

For a more complete account of language data, at least one addition would be required, namely a part representing prosodical patterns. Similarly as the morphemic string, this part must be positioned in a way allowing the statement of facts orthogonal to syntactic relations.

5.1.2 The issue of compositionality

There is a natural requirement that the relation between a surface string (a language expression) and its representation should be compositional: the representation of an expression is a function of the representations of the parts of that expression. Indeed, compositionality is a characteristic without which a grammar and the structures it licenses would be poorly suited to the task of describing and representing a human language

Now the parts of a tectogrammatical tree, as a representation of an expression, are its subtrees and individual nodes. Subtrees correspond to instances of syntactic dependency structures and nodes to content words. Tectogrammatical nodes incorporate tectogrammatical interpretation of functional morphemes in the form of grammatemes or functors, regardless of whether the grammatemes or functors are realized as bound morphemes (endings or affixes) or independent morphemes (function words).

However, many important generalizations would be missed if grammars would not describe other entities: individual syntagms corresponding to tectogrammatical dependency relations and also to constructions involving function words. In order to let grammar describe such entities, a way must be found of composing/decomposing the tectogrammatical tree from/into subtrees, nodes, and the individual dependency relations and independent functional morphemes.

---

[3]This aspect of the structure of TR can be compared to the setup of a semantic interpretation recursively composed from a predicate and its arguments.

[4]Under the assumption that only *synsem* objects can be accessed by valency requirements, there is yet another decision to be made: whether the part representing a morphemic string should be placed within the *synsem* object (on the same level as the objects *local* and *nonlocal*), or – as in Fig. 5.1 – at the topmost level. The present hypothesis is that the morphemic string is not accessible to valency requirements.

It is the latter two entities which defy straightforward compositional specification within a tectogrammatical tree, and this is why the syntactic representation of an utterance as a tectogrammatical tree necessitates a division between syntactic representation and syntactic derivation. Syntactic derivation is understood here precisely as the relation between a surface string and its tectogrammatical representation. Derivation tree is a way of representing syntactic derivation as a structure

The task of specifying such a relation may be compared to the standard goal of relating a surface string with its (extra-linguistic) meaning and the standard methods of integrating/disintegrating expressions into/from larger wholes may be invoked.

5.1.2.1 What kind of derivation structure?

In the following, I will consider three ways of specifying the (de)compositional relation, compatible with the proposed formal framework. I will identify them by the trace they leave behind – a derivation tree: flat, binary-branching and mixed.

The same reasoning can be expressed in terms of the formal entities (types) introduced above. Unlike *lexical* signs, which are constrained by the lexicon (including morphological component), *non-lexical* signs are composed from other signs and constrained by the syntactic component. Non-lexical signs corresponding to complete tectogrammatical subtrees (maximal projections of lexical items within the derivation tree) may be the only kind of non-lexical signs, or other non-lexical signs may correspond to partial subtrees (non-maximal projections). A question arises, whether and how many of these additional signs are needed, i.e., what shape the derivation tree has.

One possible answer is to postulate a sign corresponding to every distinct partial tectogrammatical subtree with 1, 2, ... n nodes immediately dependent on the governor, where n is the number of such nodes in the complete tectogrammatical subtree, with an additional sign for every function word hosted by the governor. In phrase-structure terms, such signs would correspond to categories along the head projection path in a binary branching structure, where every sign is composed of exactly two daughters, a head daughter and a non-head daughter. In dependency terms, these signs would show how individual syntagms add up along the derivation path ending in the complete subtree.

Another possible answer is to keep a minimum number of non-lexical signs, perhaps only those corresponding to complete subtrees, excluding all of those corresponding to partial subtrees. In phrase-structure terms, such signs would correspond to mother categories in a completely flat structure, where all dependents are sisters. In dependency terms, these signs would correspond to tectogrammatical subtrees.

There is some room in between the two extremes. For example, in a conservative approach the nodes of the tree may correspond to the traditional phrase-structure categories, especially in distinguishing structurally the subject of a clause (NP) from the verb phrase (VP).

There are arguments in favour of either of the two solution at the opposing ends of the scale: Binary structure reflects the insight present in dependency-based approaches about the universal presence of binary dependency relations between the governor and the dependent. Such relations are synonymous with syntagms as the building blocks of syntax. On the other hand, given that surface order not necessarily corresponds to the yield of even a binary-branching derivation tree with the still ensuing need for a separate linearization component, spurious non-lexical signs can result. In other words, it may not be clear which intermediate signs are legitimate projections and which are not.

Flat structure matches the tectogrammatical tree and keeps the number of entities at a minimum, on the other hand it may be more difficult to reconcile with compositional semantics. From the viewpoint of integration into the proposed framework, the 'binary structure' solution may seem easier, because constraints on different types of syntagmatic relations (complementation, adjunction, analytical morphology) can be applied to distinct local trees. The 'flat structure' approach requires that distinctions between types of syntagms do not presuppose the existence of distinct signs. There may be two ways of collapsing constraints on different syntagm types into a single local tree:

1. The differences between the syntagm types can be lexicalized, i.e., neutralized by the introduction of derived lexical entries within the lexical component. Thus, valency frame of a *lexical* category can receive slots for as many dependents – including adjuncts – as there are such dependents in the actual expression.[5] A single constraint can then be applied to a local tree corresponding to the maximal projection and to the complete tectogrammatical subtree, making sure that the governor discharges all its valency requirements at once.

2. The differences between the syntagm types can also be handled by relational constraints on a general type of non-lexical sign. A general implicative constraint on the local tree can cover a number of more specific instances by a disjunction of relations taking care of the individual types of dependents.

5.1.2.2 Adjuncts

There seems to be a good argument for making the complements/adjuncts distinction less severe by treating adjuncts on a par with complements (see

[5]The 'Adjuncts-as-Complements' approach has received an in-depth review by Przepiórkowski (1999a).

Przepiórkowski (1999b) and Przepiórkowski (1999a)), including their placement among valency requirements of the governor by lexical derivation. At the same time, adjuncts select their governors and properly handle semantics of the mother category, including their relative scope in case multiple adjuncts modify the same governor, by means of the *synsem*-valued attribute MODIFIED.[6]

5.1.2.3 Function words

Apart from adjuncts, an additional concern present function words, because they cannot be neatly subsumed under the patterns of adjunction or complementation.[7] With the 'binary structure' solution, additional types of syntagmatic relation is not an issue, but with the totally 'flat structure' approach a general constraint on local trees, which might be composed of both content words and function words, requires – as in the case of adjuncts – either a lexical or a relational solution.

Except for reflexive particles, function words select their hosts, and – similarly as adjuncts – they could emerge from the lexicon equipped with an attribute whose value will be identified with the *synsem* object of its head. The similarity with adjuncts extends further: more than one function word can be hosted by a single governor (e.g., an auxiliary verb and a subordinating conjunction). Therefore, it seems that a lexical solution is to be preferred again: the presence of a functional daughter in a non-lexical sign could be licensed by a slot in the valency requirements of the function word's host, derived from the host's lexical entry. Hosts of some function words would then be special in that their valency requirement for a function word is not a derived, but rather an inherent property. This is the case of, e.g., inherently reflexive verbs.

Finally, the issue of recursively embedded function words would have to be solved by leaving unspecified whether the lexical entries subject to derivation are function words or content words. Thus, an auxiliary verb can host (recursively) another. This way, the concern about a proper order of 'applying' function words to the host would be resolved and the proper values of grammateme attributes accumulated within a single immediate functional daughter.

With the organization of the lexicon and syntax sketched above, an interesting option opens up: both synthetic and analytical forms of a content word could be derived within the lexical component. For example, consider

[6]There is an objection to such a move based on conceptual grounds: lexicon should not be overloaded with purely syntactic matters. However, even when considering the status of lexicon before the introduction of adjuncts into the valency requirements, it is difficult to find a criterion by which syntactic issues should be ruled out from it.

[7]See the discussion on the role of function words in syntax in §4.4 above. The class of function words includes auxiliary and modal verbs, prepositions, subordinating and coordinating conjunctions, reflexive particles, and resumptive pronouns.

the conditional modality of the verb *bát se* 'to be afraid' expressed by the analytical form *mohla by se bát* '(she) could be afraid'. The whole form could be derived from the verb's prototypical entry as a single lexical category with a single *d-node* and four corresponding *s-nodes* (see below in §§5.2–5.3 for explanation of the terms), properly ordered within a larger expression by the general mechanism already available. This approach would nicely accommodate the traditional view of analytical morphology as being part of the same level as synthetic morphology.

Unfortunately, there are serious problems with such a purely lexical approach to analytical forms, which are manifested in constructions involving haplology of clitics, gapping in coordination, and embedding of auxiliaries. In a sentence such as (66) the reflexive particle *se* is a part of the analytical forms of both verbs.

(66) Bál jsem se usmát.
was afraid AUX RFL to smile
'I was afraid to smile.'

Whereas there are standard solutions based on syntactic valency available for treating such phenomena (Kupść, 1999), it is not clear what means could be used to eliminate the duplicates of the reflexive marker's *s-node*.[8]

In (67) the two coordinated past participle forms are content verbs, sharing a conditional auxiliary and a reflexive particle.

(67) Všichni by se tomu divili nebo smáli.
all AUX RFL it-DAT wondered or smiled
'Everybody would wonder or laugh over it.'

Again, if their analytical forms would be described as single entities in the morphological component, the resulting shape of the sentence would be hard to predict.

The final argument concerns multiple auxiliaries in a single analytical form. Examples (68a) and (68b) show that the conditional auxiliary goes with the past participle of a content word, or with the past participle of a function word.

(68) a. Přišla by.
come-PPLE COND.AUX
'She would come'
b. Mohla by přijít.
can-PPLE COND.AUX come-INF
'She could come'

[8]Incidentally, the solution to haplology of clitics proposed in §7 would be available even here, but it does not account for all necessary facts.

If all function words are derived as hosts of content words in the lexical component, all function words in an analytical form would have to be hosted directly by the content word, or their embedding would have to be specified for every such derivation. In any case, a generalization would be lost.[9]

While treating all functional morphemes by a similar morphological mechanism within the lexical component means a confrontation with a few serious obstacles, there is still the possibility of having hosts of function words specify the other components of an analytical form by quasi-valency requirements.

Conclusion: a flat structure In addition to the arguments already presented in favour of a flat structure, there is an additional reason why this structure should be preferred. The deep/surface ordering relations need access to the information about a specific semanteme on both levels. The link between the deep and the surface representation of the semanteme is available within the formal object (*sign*) representing the expression governed by the semanteme, but not within a larger object, unless the link is explicitly provided. In order to keep the deep and surface information separate, such explicit linking is not desirable. Then all semantemes participating in a local tectogrammatical tree should be available at a single point, i.e., there should be a corresponding local derivation tree which makes the semantemes available. This means that the derivation structure must be flat – at least as far as content words are concerned, all dependents (both complements and adjuncts) of a governor are sisters of the head in a single local derivation tree.

Now the question is, whether function words hosted by a semanteme should be sisters to other dependents of the semanteme. The answer is yes, although here the argument is not as conclusive, as it is only based on the aim to avoid unnecessary mismatches between the derivation and the tectogrammatical tree. As suggested above, hosts of function words will be specified for their presence by derivation within the lexical component. At the same time, the hosts' tectogrammatical and morphosyntactic properties, including agreement categories, will be open to modification by a specific function word. This will apply to all types of function words considered here: modal verbs, passive and future auxiliaries, past participle of the verb *být* as the plusquamperfect auxiliary and the past participle of the verbs *být* and *bývat* as the past conditional auxiliaries, the present indicative of *být* as the past tense auxiliary and the *by-* conditional auxiliary, subordinating conjunctions and prepositions.

In order to preserve generalizations concerning the ability of some function words to be hosted by both content and function words, recursive em-

[9]English and other languages offer plenty of other compelling examples of the embedding of auxiliaries.

bedding of function words will be allowed, which presents an exception to the otherwise flat structure.

Basic compositional constraints will be specified in §6.1.

5.2 The tectogrammatical tree

One of the primary goals of the present enterprise is to formalize the theoretical points presented in §4.1. Two possibilities will be suggested. According to the first option, the hypothesis presented in point 7 on p. 87 concerning the locality of CD ordering is hard-wired into the formal structure, preventing any non-projectivity to occur. This will be the default option, which will be – as the more restrictive hypothesis – pursued throughout the rest of this work. According to an alternative option (see footnote 6 on p. 87), the formal structure allows for non-projectivity, thus CD ordering is not necessarily tied to local trees, see §5.2.4 below.

5.2.1 Lists for brackets: the structure

The first option can be nicknamed **tree as a list of lists**. It is based on a proposal made by Sgall (1995) (also presented elsewhere) and mentioned already in footnote 25 on p. 58, whereby the tree is represented as a string of nodes and brackets. This notation has been used above in this chapter for schematic TR of example sentences.

The brackets represent the vertical order (dependency) of the tree, while the linear order of bracketed items in the string represents the horizontal order (CD). A pair of brackets encloses every subtree. Within this pair of brackets the top node of the whole tree occurs as a single unbracketed entity. If there are any nodes dependent on the top node, each of them is enclosed in a pair of brackets, similarly as the subtree above, and placed to the left or to the right of the top node, according to the horizontal order.

Coordination and apposition is represented either by a different type of brackets or – equivalently – by a different set of subscripts to the brackets. At first, only structures without coordination and apposition will be assumed.

The differences between the linear bracketed notation in FGD and its expression within the present formalism are not substantial:

1. A node is represented as a typed feature structure, rather than as a string of appended symbols.

2. Brackets as distinct symbols are not used. The content in between a pair of brackets is represented as a list. Nested brackets are treated as nested lists.

3. The specification of tectogrammatical function is treated as a feature of dependent node rather than placed as subscript to the corresponding bracket.

The tree in Fig. 5.2 is the same as the tree Fig. 3.1 on p. 59, and is repeated here for convenience.[10]

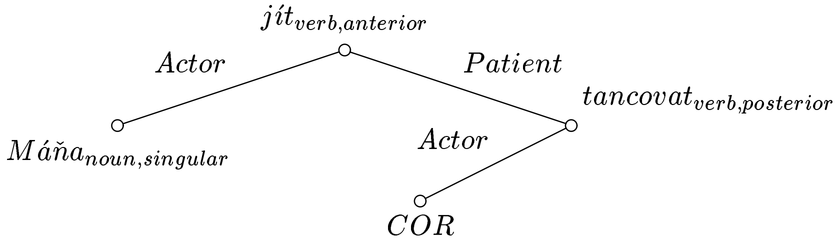


Figure 5.2: A dependency tree (same as Fig. 3.1)

The linearized tree in the standard FGD notation is shown in (69).

(69) (
$\quad$ (*Máňa*.Noun.Singular $)_{Actor}$
jít.Verb.Anterior
$\quad (_{Patient}$ (COR $)_{Actor}$ tancovat.Verb.Posterior)
)

The following structure is a simplified list-based equivalent of the above tree.[11]

(70) $\Big\langle \langle\ [Máňa]\ \rangle\ ,\ [jít]\ ,\Big\langle \langle\ [COR]\ \rangle\ ,\ [tancovat] \Big\rangle \Big\rangle$

Of all the possibilities suggested here, this format comes closest to the view of tectogrammatical representation as assumed by standard FGD:

1. Unlike the 'separate orders' approach used in the structure in Fig. 3.8 on p. 77, the 'tree as a list of lists' option leaves no place for other than node-specific information in the tectogrammatical representation and integrates the vertical and horizontal order into a single two-dimensional structure.

2. Unlike the 'tree as a flat list of nodes' option presented below in §5.2.4, the 'tree as a list of lists' option prohibits non-projectivity.

[10] In all the three representations, the symbol COR is used instead of co-indexation, which will be introduced below. The nonterminal node representing the whole sentence/utterance is omitted.

[11] In (70) angle brackets enclose a list, square brackets a feature structure, and the nodes have no internal structure. The structure of nodes at the tectogrammatical level will be introduced below.

5.2.2 The "inside" of nodes

So far, nothing has been said about the information present in labels of tectogrammatical nodes, i.e., about the internal structure of node labels. The first requirement is that they contain all and only the information necessary at the tectogrammatical level and relevant to the individual nodes. Additionally, the information should be structured in a way allowing for linguistically motivated referencing of subsets of the information.

According to the present proposal, every node is represented as an object of type *d-node* with three appropriate attributes: FUN(CTION) (specifying the node's tectogrammatical function), C(ONTEXT)B(OUND) (a binary valued feature specifying whether the node context-bound or not), and CORE. The value of the latter attribute is a complex structure of type *d-wcl*, with subtypes corresponding to tectogrammatical word classes of TR nodes. This is where the node's lexical value and grammatemes (tectogrammatical counterparts of morphological categories) are specified.

(71)
$$\begin{bmatrix} \textit{d-node} & \\ \text{FUN} & \textit{fun} \\ \text{CB} & \textit{bool} \\ \text{CORE} & \begin{bmatrix} \textit{d-wcl} & \\ \text{LEMMA} & \textit{string} \\ \dots & \end{bmatrix} \end{bmatrix}$$

The formalism requires an explicit definition of all types used, which will be provided later in §A.1. The type *fun* has maximal subtypes such as *act(or)*, *pat(ient)*, *mann(er)*, *t(ime-)sin(ce)*, corresponding to types of both inner participants (arguments) and free modifications (adjuncts).[12]

In the simplified example in Fig. 5.3, where the schematic list (70) is elaborated, only some grammatemes are present, as an illustration. The symbol *COR* for syntactic coreference is replaced by explicit co-indexation of the 'core' subparts of the relevant nodes.[13] Incidentally, in this example the remaining parts of the two nodes do not differ, but this need not be the case. In a sentence with object control of the embedded predicate and the object of the matrix clause contextually non-bound, both FUN and CB values will be different for the two nodes.[14]

[12] A more elaborate hierarchy of tectogrammatical functions can be defined: in addition to the division between arguments and adjuncts, also adjuncts related to time, space etc. can be made subtypes of an intermediate type. See §§ A.1 and C for a sample hierarchy of functions and their exhaustive listing.

[13] Co-indexation is used to express token identity of formal objects and can be replaced by equation.

[14] See the addressee of the matrix verb and the actor of the embedded verb in the following example:

(72)	Průvodčí	doporučil	střízlivým	cestujícím	si	přesednout
	conductor	advised	sober	passengers-DAT	REFL	to change seat

$$\left\langle \begin{array}{l} \left\langle \begin{bmatrix} \textit{d-node} & \\ \textsc{fun} & \textit{actor} \\ \textsc{cb} & \textit{yes} \\ \textsc{core} & \boxed{1} \begin{bmatrix} \textit{d-noun} & \\ \textsc{lemma} & \textit{Máňa} \\ \textsc{d-number} & \textit{sg} \end{bmatrix} \end{bmatrix} \right\rangle, \begin{bmatrix} \textit{d-node} & \\ \textsc{fun} & \textit{fun} \\ \textsc{cb} & \textit{no} \\ \textsc{core} & \begin{bmatrix} \textit{d-verb} & \\ \textsc{lemma} & \textit{jít} \\ \textsc{d-tense} & \textit{anterior} \end{bmatrix} \end{bmatrix}, \\ \left\langle \left\langle \begin{bmatrix} \textit{d-node} & \\ \textsc{fun} & \textit{actor} \\ \textsc{cb} & \textit{yes} \\ \textsc{core} & \boxed{1} \end{bmatrix} \right\rangle, \begin{bmatrix} \textit{d-node} & \\ \textsc{fun} & \textit{patient} \\ \textsc{cb} & \textit{no} \\ \textsc{core} & \begin{bmatrix} \textit{d-verb} & \\ \textsc{lemma} & \textit{tancovat} \\ \textsc{tense} & \textit{posterior} \end{bmatrix} \end{bmatrix} \right\rangle \end{array} \right\rangle$$

Figure 5.3: A simplified TR of *Máňa šla tancovat* as a list of lists of *d-node*s

5.2.3 Performative node

I will now return to the issue of 'performative node', which includes information about the whole sentence (see point 3 on p. 86). As in the schema of the type *sign* in Fig. 5.1 on p. 116, the value of the attribute DEEP is assumed to be of the type *d-list*. This must hold for all *sign*s. However, so far we have not seen any *d-list* which would include the performative node, or any other way of representing the information contained in it which is relevant for the sentence as a whole.

In order to do it, a solution requiring minimal modifications of the existing formal structure would consist in the introduction of a distinguished *d-node* for the performative node. The TR of a sentence (utterance) would then be formalized as a *d-list* consisting of two members: a *d-node* for the performative node, with *sent* as the FUN attribute's value, and a *d-list* for the rest of the tree.

The drawbacks of this solution stem from the fact that the performative node is very different from all other nodes in the tree, because it does not correspond to any content word. Accordingly, it has no lemma, cannot be assigned any word class, cannot be subject to coreference, DWO principles, CB/NB specification. Also, the dependency relation between the performative node and the root content word does not correspond to any 'real'

do jiného kupé.
into another compartment
'The conductor advised sober passengers to change the compartment.'
[[ACT:conductor^{cb}] advised^{nb} [ADDR:passenger$_1^{nb}$ [RSTR:sober^{nb}]]
[ACT:$\_{}_1^{cb}$] PAT:change seat^{nb} [DIR:compartment^{nb} [RSTR:another^{nb}]]]]

$$\left[\begin{array}{ll} \textit{sign} & \\ \text{PHON} & \textit{list(phonstring)} \\ \text{SYNSEM} & \left[\begin{array}{ll} \textit{synsem} & \\ \text{LOCAL} & \left[\begin{array}{ll} \textit{local} & \\ \text{CATEGORY} & \left[\begin{array}{ll}\textit{category} & \\ \text{HEAD} & \textit{head} \\ \text{VALENCY} & \textit{valency}\end{array}\right] \\ \text{DEEP} & \left[\begin{array}{ll}\textit{deep} & \\ \text{STATUS} & \textit{status} \\ \text{TREE} & \textit{d-list}\end{array}\right] \\ \text{CONTENT} & \textit{content}\end{array}\right] \\ \text{NONLOCAL} & \textit{nonlocal}\end{array}\right] \\ \text{SURFACE} & \textit{s-list}\end{array}\right]$$

Figure 5.4: The subparts of *sign* including the 'performative node'

tectogrammatical function.

Therefore, it is desirable to find a solution where the 'performative node' is treated in a way different from other nodes corresponding to 'real' semantemes. The solution consists in a slight modification of the value of the attribute DEEP. Instead of *d-list*, its type is *deep* with two appropriate attributes: STATUS and TREE. The value of STATUS specifies whether the relevant tree is embedded (e.g., governed by the main verb of the sentence) or not embedded, while the value of TREE is the tree itself. The new setup, including other subparts of *sign* is illustrated in Fig. 5.4.

5.2.4 An alternative: the tree as a flat list of nodes

In order to allow for nodes of the tree to be horizontally ordered without necessarily following the sequence of their governors (i.e., without the tree being necessarily projective) the following formal adjustments are made to the structure presented above:

1. All nodes of a tree are represented as a single list of *d-node* type objects ordered by (a modified definition of) CD.

2. Dependency relations are represented by means of an attribute specifying the node's governor. More precisely, there is a new attribute GOV(ERNOR), which is appropriate for the type *fun* and for which the appropriate value is again *fun*. A node's GOV value is its governor FUN value. Thus, any path following the GOV attributes eventually

terminates in the top node of the whole tree.[15]

Other assumptions made above in the present section §5.2 remain unchanged. The structure in Fig. 5.5 below is the adjusted counterpart of that in Fig. 5.3.

$$
\left\langle
\begin{array}{l}
\left[\begin{array}{ll}
\textit{d-node} & \\
\textsc{fun} & \left[\begin{array}{ll}\textit{actor} & \\ \textsc{gov} & \boxed{2}\end{array}\right] \\
\textsc{cb} & \textit{yes} \\
\textsc{core} & \boxed{1}\left[\begin{array}{ll}\textit{d-noun} & \\ \textsc{lemma} & \textit{Máňa} \\ \textsc{d-number} & \textit{sg}\end{array}\right]
\end{array}\right],
\left[\begin{array}{ll}
\textit{d-node} & \\
\textsc{fun} & \boxed{2}\left[\begin{array}{ll}\textit{fun} & \\ \textsc{gov} & \textit{fun}\end{array}\right] \\
\textsc{cb} & \textit{no} \\
\textsc{core} & \left[\begin{array}{ll}\textit{d-verb} & \\ \textsc{lemma} & \textit{jít} \\ \textsc{d-tense} & \textit{anterior}\end{array}\right]
\end{array}\right], \\
\left[\begin{array}{ll}
\textit{d-node} & \\
\textsc{fun} & \left[\begin{array}{ll}\textit{actor} & \\ \textsc{gov} & \boxed{3}\end{array}\right] \\
\textsc{cb} & \textit{yes} \\
\textsc{core} & \boxed{1}
\end{array}\right],
\left[\begin{array}{ll}
\textit{d-node} & \\
\textsc{fun} & \boxed{3}\left[\begin{array}{ll}\textit{patient} & \\ \textsc{gov} & \boxed{2}\end{array}\right] \\
\textsc{cb} & \textit{no} \\
\textsc{core} & \left[\begin{array}{ll}\textit{d-verb} & \\ \textsc{lemma} & \textit{tancovat} \\ \textsc{tense} & \textit{posterior}\end{array}\right]
\end{array}\right]
\end{array}
\right\rangle
$$

Figure 5.5: A simplified TR of *Máňa šla tancovat* as a flat list of *d-node*s

The 'flat list' representation of (not necessarily projective) tectogrammatical tree is based on the format for linearization grammars of Penn (1999a) and Penn (1999b). I will describe Penn's approach to linearization only very briefly, as it will be presented in more detail later in §5.3.3.2.

Penn's *domain lists* include objects corresponding to individual words in the surface string. Their order is constrained simultaneously by principles of syntax, prosody and discourse, and by their appurtenance to one of possibly many recursively embedded *word order domains* in the list, each of a certain type. These domains partition a sentence into *phenogrammatical* constituents, and result in a tree structure. This way of representing trees has been adopted in the 'tree as a flat list of nodes' version of TR sketched above.

One important difference between Penn's proposal and the latter is in the relaxation of the projectivity requirement, which is not difficult to realize as such. Penn's domain lists have to obey three formal principles in order to count as proper trees: the principles of Matrix Compaction, Relevance and Planarity. Whereas the former two principles guarantee that the

[15]Constructions involving function words may result in the need to nonmonotonically alter *d-wcl* objects, where some grammatemes of a content word do not correspond to values predicted from the content word 'on its own' with the relevant function words uninterpreted. This is the motivation behind the choice to make *fun* objects rather than entire *d-node*s appropriate as the value of *gov*.

tree is rooted and the paths ending in the root node correspond to a legitimate sequence of successively embedded domains, the principle of Planarity ensures that there are no discontinuities, i.e., that the tree is projective. This principle can be relaxed depending on specific configurations of certain kinds of nodes, in order to allow for a restricted non-projectivity, or completely withdrawn, if issues of projectivity are better solved by less general grammar constraints.[16]

5.2.5 Coordination and apposition

Coordination and apposition are syntactic phenomena which defy straightforward formalization. As pointed out above, FGD recognizes their status, involving relations different from dependency, by introducing another dimension of the tectogrammatical tree. Projected onto a two-dimensional plane, a TR node represents either a single semanteme or a coordinated/apposed (c/a) structure. A c/a structure is revealed in the third dimension.

Equivalently, c/a structures can be represented in a linearized notation by being enclosed in brackets of a different kind (or by labelling brackets), see §4.5 above. Another option is to make the formal rendering of coordination and apposition fit the standard two-dimensional dependency scheme, e.g. by treating conjunction as the governor and conjuncts as its dependents, and by making one of the apposed element dependent of the other.

The present formalism allows for either three- or two-dimensional solution. The former solution – perhaps a more theoretically plausible one – is based on the introduction of additional types into the signature. More specifically, the additional types extend the trivial classification of lists. The new subtypes of the type *list* stand either for a dependency subtree or for a kind of c/a structure.

Let me present the gist of this solution and relevant issues by using the example (65), introduced on p. 113, repeated here for convenience as (73).

(73) A duckling and several hens of our neighbour disappeared.

$[\ [\ \{\ \text{duckling}^{cb}\ \text{hen}^{cb}\ {}_{rstr}[\ \text{several}^{nb}\ ]\ \}_{conj}$

${}_{appurt}[\ [\ \text{we}^{cb}\ ]_{rstr}\ \text{neighbour}^{cb}\ ]\ ]_{act}\ \text{disappear}^{nb}\ ]$

The structure in (73) can be translated into a structure compatible with the 'tree as a list of lists' formalization as (74), using the assumptions introduced in §5.2.

(74) $\langle\ \langle\ \{\ \text{duckling}_{act},\ \text{hen}_{act},\ \langle\ \text{several}_{rstr}\ \rangle\ \}_{conj},$

$\langle\ \langle\ \text{we}_{rstr}\ \rangle,\ \text{neighbour}_{appurt}\ \rangle\ \rangle,\ \text{disappear}\ \rangle$

[16]See Penn (1999b, p. 192–196) for an RSRL formalization of the principles.

The angle brackets enclose lists of the 'dependency subtree' type, while braces enclose lists of the *c/a* type, the specific type of coordination or apposition being marked by a subscript to the right brace. The information on contextual boundness is omitted because it is not immediately relevant to the issues discussed here. In line with the proposal for a formalization of TR presented above, tectogrammatical function is a property of a node, rather than of an edge as in (73), thus there are in effect two Actor positions and two edges linking the verb governor with its coordinated Actor.

The nodes themselves should be thought of as objects of the *d-node* type.

Unfortunately, (74) is flawed in one substantial aspect: the internal structure of conjuncts in a coordinated structure is not represented unambiguously. If the order of conjuncts in (74) is reversed, as in (75), the governor of 'several' can be either 'hen' or 'duckling'.

(75) $\{\ \text{hen}_{act},\ \langle\ \text{several}_{rstr}\ \rangle,\ \text{duckling}_{act}\ \}_{conj}$

In the notation used in (73) this is not an issue: functors, which label brackets rather than nodes, always stand between the governor and the dependent.

A possible remedy is to introduce another type of list, as in (76), where the *co* type lists group nodes belonging to a conjunct.

(76) $\{\ \{\ \text{hen}_{act},\ \langle\ \text{several}_{rstr}\ \rangle\ \}_{co},\ \{\ \text{duckling}_{act}\ \}_{co}\ \}_{conj}$

Another solution would be to specify directionality of the dependency relation, either at the conjunct's governing node, or for the conjunct list type. Both solutions require additional formal specifications.

No matter which of the two solutions is chosen, another potential problem emerges. In (73), governors of the two conjuncts are not immediately accessible from the verb, but at least they can be reached by descending two levels (across the dependency bracket and the coordination bracket) and the functor, labelling the top dependency bracket, is visible at the top level.

On the other hand, in (77), showing the structure (76) in context, the functor and the conjunct's governor can be seen only after descending three levels.

(77) $\langle\ \langle\ \{\ \{\ \text{duckling}_{act}\ \}_{co},\ \{\ \text{hen}_{act},\ \langle\ \text{several}_{rstr}\ \rangle\ \}_{co}\ \}_{conj}$
$\langle\ \langle\ \text{we}_{rstr}\ \rangle,\ \text{neighbour}_{appurt}\ \rangle\ \rangle,\ \text{disappear}\ \rangle$

If, however, functors are treated as list types, they can label dependency brackets as in (73) and be immediately accessible from the governing node, see (78).

(78) $\langle\ \langle\ \{\ \{\ \text{duckling}\ \}_{co},\ \{\ \text{hen},\ \langle\ \text{several}\ \rangle_{rstr}\ \}_{co}\ \}_{conj}$
$\langle\ \langle\ \text{we}\ \rangle_{rstr},\ \text{neighbour}\ \rangle_{appurt}\ \rangle_{act},\ \text{disappear}\ \rangle$

Then, if the dependency list types are further specified for directionality, the resulting structure may look like (79), which is in fact identical to (73), with the additional advantage of being explicitly related to tools available in the present formalism.

(79) $\langle\ \langle\ \{$ duckling, hen, $_{rstr}\langle$ several $\rangle\ \}_{conj}$
$_{appurt}\langle\ \langle$ we $\rangle_{rstr}$, neighbour $\rangle\ \rangle_{act}$, disappear $\rangle$

Assuming the extension of signature by an appropriate hierarchy of list types for TR functions and *c/a* constructions, and elimination of the attribute FUN from the type *d-node*, the solution illustrated by (79) seems to correspond to the theoretical treatment of coordination very closely.

However, a question may be raised whether the typed list is a proper representative of a *c/a* construction, especially in view of the fact that there is no other information beyond the type of *c/a* construction available at the level of the list. It should also be noted that this *c/a* type list (*conj* in (79)) is the only location where any information about the *c/a* construction as a whole can be specified. If any such information beyond the type of *c/a* construction is required, it has to be derived from its individual conjuncts.

This issue is not so much a question of an appropriate formalization as a theoretical problem. Arguments can be found in favour of treating *c/a* constructions as loosely restricted collections of elements, making the list type an adequate structure. For example, in (81) the verb can agree either with the first conjunct or with the coordination as a whole.[17]

(81) a. Přišla / *Přišel / Přišli Helena a Petr.
came-SG,FEM./SG,MASC./PL Helena and Petr
'Helena and Petr arrived.'

b. *Přišla / Přišel / Přišli Petr a Helena.
came-SG,FEM./SG,MASC./PL Petr and Helena
'Petr and Helena arrived.'

For a proper agreement licensing, the access to the individual conjuncts is important and the representation of coordination as a list of conjuncts is appropriate. However, the relevance of this argument for TR is questionable, because agreement is not a tectogrammatical phenomenon.

A more compelling case can be made on the grounds of the possibility to coordinate 'unlike categories', such as those in (82).

(82) Tom is a party member and proud of it.

[17] This option is available only if verb precedes subject:

(80) a. Helena a Petr *nepřišla / *nepřišel / nepřišli.
b. Petr a Helena *nepřišla / *nepřišel / nepřišli.

Since *d-nodes* are specified for word class, it is impossible to identify a *species* of word class to cover both conjuncts.[18]

But there may be reasons why the properties of *c/a* constructions should have the same form as those of other syntagms. One of them is the general interchangeability with non-*c/a* constructions. Another reason is the fact that the properties of a coordinated construction as a whole may be referenced from other nodes within the tree, possibly from multiple nodes. Since this may actually involve pronominal or other type of co-reference, this case is relevant for TR. Even if properties of the coordinated structure are determined by the conjuncts' values, their explicit presence may be desirable at least in the case of multiple referencing nodes for reasons of economy.

An alternative solution, which allows for grouping together richer information about the whole *c/a* construction in a single *d-node* is very straightforward and has often been used before as a 'technical' substitute for the 'more theoretically plausible' 3D approach. In syndetic and polysyndetic constructions, it is based on treating the conjunction as a regular TR citizen and the governor of the construction. Thus, the node corresponding to the conjunction is the representative of the whole coordinated structure.

In apposition pairs, one of the members is the governor, the other the dependent.

In (83) our example sentence is represented with conjunction as the governor (see also Fig. 4.2 on p. 114).

(83) $\langle\ \langle\ \langle\ \text{duckling}_{co}\ \rangle,\ \text{and}_{act},\ \langle\ \text{hen}_{co},\ \langle\ \text{several}_{rstr}\ \rangle\ \rangle,$
$\langle\ \langle\ \text{we}_{rstr}\ \rangle,\ \text{neighbour}_{appurt}\ \rangle\ \rangle,\ \text{disappear}\ \rangle$

The issue of selecting one or the other solution will be left unresolved at this point.[19]

5.3 The surface string

This section starts with an overview of FGD's stance on the level sensitive to surface word order (the level of morphemics). Next, I will provide foundations for integration of the tree-string relation into the present framework, reflecting the theoretical position. This part will be followed by two alternative formal solutions proposed for specifying surface word order within

[18] There may be two solutions to this problem. The first involves 'restoration' of larger coordinated segments, VPs or clauses in the present case: 'Tom [is a party member and is proud of it]. The second solution is based on allowing more general categories, other than species, as values for coordinated structures.

[19] An overview of coordination issues in Polish, a language closely related to Czech, together with proposals of solutions to some of them can be found in Anna Kupść and Małgorzata Marciniak and Agnieszka Mykowiecka (2000). A rigorously defined extension of TR for coordination was presented in Kučerová (1995), together with examples from Russian.

HPSG. Finally, I will introduce a way of constraining the surface string within the present framework.

5.3.1 Theoretical assumptions

The main points of FGD concerning the level of morphemics, which are relevant to the subsequent proposal of its formal rendition, can be summarized as follows:

1. Unlike representations at the level of tectogrammatics, representation of expressions at the level of morphemics is one-dimensional, directly corresponding to the temporal or spacial order of strings of phonemes or graphemes, respectively. In the following, the order of items at this level will be called **surface order**, in contrast with **deep order**, the horizontal ordering of nodes of the tectogrammatical tree.

2. Function words are treated as citizens equal to content words and can be subjected to the same ordering regularities.

3. In the default case, the order of content words is the same at both levels.

4. The default placement of function words is left-adjacently to the content words they modify. If more than one function word modifies a content word, their position relative to the content word and to other function words is subject to constraints specific to their categories.

5. The default positions can be overridden in the standard version of FGD by the operation of movement rules.[20] Movement rules reorder either content words in cases where the surface word order differs from its deep counterpart,[21] or function words in cases where they occur non-adjacently to the content words they modify.[22]

The involvement of movement rules makes the resulting tree-string relation rather complex: not only is every 'deep node' related to a (possibly empty) string of 'surface items', preserving the default deep order among the

[20]Movement rules in FGD have a slightly different flavour that those used in the GB theory. Whereas transformations and movement rules in GB operate on the s-structure, i.e., on a tree, movement rules in (the recent versions of) FGD operate during the transition between the tectogrammatical tree and the morphemic string. In both theories, movement rules need access to structural information, but there is no structure beyond linear order available at the FGD level of morphemics. Also, morphemic string is specified only for a complete utterance (corresponding to its tectogrammatical tree), rather than for every part of the utterance (corresponding to the subtrees).

[21]For a list of types of DWO/SWO mismatches and some examples see p. 99.

[22]For a list of types of function words positioned non-adjacently with respect to their hosts see p. 105.

strings, but at the same time all those items may be reordered by movement rules, taking into account both the tectogrammatical tree and the items in the string.

5.3.2 Main features of the present approach

The present proposal represents an attempt to make the tree-string relation at the same time more modular and compatible with the adopted formalism.

In the following, its features will be pointed out, highlighting the differences between the standard FGD treatment of the level of morphemics and its relation to the tectogrammatical tree on the one hand and the present proposal on the other.

1. The specification of the level of morphemics, an object of type *s-list* as the value of attribute SURF(ACE), is present in the representation of every expression, including lexical categories and all their projections.

2. All constraints concerning the level of morphemics, including those relating it to the tectogrammatical tree, apply to the representation of every expression.

3. Function words are licensed by lexical specifications and the general mechanism for combining syntactic components, while their position in the surface ordering is constrained again by a general mechanism governing the surface order of all items.

4. The two types of constraints on surface word order – the identity relation with deep word order for some content words and the reordering regularities for other cases – are separate. The items in an *s-list*, objects of type *s-node*, are required to include the information necessary for imposing an appropriate order. In the case of a content word obeying the default deep order, this information lets the *s-node* equivalent of the content word be ordered according to the deep order. In cases where the deep order is not preserved, the *s-node* to be dislocated includes information about a position which it is supposed to occupy.[23]

[23] In fact, the formalism adopted here does not permit the use of default values or default constraints. The term 'default' is used here as a metaphor for regularities which would need no additional treatment in a formalism using movement rules. More specifically, these regularities concern cases where movement rules do not apply and the surface order of content words corresponds to the horizontal order of relevant terminal nodes in TR with function words positioned as described in point 4 on p. 133. In this aspect, a formalism using movement rules scores better than a formalism which does not allow for defaults (P. Sgall, p.c.). In the present formalism, such regularities must be made explicit in the same way as any other regularity, i.e., by a constraint applying to relevant cases. A constraint which simulates this aspect of behaviour of a formalism using movement rules will be metaphorically called 'default constraint'.

5. Due to the fact that every non-lexical *s-list* is composed from other *s-list*s, every *s-list* can either be specified as necessarily continuous within all larger *s-list*s or allowed to occur discontinuously within them.[24] In order to make such discontinuities possible, the way the mother's *s-list* is composed differs from the way its *d-list* is composed. A *d-list* is composed by inserting entire non-head daughters' *d-list*s into the head daughter's *d-list*, which makes *d-list*s equivalent to projective trees. However, in the case of *s-list*s, a more relaxed relation allows for individual items in the daughters' *s-list*s to be 'shuffled' within the mother's *s-list*, provided that the relative order of items within the *s-list* of each daughter is preserved in the mother's *s-list* and the items are allowed to occur discontinuously.[25] In conjunction with constraints using surface-positional information in the items, this relation among *s-list*s of a non-lexical category allows and properly constrains expressions involving discontinuities.

6. In effect, the surface-positional information consisting of the specification of the item's position in an *s-list* together with the specification of the continuous *s-list* in which it is required to occur means that the representation of morphemic string is in fact a more structured object. For example, components of a main clause can be identified as parts of contiguous non-overlapping strings, each of which corresponds to some distinguished word order positions: initial field, second position clitic cluster, rest field. These subfields can be further subdivided: the second position clitic cluster into fields corresponding to the different kinds of clitics according to their ordering requirements, the other fields can include subfields corresponding to contiguous nominal, prepositional, adjectival groups, within which the surface word order regularities can be stated. Although this recursive classification of items can be diagrammed as a tree structure, expressing the surface word-order constituency (following Penn (1999b), a *phenogrammatical tree*), components of the expression are required to obey word order constraints applied in a monotonous way.

[24] Imposing continuity on the items of an *s-list* is equivalent to the notion of *compaction* of *domain objects* in Kathol (1995), Penn (1999a) and others within the framework of *linearization grammars*, see §5.3.3 below. It is easy to see that *s-list*s can be compared to their *word order domains*, i.e., lists of objects ordered according to the surface word order of their phonological yields. Thus, items in a *s-list* may be specified as constituting a continuous string. However, an item may also be unspecified in this respect, or specified as part of any larger *s-list*, up to the top *s-list*, corresponding to the whole utterance. This would allow such an item to be 'moved' out of its original *s-list* (using the procedural terminology for a moment) and placed at the specified position.

[25] This relation is equivalent to the relation of `shuffle` or `sequence-union`, proposed first by Reape (1994) and elaborated later by Kathol (1995), Penn (1999a), see §5.3.3 below.

5.3.3 Formal accounts of linearization

It is time now to concentrate on the setup of *s-list*. There are two basic options. The list can either be structured in such a way that items required to occur continuously are embedded in objects representing the continuous strings, so that the continuously occurring items are not visible at the higher level. The surface-positional information relevant for an item is partly implicit in the fact what is the object 'hiding' the item. This is the approach suggested for similar structures (word order domains) within the context of *linearization grammars* by Reape (1994) and Kathol (1995).

The other approach, proposed by Penn (1999a) for the treatment of word order in Serbo-Croatian, is based on a flat list of elementary items. Each item specifies its position in the 'phenogrammatical tree' by a path of word order regions, starting from the lowest position ('field') of the item itself within the immediate region and terminating in the highest region: the matrix clause. At the cost of additional constraints making this kind of structure possible, this approach has the advantage of being able to monotonically absorb constraints on surface word order coming from various sources: syntax, discourse, and prosody.

5.3.3.1 Linearization according to Kathol (1995)

The idea of using a separate level for the linear order of words instead of using the linear order of terminal nodes of a syntactic tree is far from new, but it was Reape (1994), who presented the first realistic proposal for integrating such an approach with a constraint-based framework. The proposal was later refined by Kathol (1995).[26]

The approach employs an attribute DOMAIN, appropriate to *sign*s, which encodes linear order of items associated with the sign, i.e., its order domain. Its value is a list of *domain-object*s:

(84) $\begin{bmatrix} \textit{domain-object} & \\ \text{PHONOLOGY} & \textit{list(phon-string)} \\ \text{SYNSEM} & \textit{synsem} \\ \text{TOPOLOGY} & \textit{field} \end{bmatrix}$

A domain object includes information on the associated string (phonology), its *synsem* value, and the attribute TOPOLOGY, specifying position of the string within a larger string. The *synsem* value is lexically specified as being identical to the word's synsem. Similarly, the *field* value may be fully specified in the lexicon, or underspecified in the lexicon in order to be fully specified by the grammar, allowing for multiple placement possibilities. The order of domain objects corresponds to the order of words associated with

[26] See ibid. for an overview of grammars based on a separate linearization component.

the sign by identifying the value of the sign's phonology with concatenated phonologies of the domain objects:

$$(85)\quad sign \rightarrow \begin{bmatrix} \text{PHONOLOGY} & \boxed{1} \oplus \ldots \oplus \boxed{n} \\ \text{DOMAIN} & \left\langle \begin{bmatrix} \textit{domain-object} \\ \text{PHONOLOGY } \boxed{1} \end{bmatrix}, \ldots, \begin{bmatrix} \textit{domain-object} \\ \text{PHONOLOGY } \boxed{n} \end{bmatrix} \right\rangle \end{bmatrix}$$

If the mother's list of domain objects were an append of those of its daughters, no discontinuity could arise: the list would consist of continuous sequences corresponding to the constituents. In order to allow discontinuities at a minimal cost in terms if formal complexity, the daughters' lists can be interleaved while preserving the order of members in each of the daughter's list within the mother's list.[27] The relation which interleaves two lists in this way is called sequence union (Reape, 1994) or shuffle (Kathol, 1995) and – given free order of domain objects in the 'lower' lists – allows any permutation of domain objects (the relation `shuffle/3` is defined in §A.3).

Obviously, this measure of word-order freedom is not always needed, even in languages exhibiting the so-called free word order. In FGD, the preference for continuous surface realization is one of the consequences of the unmarked case of identity of deep and surface word order, while deviations from the identity are considered secondary and conditioned by a marked context. This is in contrast to theories where continuity is a result of constituency structure.

Kathol explicitly distances himself from constituency-based approaches to word order. In order to specify continuous sequences of domain objects, a list consisting of such objects is not shuffled with other lists. Instead, a list whose domain objects are supposed to form a continuous sequence occurs in the mother's domain list as a single domain object. The relation responsible for 'packing' multiple domain objects into a single domain object is called compaction. In a toy Czech grammar, nominal groups could be compacted while embedded infinitival clauses could be shuffled.

Fig. 5.6 on p. 138 shows schematically a part of the derivation of a simple German clause (possibly a question) *liest Hans das Buch*. Even though the derivation is binary-branching and the VP *liest das Buch* is split by the subject, there is still a constituent corresponding to the VP in the derivation tree.

First, the verb *liest* combines with an NP *das Buch* into a VP, where the NP appears necessarily as a continuous string. Therefore, the two domain objects for *das* and *Buch* are compacted into a single one for *das Buch*. However, when the VP is combined with the subject NP, the domain objects are

[27]Given a head daughter with the list consisting of elements a, b, c and a non-head daughter consisting of elements d, e, f, the mother's list can be d, a, b, e, c, f, but not a, e, b, d, c, f.

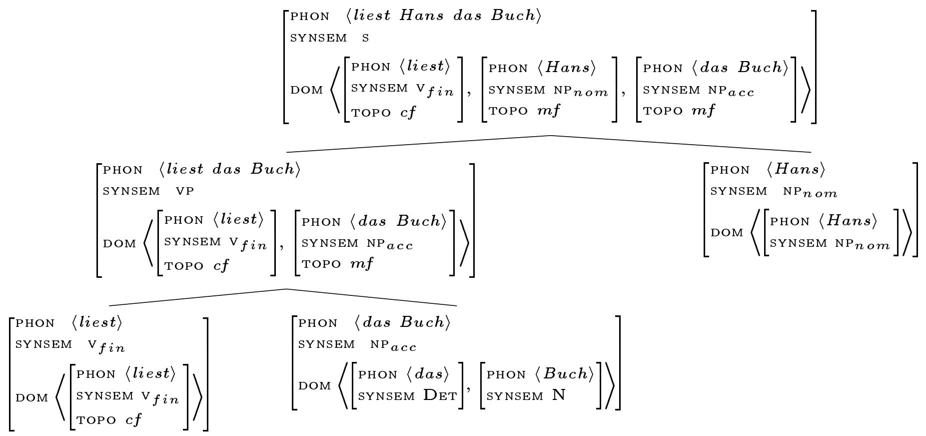


Figure 5.6: A tree showing the setup of Kathol's domain objects

shuffled rather than compacted, which allows for the intended discontinuous sequence.

Note that the synsem value of the compacted domain object for *das Buch* is identical to that of the corresponding constituent, while its phonology consists of concatenated phonologies of the constituent's domain objects. The values of the TOPO attributes stand for the terms traditional in German syntax: *vf* (Vorfeld = the initial position), *cf* (linke Satzklammer = Comp), *mf* (Mittelfeld = middle field), *vc* (rechte Satzklammer = verb cluster), *nf* (Nachfeld = extraposition field). These values can be used by linear precedence (LP) constraints, each a relation whose conjunction with all other LP constraints defines a general ordering relation `order_constraints/1` which must be satisfied by the domain list of all signs.

As it stands, the compaction relation takes care of continuous strings associated with a constituent daughter. In order to handle cases of extraposition of PP from NP, where compaction of only some of the domain objects is needed, Kathol (1995, §8.3.2) introduces the relation of partial compaction, whereby designated domain objects (specified as an additional argument of the relation) are shuffled into a higher domain, while the remaining elements are compacted. In an example (see ibid.), the immediate daughters of a VP *einen Hund füttern der Hunger hat* are an NP *einen Hund der Hunger hat* and a V *füttern*. Now instead of compacting the whole NP into the VP, there are some "designated domain objects", here a single object for the relative clause, which do not compact with the rest of the NP, and are free to occur clause-finally by virtue of their topological field specification of *nf* (Nachfeld).

Unfortunately, this solution has the unpleasant consequence of retaining the synsem value of the whole NP *einen Hund der Hunger hat* only for the compacted part *einen Hund*. The same problem can be seen also in Kupść (2000, §2.4.2), where partial compaction is used for PPs in Polish: the preposition and the first domain object of the following NP are compacted, while the rest of the PP is shuffled – see Fig. 5.7.[28]

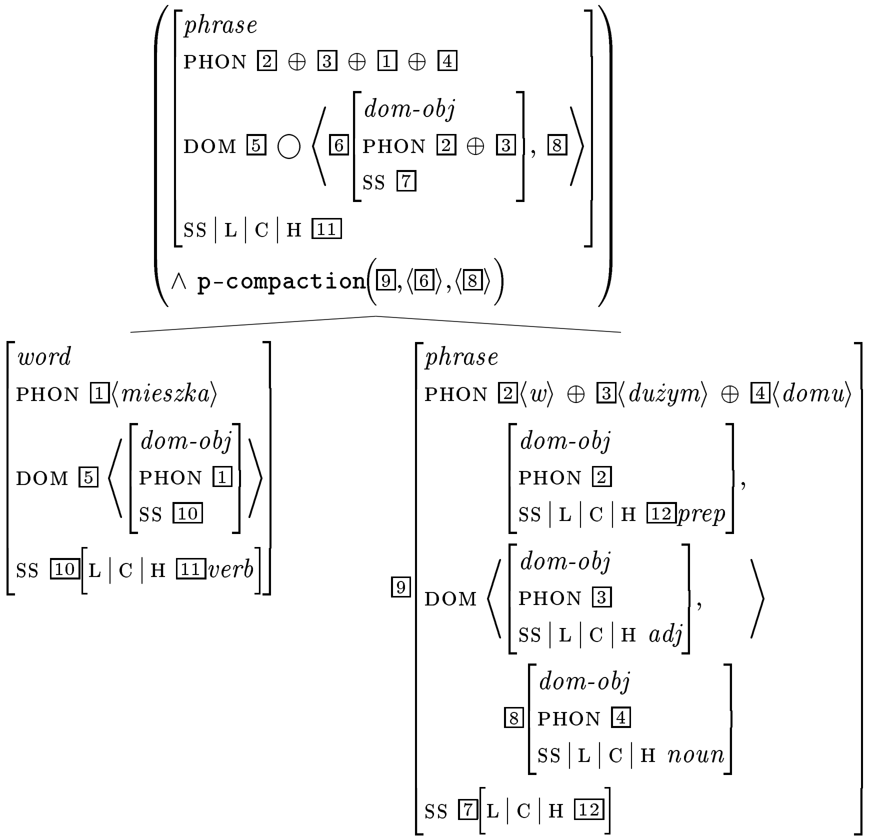


Figure 5.7: Partial compaction of preposition and adjective – adapted from Kupść (2000, §2.4.2)

In the sign for the Polish VP *mieszka w dużym domu* 'lives in a big house' the domain object for the verb *mieszka* (the only member of $\boxed{5}$) is shuffled into a list consisting of two domain objects, $\boxed{6}$ and $\boxed{8}$, two parts of the PP, which can be split as in *w dużym mieszka domu*.[29] Note that the domain object $\boxed{6}$ (preposition+adjective) does not correspond to any constituent (the daughters of the PP *w dużym domu* are the preposition and the NP). However, its synsem value ($\boxed{7}$) is identical to that of the whole PP, which is not correct.

[28] The paths SYNSEM|LOCAL|CATEGORY|HEAD are abbreviated as SS|L|C|H.

[29] In the domain list of the mother category, the domain object for *mieszka* is shuffled with a list consisting of the two other domain objects for *w dużym* and *domu*. One possible result is that identified by the mother's phonology, which is the string *w dużym mieszka domu*.

5.3.3.2 Linearization according to Penn (1999)

Penn's approach is motivated by the aim to adequately describe word-order facts in Serbo-Croatian, where prosody and discourse have been found to interact with syntax in constraining word order. While retaining the list of domain objects as the locus for constraining word order as the result of the various factors, the relatively minor role of syntactic structure in determining word order in Serbo-Croatian has led Penn to disassociate the domain objects from constituency structure. Thus, there is no SYNSEM attribute in a domain object:

(86) $\begin{bmatrix} \textit{domain-object} & \\ \text{PHONOLOGY} & \textit{list(phon-string)} \\ \text{COMPACTION} & \textit{field} \end{bmatrix}$

This means that the issue of 'false' synsems in partially compacted domain objects cannot arise. As in Kathol (1995), the phonology of a sign is identical to the append of the phonologies of its domain objects, in the order they appear in the sign's domain list. However, compaction of domain objects is performed in a different way. The value of the attribute COMPACTION plays a double role: it identifies the object's field, and a region in which the object is compacted with other objects. This is possible by the following signature of the type *field*:

(87) *field*

matrix	
pre-cf	REGION *field*
cf	REGION *field*
post-cf	REGION *field*
rf	REGION *field*
ef	REGION *field*

I will show first how the field of the object is specified. The subtypes of *field* identify the object's position within the sentence by specifying its field: the whole sentence (*matrix*), the initial (pre-clitic) field (*pre-cf*), the clitic field (*cf*), the post-clitic field (*post-cf*), the 'rest' field for all other objects (*post-cf*), and the field for an embedded clause (*post-ef*).[30] The value of COMPACTION is either *matrix*, without any attributes, or any of the other types, which require the attribute REGION. The value of the latter is again a *field* and it specifies a larger field (a 'region'), in which the field occurs. If the value of COMPACTION is not *matrix*, then the path composed by appending all REGION values of the recursively embedded *field*s eventually terminates by *matrix*. This is what imposes upon the flat list of domain objects a kind of tree structure with nodes labelled by recursively embedded *field* types.

[30]For a realistic grammar, many more fields are needed.

Compaction is effected by compacted domain objects sharing their *field* values. Consider Fig. 5.8.[31]

$$\left\langle \begin{bmatrix} \textit{dom-obj} \\ \text{PHON } \langle \textit{Ivan} \rangle \\ \text{C} \begin{bmatrix} \textit{pre-cf} \\ \text{R } \boxed{0}\ \textit{matrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{dom-obj} \\ \text{PHON } \langle \textit{mi} \rangle \\ \text{C} \begin{bmatrix} \textit{dat-cf} \\ \text{R } \boxed{1} \begin{bmatrix} \textit{cf} \\ \text{R } \boxed{0} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{dom-obj} \\ \text{PHON } \langle \textit{je} \rangle \\ \text{C} \begin{bmatrix} \textit{je-cf} \\ \text{R } \boxed{1} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{dom-obj} \\ \text{PHON } \langle \textit{rekao} \rangle \\ \text{C} \begin{bmatrix} \textit{rf} \\ \text{R } \boxed{0} \end{bmatrix} \end{bmatrix}, \\ \begin{bmatrix} \textit{dom-obj} \\ \text{P } \langle \textit{da} \rangle \\ \text{C} \begin{bmatrix} \textit{pre-cf} \\ \text{R } \boxed{2} \begin{bmatrix} \textit{ef} \\ \text{R } \boxed{0} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{dom-obj} \\ \text{PHON } \langle \textit{voli} \rangle \\ \text{C} \begin{bmatrix} \textit{rf} \\ \text{R } \boxed{2} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{dom-obj} \\ \text{PHON } \langle \textit{Mariju} \rangle \\ \text{C} \begin{bmatrix} \textit{rf} \\ \text{R } \boxed{2} \end{bmatrix} \end{bmatrix} \right\rangle$$

Figure 5.8: A domain list according to Penn (1999)

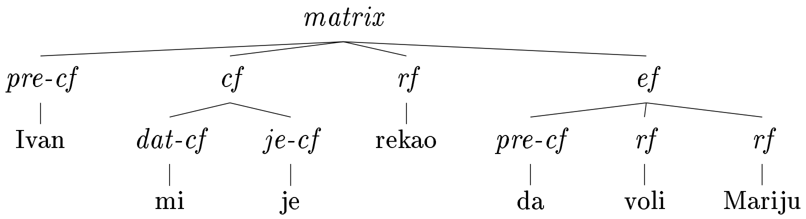


Figure 5.9: The structure of topological fields in Fig. 5.8

The highest region of all domain objects in the sentence *Ivan mi je rekao da voli Mariju* 'Ivan told me that he loves Mary' is *matrix*. The region *matrix* consists of the field *pre-cf* for *Ivan*, the region *cf* for the 2nd position clitics, the field *rf* for the rest of the main clause, and the region *ef* for the subordinate clause. All these fields and regions compact to *matrix*, which is shown by co-indexing the most deeply embedded REGION attribute ($\boxed{0}$). The two clitics compact to *cf* (note the coindexing by $\boxed{1}$), while being distinguished by their different fields *dat-cf* and *je-cf*. The verb *rekao* is the only inhabitant of the main clause's field *rf*. The domain objects belonging to the subordinate clause compact to *ef* ($\boxed{2}$). There is the complementizer *da* in the subordinate clause's *pre-cf* field and two inhabitants of the subordinate clause's *rf* field: *voli* and *Mariju*. Note that they only compact to the subordinate clause, not to the rest field.

Of course, in addition to compaction, the domain objects are subjected to constraints specifying their relative order by reference to the field/region values. Here it is important to order objects according to fields at a proper

[31] The attributes COMPACTION and REGION are abbreviated as C and R, respectively.

level. Penn uses the term *relative topological fields* (henceforth RTFs), which are defined for every two objects on a domain list as the types of the deepest non-identical structures in the objects' COMPACTION values whose REGION values are identical. If the COMPACTION values are the same, then RTFs are of that type.

Thus, RTFs of *da* and *voli* are *pre-cf* and *cf*, respectively. RTFs of *rekao* and *da* are *rf* and *ef*. Linear precedence (surface order) is determined by requiring that each pair of domain objects on a domain list is ordered in accordance with the ordering specified for their RTFs. The use of RTFs in the imposition of linear precedence allows to order domain objects not only by their 'terminal' field values, but also according to regions. In effect, domain objects are properly ordered in all regions. They behave as if they were parts of recursively embedded *phenogrammatical constituents*, forming a tree rooted in *matrix* with nodes corresponding to the regions and fields as the terminal nodes. The notion of RTFs can then be viewed as allowing to order sister constituents within such a tree.

In the simple grammar, the order of fields which compact to *matrix* or *ef* is required to be *pre-cf, cf, post-cf, rf* and *ef*. Only the *pre-cf* field is obligatory, the other fields except *rf* may appear at most once. Similar constraints apply to fields which compact to *cf*.

There is an additional concept of *field relative to region*. This is the field of a domain object relative to other objects in the same region. Thus, the field of *mi* relative to the region *cf* is *dat-cf*, while the field of the same domain object relative to the region *matrix* is *cf*.

Using the flat list structure for ordering and compacting domain objects allows to monotonically add constraints originating in different levels of grammar: syntax, prosody and discourse. Penn factors out the effect of constraints originating in prosody and discourse from syntactic constraints by using separate structures specifying prosodical and discourse constituency, where the constituents are equipped with their own domain lists indicating order and compaction. Thus, the compaction of an example PP *u lepi grad* in the *rf* region is the result of monotonically adding the information on the prosodical compaction of *u* and *lepi* and on the syntactic compaction of *lepi* and *grad*. If compaction were effected by creating a new domain object as in (Kathol, 1995), conflicting constraints on the PP would stand in the way of a correct result.

The principles and relations used for Penn's Serbo-Croatian examples are described with all formal details in RSRL in Penn (1999b, p. 192–196). See also Richter (2000, p. 336–350) for a discussion of their properties and their formulation in the canonical RSRL notation. Some of them are presented with only minor modifications as parts of this framework below in §6.4 and §A.3.

5.3.4 Formalization of surface word order

There are at least two reasons for adopting Penn's approach to linearization in preference to Kathol's. Firstly, since Penn's domain objects do not include synsem values, there is no issue whether they correspond to the phonology of a sign. Secondly, the factors responsible for word order phenomena Czech also seem to have origins in various levels, similarly as in Serbo-Croatian. On the other hand, Kathol's approach has the advantage of being simpler and better studied. Nevertheless, I will adopt a modification of Penn's approach for the formalization of surface word order.

In most cases, there will be direct parallels. The main difference is in simplifying the representation of the interacting kinds of constraints: there are no separate objects representing prosodical and discourse constituency, i.e., no lists of domain objects indicating compaction only due to prosodical or discourse factors. Constraints of these types are applied directly to the single domain list, the *s-list*.[32] On the other hand, the list expressing surface order must be related to deep word order within the corresponding *d-list*, which does include discourse-related information.

The solution is based on two hypotheses: (i) The deep/surface order relation can be stated without duplicating the 'deep' information in the surface structure or the 'surface' information in the deep structure. (ii) Other constraints on surface order can be stated by using only the *field* value. These two hypotheses enable to postulate a parallel to Penn's domain object, which does not include other information beyond phonology and its field.

Yet there are two modifications: (i) There is one property which seems to be appropriately attributed to the proposed parallel of Penn's domain object, namely a rudimentary representation of sentential stress by means of a binary-valued attribute I-CENTRE. The main reason for its inclusion inside the domain object is its interaction with surface word order. (ii) In order to specify the domain object corresponding to the lexical head of a sign, the *head* object of each sign has an additional attribute (S-NODE) whose value is co-indexed with the corresponding domain object.

Every item in the *s-list* is represented as an object of type *s-node*, see (88).

(88)
$$\begin{bmatrix} \textit{s-node} & \\ \text{PHONOLOGY} & \textit{list(phonstring)} \\ \text{FIELD} & \begin{bmatrix} \textit{field} & \\ \text{REGION} & \textit{field} \end{bmatrix} \\ \text{I-CENTRE} & \textit{boolean} \end{bmatrix}$$

The role of the two attributes appropriate to the type *s-node* is explained

[32] There is no fundamental reason for not adopting such separate domain lists. Admittedly, the main reason is that the issue would go beyond the scope of this work.

below:

- PHON(OLOGY) – Specifies the item's contribution to the list of 'phonological' strings of the whole expression. The value is a list, therefore a single item may correspond to more than one string.

- F(IE)LD – Specifies the item's position and the possibility to form continuous sequences with other items in the utterance in terms of its *field* and *region*(s). The type *field* has subtypes such as pre-clitic field (*precl-fld*), clitic field (*cl-fld*), rest field (*rest-fld*), and matrix field (*matrix-fld* – for the field covering whole utterance). The attribute R(EGION) is appropriate for every subtype of *field* except *matrix-fld*, the value of this attribute is again of type *field* and specifies the field (or 'region') of which the current filed is a part. See (89) for an example of how the position of the Czech reflexive particle *se* is specified within the whole (matrix) clause.[33]

(89)
$$\begin{bmatrix} \textit{s-node} & \\ \text{PHONOLOGY} & \langle \textit{se} \rangle \\ \text{FIELD} & \begin{bmatrix} \textit{cl-rfl-fld} & \\ \text{REGION} & \begin{bmatrix} \textit{cl-fld} & \\ \text{REGION} & \textit{matrix-fld} \end{bmatrix} \end{bmatrix} \\ \text{I-CENTRE} & \textit{no} \end{bmatrix}$$

A continuous ('compacted') string of *s-node*s is represented by co-indexing. If, for example, a nominal group is supposed to occur continuously within any larger *s-list*, its own *s-list* would look like that in (90).[34]

(90)
$$\left\langle \begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{černý} \rangle \\ \text{FLD} & \begin{bmatrix} \textit{l-adj-fld} & \\ \text{R}\ \boxed{1} & \begin{bmatrix} \textit{noun-fld} \\ \text{R}\ \textit{field} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{vlk} \rangle \\ \text{FLD} & \begin{bmatrix} \textit{noun-fld} \\ \text{R}\ \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle$$

In this example, the higher region is not specified (note the unspecific type *field*). The information becomes available within a sign combining

[33] There are several subtypes of *se* as a reflexive particle, but as a clitic they all behave in the same way, so they can all be assumed to occupy the position specified by the field type *cl-rfl-fld*. Note that reflexive particles are not part of the intonation centre.

[34] Note that some attribute names are abbreviated and that – at least in the unmarked case – the deep order would be reversed. The attribute I-CENTRE is omitted here and below: its value cannot be made specific without context.

the nominal group with other strings into a larger expression, possibly a clause. In this case, the higher region would be either *precl-fld* or *rest-fld*, themselves being a part of *matrix-fld*, see (91).

(91)
$$\left\langle \begin{bmatrix} \textit{s-node} & \\ \textsc{phon} & \langle \textit{černý} \rangle \\ \textsc{fld} & \begin{bmatrix} \textit{l-adj-fld} & \\ \textsc{r} & \boxed{1} \begin{bmatrix} \textit{noun-fld} & \\ \textsc{r} & \begin{bmatrix} \textit{rest-fld} \\ \textsc{r}\ \textit{matrix-fld} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} & \\ \textsc{phon} & \langle \textit{vlk} \rangle \\ \textsc{fld} & \begin{bmatrix} \textit{noun-fld} \\ \textsc{r}\ \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle$$

The FIELD value of the *s-node* for *černý* in (90) and (91) is still simplified. In order to impose a proper order within the regions governed by adjectives and adverbs, additional fields are introduced, see (92). The setup of the region *np-fld* is shown in Table 6.4 on p. 187.

(92)
$$\left\langle \begin{matrix} \begin{bmatrix} \textit{s-node} & \\ \textsc{phon} & \langle \textit{velmi} \rangle \\ \textsc{fld} & \begin{bmatrix} \textit{adv-fld} \\ \textsc{r}\ \boxed{2} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} & \\ \textsc{phon} & \langle \textit{černý} \rangle \\ \textsc{fld} & \begin{bmatrix} \textit{l-adj-fld} & \\ \textsc{r} & \boxed{2} \begin{bmatrix} \textit{l-adj-fld} \\ \textsc{r}\ \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \\ \begin{bmatrix} \textit{s-node} & \\ \textsc{phon} & \langle \textit{vlk} \rangle \\ \textsc{fld} & \begin{bmatrix} \textit{noun-fld} & \\ \textsc{r} & \boxed{1} \begin{bmatrix} \textit{noun-fld} & \\ \textsc{r} & \begin{bmatrix} \textit{rest-fld} \\ \textsc{r}\ \textit{matrix-fld} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{matrix} \right\rangle$$

- I-CENTRE The value of this attribute is positive, iff the *s-node* corresponds to an item which is part of the intonation centre of an utterance. Intonation centre is assumed to correspond to focus proper, i.e., the final item of a *d-list*, if contextually non-bound. Multiple *s-nodes* within an *s-list* may be I-CENTRE-positive.

The example *s-list* in Fig. 5.10 assumes that the matrix clause consists of a pre-clitic field (*precl-fld*), a clitic field (*cl-fld*) and a rest field (*rest-fld*). Of the three, only the latter is obligatory. The clitic field is further divided into subfields corresponding to different types of clitics, which are required to occur in a certain order within the cluster of clitics. Most other fields, such as the noun group field (*np-fld*), can occur as subfields either in the preclitic field or in the rest field. There is a specific order requirement within some fields, overriding the deep word order: clitics in a matrix clause have to follow the pre-clitic field and precede the rest field. Adjectives in a noun group field tend to precede the noun field (this does not hold in specific cases), preposition in a prepositional group field has to precede everything else. Some fields can be specified lexically (clitics). In order to specify other fields, constraints must be applied to nonlexical signs.[35]

This way, continuity is specified (*compaction* in Kathol's and Penn's terms). Whenever items should form a continuous field, their field specification is co-indexed. This has the same effect as grouping them into an object in the approaches of Reape and Kathol.

$$\left\langle \begin{array}{l}
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{mohla} \rangle \\ \text{F} & \begin{bmatrix} \textit{pre-cl-fld} \\ \text{R}\ \boxed{1}\ \textit{mtx-fld} \end{bmatrix} \\ \text{I-C} & \textit{no} \end{bmatrix},
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{by} \rangle \\ \text{F} & \begin{bmatrix} \textit{cl-be-fld} & \\ \text{R} & \boxed{2} \begin{bmatrix} \textit{cl-fld} \\ \text{R}\ \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{I-C} & \textit{no} \end{bmatrix},
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{se} \rangle \\ \text{F} & \begin{bmatrix} \textit{cl-rfl-fld} \\ \text{R}\ \boxed{2} \end{bmatrix} \\ \text{I-C} & \textit{no} \end{bmatrix},
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{bát} \rangle \\ \text{F} & \begin{bmatrix} \textit{rest-fld} \\ \text{R}\ \boxed{1} \end{bmatrix} \\ \text{I-C} & \textit{no} \end{bmatrix}, \\
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{velkého} \rangle \\ \text{F} & \begin{bmatrix} \textit{l-adj-fld} \\ \text{R}\ \boxed{3} \end{bmatrix} \\ \text{I-C} & \textit{yes} \end{bmatrix},
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{černého} \rangle \\ \text{F} & \begin{bmatrix} \textit{l-adj-fld} \\ \text{R}\ \boxed{3} \end{bmatrix} \\ \text{I-C} & \textit{yes} \end{bmatrix},
\begin{bmatrix} \textit{s-node} & \\ \text{PHON} & \langle \textit{vlka} \rangle \\ \text{F} & \begin{bmatrix} \textit{noun-fld} & \\ \text{R} & \boxed{3} \begin{bmatrix} \textit{noun-fld} & \\ \text{R} & \begin{bmatrix} \textit{rest-fld} \\ \text{R}\ \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{I-C} & \textit{yes} \end{bmatrix}
\end{array} \right\rangle$$

Figure 5.10: Example of an *s-list*

Since every *d-list* consists of appended continuous *d-lists* of non-head daughters, shuffled with the head daughter's *d-list*, relative order of any two *d-nodes* persists throughout all *d-lists* where these two nodes occur at different levels of embedding. Additionally, since a flat derivation structure is assumed with all dependents occurring as syntactic sisters to each other and to the head, the constraint on the deep/surface word order correspondence needs to be applied only within the construction where these two nodes are combined into a single *d-list* and *s-list*.

5.4 Lexical issues

In the context of the theory of Functional Generative Description, a constraint-based formal framework, and Czech as a language with rich mor-

[35]A more detailed description of the system of fields is given in §6.5.

phology, the role of lexicon, including morphological component, can hardly be overestimated. However, I will not delve into lexical issues more than absolutely necessary, in order to provide a frame of reference for the rest of the present work.

5.4.1 Valency

The concept of valency is used here in a broad sense as including all items being selected: inner participants (actants), free modifications (circonstants), and functional daughters. Optionality (both surface and deep) and iterability are all handled by lexical derivation.

Thus, the proposed arrangement of valency properties of signs is supposed to serve several distinct purposes:

1. The arrangement has to provide the 'bookkeeping' of valency requirements: which of them are satisfied within the current sign and which are to be satisfied higher up in the derivation tree. This is what the VALENCY attribute is good for.[36]

2. The arrangement should support lexically conditioned variants of systemic ordering, serve as the basis of a binding theory and possibly other solutions to linguistic phenomena sensitive to systemic ordering. This part is the responsibility of the attribute DEPENDENTS.

3. The arrangement has to differentiate between three kinds of valency requirements for participants, free modifications and function words. This is the purpose of the three appropriately named valency attributes.

4. Subject should be distinguished among the participants, and it should be possible to express the fact that there is no subject – see the SUBJECT attribute.

The individual valency-related attributes are all present within the *category* type, and the value of all of them is of the type $list(synsem)$:[37]

- The attribute VAL(ENCY) lists all items with which the sign needs to combine, be it participants, free modifications or functional items. The list consists of the list of functional items appended to the list of dependents.

- The attribute DEP(ENDENT)S lists participants and free modifications, ordered according to SO. The list includes all members of the lists of participants and free modifications, shuffled together.

[36] Actually, with the flat derivation structure, all valency requirements are satisfied within a single local tree. Thus, at least a part of the bookkeeping function is not necessary.

[37] The subject list has either one member or it is empty.

- The attribute F-WORDS lists functional items. Members of this list are specified in a basic lexical entry (for reflexive particles with inherent reflexives), or by lexical derivation (for past, conditional, passive and future auxiliaries, prepositions, and conjunctions).

- The attribute P(ARTICI)PANTS lists participants (those complements which are not free modifications), ordered according to SO. Participants are specified in the basic lexical entry.

- The attribute FREE-MOD(IFIER)S lists free modifications, ordered according to SO. Some free modifications may be specified in the basic lexical entry (in many syntactic frameworks, such items would be considered complements, e.g., the PP in *I put the vase on the table*). Other free modifications are specified by lexical derivation (such items correspond to adjuncts).

- The attribute SUBJECT selects one or none of the participants as the subject. Some verbs (e.g., meteorological verbs in Czech in the normal usage) have no participants, therefore there can be no subject. During lexical derivation, subject specification in the basic entry may be altered or the subject may be left out (as in the derivation of passives).

The VALENCY COMPOSITION PRINCIPLE (93), which applies to *lexical* signs, is responsible for assembling the values of DEPENDENTS and VALENCY.

(93) VALENCY COMPOSITION PRINCIPLE
lexical →

$$\left(\begin{array}{l}\left[\text{SYNSEM} \mid \text{LOCAL} \mid \text{CATEGORY} \begin{bmatrix} \text{VALENCY} & \boxed{val} \\ \text{DEPENDENTS} & \boxed{dep} \\ \text{F-WORDS} & \boxed{fw} \\ \text{PARTICIPANTS} & \boxed{part} \\ \text{FREE-MODIFIERS} & \boxed{fm} \\ \text{SUBJECT} & \boxed{subj} \end{bmatrix}\right] \\ \wedge\ \mathtt{append}\left(\boxed{fw}, \boxed{dep}, \boxed{val}\right) \\ \wedge\ \mathtt{shuffle}\left(\boxed{part}, \boxed{fm}, \boxed{dep}\right) \\ \wedge\ \mathtt{member}\left(\boxed{subj}, \boxed{part}\right) \end{array}\right)$$

If a single systemic ordering is assumed for all lexical items, another relation can be added which would properly order the list of dependents. Otherwise, the order must be specified for each item, or relations must be applied to classes of items.

5.4.2 Lexicon, inflection, derivation

The setup of lexical entries and the description of regular lexical phenomena including inflection, derivation, diathesis in the context of HPSG or constraint-based grammars in general has received due attention in recent years: (Flickinger, 1987), (Riehemann, 1993), (Oliva, 1994), (Meurers, 1995), (Davis, 1997), (Davis and Koenig, 1999), (Skoumalová, 2001). In addition, there is hardly an HPSG paper or monograph which would not deal with or bear upon lexical issues.

The solution which has been used for many lexical and even non-lexical phenomena is called lexical rules. Although their theoretical and formal status is not without problems, given the formalism of RSRL, I will take the easier route and assume the so-called description-level lexical rules for the treatment of all the phenomena mentioned above.[38]

Thus, a lexical item will be stored in its base form and with its prototypical properties. Lexical rules can then be used to derive non-base entries describing various forms according to specific inflectional and derivational paradigms, entries with diathetical alternations of the prototypical valency, entries requiring additional complementation by adjuncts and function words. In most aspects, this approach to lexical issues follows that of Przepiórkowski (1999a, §9.2.2), which in turn is based on Manning, Sag, and Iida (1997).

- *sign*
  - *lexical*
    - *basic*
    - *derived* STEM *lexical*
      - *0-deriv*
        - *adj-deriv*
        - *fun-deriv*
        - ...
      - *pass-deriv*
      - *past-deriv*
      - ...
  - *non-lexical*

Figure 5.11: A part of signature for lexical entries

In Fig. 5.11 the relevant part of signature is shown. The type *lexical* is partitioned into *basic* and *derived*. The lexicon is specified as a disjunction of objects of type *basic*, where LE_n is an abbreviation for a lexical entry.

(94) $basic \rightarrow (\mathrm{LE}_1 \vee \mathrm{LE}_2 \vee \ldots \vee \mathrm{LE}_n)$

[38] Of course, for a realistic implementation of Czech morphology, a separate system implementing the morphological component is a must.

A lexical entry is a constraint on how one of the possible *basic* signs may look like. Another option for a lexical sign is to be of the type *derived*. In addition to all attributes appropriate for a lexical sign, a derived type has one additional attribute, namely STEM. Its value is a structure of the *lexical* type, from which the current entry is derived. Since *lexical* is the supertype of *basic* and *derived*, derivation may be recursive.

One of the subtypes of *derived* is the type *0-deriv*, which denotes all entries whose derivation type never involves a change in their phonology. The constraint in (95) applies to all such entries.

(95)
$$\textit{0-deriv} \rightarrow \begin{bmatrix} \textsc{phon}\ \boxed{1} \\ \textsc{stem} \mid \textsc{phon}\ \boxed{1} \end{bmatrix}$$

An example of such a derivation type is *adj-deriv*, a type which denotes all entries which have an adjunct added as one of their valency requirements. The following constraint must be obeyed by all entries of that type:

(96) *adj-deriv* →
$$\left(\begin{bmatrix} \textsc{synsem} \mid \textsc{local} \mid \textsc{category} \mid \textsc{free-mods}\ \boxed{4} \\ \textsc{content}\ \boxed{3} \\ \textsc{stem} \mid \textsc{synsem}\ \boxed{2} \begin{bmatrix} \textsc{local} \mid \textsc{category} \mid \textsc{free-mods}\ \boxed{1} \end{bmatrix} \end{bmatrix} \wedge \ \textbf{append}(\ \boxed{1}, \left\langle \begin{bmatrix} \textsc{local} \begin{bmatrix} \textsc{category} \begin{bmatrix} \textsc{head} \mid \textsc{modified}\ \boxed{2} \\ \textsc{valency}\ \langle\ \rangle \end{bmatrix} \\ \textsc{content}\ \boxed{3} \end{bmatrix} \end{bmatrix} \right\rangle, \boxed{4}\ ) \right)$$

The value of the attribute FREE-MODS is a list of free modifications of the head (see §5.4.1 above). In every sign of the type *adj-deriv* this list ($\boxed{4}$) is identified with the stem's FREE-MOD value ($\boxed{1}$) to which another item is added, namely a saturated (VALENCY $\langle\ \rangle$) *synsem* object. Assuming the standard HPSG-like treatment of semantics, the added item's CONTENT value is identified with that of the derived entry ($\boxed{3}$). This is possible only because the added item has access to the stem's *synsem* via its MODIFIED attribute ($\boxed{2}$). The addition of another valency requirement has no influence on the value of the attribute DEEP, where tectogrammatical information about the sign is located.

The sign in (97) is the lexical entry of a possible satisfier of the added valency requirement, provided that the derived item is a noun. Many attributes are abbreviated: PHON(OLOGY), LOC(AL), CAT(EGORY), MOD(IFIED), CONT(ENT), RESTR(ICTION), INST(ANCE). The item's semantic interpretation (its *content*) is identical with the *content* of the modified item, except for one added restriction on its index. This is the way a noun's *content* is represented: as an index and a set of restrictions.[39] According to the specification of the *0-deriv* type in (95), the content of the derived item is identified with that of the modifier.

(97)
$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{basic}} \\
\textsc{phon} & \textit{červený} \\
\textsc{synsem}\,|\,\textsc{loc} & \left[\begin{array}{ll}
\textsc{cat}\,|\,\textsc{head} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{adjective}} \\
\textsc{mod} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textsc{loc}\,|\,\textsc{cat}\,|\,\textsc{head}\ \textit{noun}} \\
\textsc{cont} & \left[\begin{array}{ll}
\textsc{index} & \boxed{1} \\
\textsc{restr} & \boxed{3}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\textsc{cont} & \left[\begin{array}{ll}
\textsc{index} & \boxed{1} \\
\textsc{restr} & \left\{\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{red}} \\
\textsc{inst} & \boxed{1}
\end{array}\right]\right\} \cup \boxed{3}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

The types *pass-deriv* and *past-deriv* are constrained in a similar way, however, there are a few important differences: the phonology of these types is not identical to the phonology of their stem, the derivation does not affect the items's content, and a valency for a function word is added.

(98) *pass-deriv* →
$$\left(\begin{array}{l}
\left[\begin{array}{ll}
\textsc{phon} & \boxed{1} \\
\textsc{synsem} & \left[\begin{array}{ll}
\textsc{loc} & \left[\begin{array}{ll}
\textsc{cat}\ \boxed{2} & \left[\begin{array}{ll}
\textsc{head} & \textit{pass-pple} \\
\textsc{subj} & \langle[\textsc{synsem}\,|\,\textsc{loc}\,|\,\textsc{deep}\,|\,\textsc{tree}\ \boxed{3}]\rangle \\
\textsc{ppants} & \boxed{0} \oplus \boxed{4} \\
\textsc{cont} & \boxed{6}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\textsc{stem} & \left[\begin{array}{ll}
\textsc{phon} & \boxed{7} \\
\textsc{synsem}\,|\,\textsc{loc} & \left[\begin{array}{ll}
\textsc{cat} & \left[\begin{array}{ll}
\textsc{head} & \textit{base} \\
\textsc{subj} & \boxed{8} \\
\textsc{ppants} & \boxed{0} \oplus \boxed{8} \oplus \boxed{4}
\end{array}\right] \\
\textsc{cont} & \boxed{6}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\wedge\ \texttt{pass_pple}(\boxed{2}, \boxed{7}, \boxed{1}) \\
\wedge\ \texttt{member}([\textsc{fun}\ \textit{patient}], \boxed{3})
\end{array}\right)$$

The type *pass-deriv* makes sure that the derived item's phonology is right, and that the valency requirement corresponding to the tectogrammatical function of *patient* is promoted to the subject position. Relation `pass_pple/2` holds between *category* ($\boxed{2}$), including morphological specifications, the base form and the passive participle of a verb. This relation is responsible for determining appropriate phonology values. The relation

[39] I am ignoring complexities arising due to quantification

`member/2` identifies the valency requirement corresponding to the tectogrammatical function of *patient*.

The passive participle form may be used attributively or predicatively. In the latter case, it is the governor of a finite clause and requires an auxiliary. The type *fun-deriv* in (99) adds the corresponding requirement to the list of function words which should be hosted by the derived item:

(99) *fun-deriv* →

$$\left(\begin{bmatrix} \text{SYNSEM} \mid \text{LOC} \mid \text{CAT } \boxed{1}[\text{F-WORDS } \boxed{3}] \\ \text{STEM} \mid \text{SYNSEM} \mid \text{LOC} \mid \text{CAT } \boxed{2}[\text{F-WORDS } \boxed{4}] \end{bmatrix} \\ \wedge\ \texttt{append}(\ \boxed{4}, \left\langle \begin{bmatrix} \text{LOC} \mid \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{HOST } \boxed{2} \\ \text{COMPLEX } \boxed{1} \end{bmatrix} \\ \text{VALENCY } \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle, \boxed{3}\ ) \right)$$

The type *fun-deriv* is defined in a way which assumes that function words include a rich specification of its contribution to the host's sign. This is done via two head attributes, which are appropriate for function words: HOST, which is identified with the stem's *local* value, and COMPLEX, which is identified with the derived item's *local* value. Thus, a function word may constrain and modify many properties of its host.[40] This type is also used for other function words, e.g., if the passive auxiliary is in the past participle form which requires the past auxiliary.[41]

5.5 Conclusions

In this chapter, the building blocks of the present approach have finally received a distinct shape.

First of all, the issue of how to describe the relation between surface string and tectogrammatical representation in a compositional way was decided by adopting a flat derivation structure with function words standing as sisters to dependents (except for cases where recursive hosting of function words by other function words is appropriate).

Tectogrammatical representation is formalized as a recursive structure, a list consisting of a non-list structure (*d-node*) corresponding to the governor and other lists of the same kind. The performative node is represented as an attribute appropriate to a structure representing the whole expression.

[40] The possibility to modify so much of the host may eventually be found unnecessary and the scope may be restricted. It can certainly be restricted for function words which do not require any modification in the host's tectogrammatical properties.

[41] The past participle form corresponds to the type *past-deriv*. Unlike *pass-deriv*, there is no change in the stem's valency. However, tense and/or aspect of the stem should be modified.

Surface string of words is formalized as a non-recursive list. However, information necessary to determine their position and adjacency is encoded within structures corresponding to the individual words (*s-node*s). This will enable to solve a number of word order phenomena in Czech as a result of interacting constraints originating in more than one level of description.

Together with the final sketch of solutions to lexical issues, this chapter should provide enough background for the specifics of the formalization in the following chapter.

Chapter 6

The formalization

6.1 The 'backbone' constraints

As shown above in §5.1.2, each expression is either of type *lexical*, licensed by the lexical components (lexicon in conjunction with inflection and derivation modules), or it is a syntactic construction of type *non-lexical*, licensed by the combinatory components of grammar. E.g., the *d-list* of a *lexical* sign is determined by the lexical component, while the *d-list* of a syntactic construction consists of *d-list*s of the construction's syntactic components.

In this section, I will propose several basic constraints on non-lexical signs, which regulate the setup of derivation structure, valency, deep, surface and phonology lists, regardless of the level on which they operate: a constraint licensing the composition of *d-list*s will be presented alongside a constraint on the sharing of head features. The constraints will be based on the modified flat derivation structure (see §5.1.2).

6.1.1 Signature for the setup of non-lexical signs

See Fig. 6.1 on p. 156. The level of indentation of types represents the level of embedding, as in Fig. 3.6 on p. 75. Some types are treated as atoms, although they should be specified later for appropriate attributes and/or subtypes. Such types are followed by '...'.

Contrary to what has been assumed so far, the more specific types *d-list*, *s-list*, and the parametric list type *list(phonstring)* do not occur in the signature. Instead, the plain type *list* is used and the further restrictions on the list elements are expressed by constraints.

6.1.2 Composition of *d-lists*

The following constraint (called 'principle') shows how tectogrammatical subtrees of the daughters (SYNSEM|LOCAL|DEEP|TREE) relate to a single tree

top
- *sign* — PHONOLOGY *list*; SYNSEM *synsem*; SURFACE *list*
  - *lexical*
  - *non-lexical* — HEAD-DAUGHTER *sign*; NONHEAD-DAUGHTERS *list(sign)*
- *synsem* — LOCAL *local*; NONLOCAL *nonlocal*
- *local* — CATEGORY *category*; DEEP *deep*; CONTENT ...
- *nonlocal* — ...
- *category* — HEAD *head*; VALENCY *list(synsem)*; DEPENDENTS *list(synsem)*; F-WORDS *list(synsem)*; PARTICIPANTS *list(synsem)*; FREE-MODIFIERS *list(synsem)*; SUBJECT *list(synsem)*; A-MORPH *a-morph*
- *deep* — STATUS *status*; TREE *list*
- *head* — MODIFIED *sign*; S-NODE *s-node*; F-WORD *bool*; ...
- *status*
  - *embedded*
  - *unembedded* — SENTMOD *sentmod*; ...
- *sentmod*
  - *enunc*
  - ...
- *list*
  - *e-list*
  - *ne-list* — FIRST *top*; REST *list*

Figure 6.1: The part of signature relevant for 'backbone' constraints

in the mother category. The principle is expressed formally in the AVM notation as an implication.

Content words In the basic case involving only content words, in every *non-lexical* sign the mother's *d-list* consists of the head daughter's *d-list* into which the non-head daughters' *d-lists* are inserted.

I will refer to this formulation as version 1 of the Deep List Composition Principle (DLCP).

(100) DEEP LIST COMPOSITION PRINCIPLE for content words

$$\begin{bmatrix} \textit{non-lexical} \\ \text{HEAD-DAUGHTER} \mid \text{SYNSEM} \mid \text{LOCAL} \mid \text{CATEGORY} \mid \text{HEAD} \mid \text{F-WORD} \quad \textit{no} \end{bmatrix}$$
$$\rightarrow \left(\begin{array}{l} \begin{bmatrix} \text{SYNSEM} \mid \text{LOCAL} \mid \text{DEEP} \mid \text{TREE } \boxed{5} \\ \text{HEAD-DAUGHTER} \mid \text{SYNSEM} \mid \text{LOCAL} \mid \text{DEEP} \mid \text{TREE } \boxed{1} \\ \text{NONHEAD-DAUGHTERS } \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{collect\_dlists}(\boxed{2}, \boxed{3}) \\ \wedge\ \texttt{append}(\boxed{1}, \boxed{3}, \boxed{4}) \\ \wedge\ \texttt{permute}(\boxed{4}, \boxed{5}) \end{array} \right)$$

The notion of ‘inserting *d-lists*’ is expressed by means of three relations, which are defined in §A.3. The relation `collect_dlists/2` extracts a *d-list* from every non-head daughter and puts it on the list $\boxed{3}$.[1] This list of *d-lists* is appended with the head daughter's *d-list* ($\boxed{1}$), yielding $\boxed{4}$, formally again a *d-list*. This list is permuted into the mother's *d-list* ($\boxed{5}$) and is subject to all constraints on *d-lists* presented above.[2]

As an example of the effect of this constraint, consider the sentence (21) on p. 76, repeated here as (101), which was already represented as a dependency tree in Fig. 3.7 on p. 76.

(101)

Pepa	dneska	pase	sousedovu	kozu
Pepa-NOM	today	graze-PRES-3RD-SG	neighbour-POSS	goat-ACC

‘Today Pepa is grazing the neighbour's goat’

An AVM ‘separate orders’ structure corresponding to the tree was suggested in Fig. 3.8 on p. 77. The structure presented in Fig. 6.2 on p. 158 takes up the same sentence and – assuming the ‘tree as a list of lists’ approach – makes explicit the composition of the topmost *d-list* from *d-lists* of the syntactic components.

[1] If a daughter's *d-list* is empty, as in the case of reflexive particles, nothing is put on the list $\boxed{3}$.

[2] Strictly speaking, the entities labelled $\boxed{3}$ and $\boxed{4}$ do not have any real existence in the model and should be treated as chains, rather than lists, see the chapter on RSRL. I will ignore this formal aspect and treat all chains as lists.

$$
\begin{bmatrix}
\textit{non-lexical} \\
\textsc{phon}\ \left\langle \textit{Pepa, dneska, pase, sousedovu, kozu} \right\rangle \\
\textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \left\langle \boxed{2}\left\langle [\textit{Pepa}] \right\rangle, \boxed{3}\left\langle [\textit{dnes}] \right\rangle, \boxed{1}[\textit{pást}], \boxed{4}\left\langle \boxed{5}[\textit{koza}], \boxed{6}\langle [\textit{soused}] \rangle \right\rangle \right\rangle \\
\textsc{hd}\ \begin{bmatrix} \textit{lexical} \\ \textsc{phon}\ \langle \textit{pase} \rangle \\ \textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \langle \boxed{1} \rangle \end{bmatrix} \\
\textsc{nhd}\ \left\langle \begin{bmatrix} \textit{lexical} \\ \textsc{phon}\ \langle \textit{Pepa} \rangle \\ \textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \boxed{2} \end{bmatrix}, \begin{bmatrix} \textit{lexical} \\ \textsc{phon}\ \langle \textit{dneska} \rangle \\ \textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \boxed{3} \end{bmatrix}, \begin{bmatrix} \textit{non-lexical} \\ \textsc{phon}\ \langle \textit{sousedovu, kozu} \rangle \\ \textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \boxed{4} \\ \textsc{hd}\ \begin{bmatrix} \textit{lexical} \\ \textsc{phon}\ \langle \textit{kozu} \rangle \\ \textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \langle \boxed{5} \rangle \end{bmatrix} \\ \textsc{nhd}\ \left\langle \begin{bmatrix} \textit{lexical} \\ \textsc{phon}\ \langle \textit{sousedovu} \rangle \\ \textsc{ss} \mid \textsc{l} \mid \textsc{d}\ \boxed{6} \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Figure 6.2: An AVM showing the composition of *d-list*s in a flat derivation tree

A few abbreviations have been used to make the AVM structure more compact: (i) SS|L|D|T stands for the path SYNSEM | LOCAL | DEEP | TREE, HD for HEAD-DAUGHTER, and NHD for NONHEAD-DAUGHTERS, (ii) other than the most relevant attributes are suppressed, (iii) phonology substrings are not co-indexed, and (iv) *d-node*s are abbreviated as lemmas. The derivation tree is flat: all dependents are sisters in a single local tree.

Empty nodes So far, we have not seen how to license a tectogrammatical tree involving 'empty' nodes. Such nodes have no surface counterpart, there are more nodes in the tree than content words in the expression. Using GB terminology, they can be exemplified by PRO and pro elements: *Máňa šla tancovat* (an example for *PRO*), represented above in Fig. 5.3, or sentences without overt subjects (examples of *pro*-drop). The solution adopted here will retain the general mechanism of *d-list* composition in DLCP above at the cost of positing derived lexical categories for governors which have non-overt dependents. Thus, a lexical category for finite verbs (except for inherently subjectless verbs such as *pršet* 'to rain') will be available in two flavours: as a verb with a subject valency and a single *d-node* in its *d-list* (representing the verb itself), and as a verb without any subject valency and an additional item in its *d-list*: an embedded *d-list* preceding the governor *d-node* (non-overt nodes are always CB and less dynamic than their governor) including a *d-node* representing the unexpressed subject. On the other hand, a non-finite verb will retain its subject valency (of course, a non-finite subjectless verbs will have none), which will be identified with the subject of a finite verb possibly occurring in the sentence. For more detail and other lexicon-related issues see §5.4 below.

Function words The last outstanding problem in the *d-list* composition concerns function words (see §5.1.2.3). In FGD, function words are independent morphemes such as prepositions, conjunctions, subjunctions, auxiliary verbs including modals and verbal particles (e.g. *se* with inherent reflexives). Function words have no independent existence at the tectogrammatical level, they turn into grammatemes or other information present in a relevant content word. Nevertheless, function words share a number of syntactically important characteristics with content words: e.g., they are subject to agreement and (surface) linearization constraints. The parallel sign-based description of levels allows to posit functional lexical categories with an empty *d-list*. Function words could be selected by quasi-valency requirements of the relevant content word and the only addition to the DLCP above will then be that it concerns only non-empty *d-list*s. The content word's *d-node* will be specified in the lexicon as including the function word's interpretation.

However, this solution will work only for cases in which the function word can be predicted to occur and accordingly interpreted by a lexical specification of the relevant content word. This is a fairly natural approach to verbal particles (in inherent reflexives). For other cases, where it seems that it is rather the function word that predicts the presence of a content word, the more natural solution would be to let function words select their appropriate content words. However, the content word's *d-node* cannot take into account the function word's contribution to its set of grammatemes. Instead of letting function words specify (the core features of) the content word's *d-node* together with the function word's contribution, the proposed solution is based on the derivation of lexical entries for analytical forms of the hosts of function words. The function words are not present in the host's phonology, but as valency requirements. By satisfying the valency requirements, a function word can modify the host's core features and specify some of its morphosyntactic properties, namely its agreement features.

The information necessary to do this is the function word's lexically specified contribution and the host's own *d-node*. The host's *d-node* is accessible together with the entire *synsem* object to the function word by means of identifying the value of one of its attributes with the host's *synsem* in the derived lexical entry. Similarly, the *d-node* including the function word's contribution can be specified by another of the function word's attributes and identified in the derived lexical entry with the host's *d-node*. In this way, a function word such as the future auxiliary in *budu plavat* 'I will swim' is represented at the tectogrammatical level as a grammateme (of posteriority). The same grammateme will be part of the representation of a single wordform *poplavu* ('I will swim', morphologically present tense perfective form with future meaning), where the grammateme does not correspond to a function word but rather to the prefix *po-*.

Since the specification of the function word's contribution is provided by the lexical component and the lexical entries of function words and since the *d-list* value of function words is empty, the DLCP in (100) does not require any modification, except for lifting the condition that it applies only to autosemantic heads: otherwise, recursive embedding of function words would not be possible.

DLCP version 2 can then be stated as follows:

(102) In every *non-lexical* sign the mother's *d-list* consists of the head daughter's *d-list* into which the non-head daughters' *d-list*s are inserted.

In the AVM notation:

(103) Deep List Composition Principle version 2

$$\textit{non-lexical} \;\rightarrow\; \left(\begin{array}{l}\left[\begin{array}{l}\text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{TREE}\ \boxed{5}\\ \text{HEAD-DAUGHTER}\,|\,\text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{TREE}\ \boxed{1}\\ \text{NONHEAD-DAUGHTERS}\ \boxed{2}\end{array}\right]\\ \wedge\ \texttt{collect\_dlists}(\boxed{2}, \boxed{3})\\ \wedge\ \texttt{append}(\boxed{1}, \boxed{3}, \boxed{4})\\ \wedge\ \texttt{permute}(\boxed{4}, \boxed{5})\end{array}\right)$$

6.1.3 Composition of *s-list*s

The constraint below shows how the surface order of words corresponding to the individual daughters relates to the surface order of words corresponding to the mother category.

(104) In every *non-lexical* sign the mother's *s-list* consists of the daughter's *d-list*s, shuffled together.

A list created by 'shuffling together' other lists consists of a permutation of all members of the lists, provided that the relative order of members with each of the lists is retained. In the AVM notation, the constraint can be expressed as follows:

(105) Surface List Composition Principle

$$\textit{non-lexical} \;\rightarrow\; \left(\begin{array}{l}\left[\begin{array}{l}\text{SURFACE}\ \boxed{5}\\ \text{HEAD-DAUGHTER}\,|\,\text{SURFACE}\ \boxed{1}\\ \text{NONHEAD-DAUGHTERS}\ \boxed{2}\end{array}\right]\\ \wedge\ \texttt{collect\_slists}(\boxed{2}, \boxed{3})\\ \wedge\ \texttt{append}(\boxed{1}, \boxed{3}, \boxed{4})\\ \wedge\ \texttt{multi\_shuffle}(\boxed{4}, \boxed{5})\end{array}\right)$$

The relation `collect_slists/2` extracts an *s-list* from every non-head daughter and puts it on the list $\boxed{3}$.[3] The first argument ($\boxed{4}$) of the relation **`multi_shuffle/2`** consists of the head daughter's *s-list* ($\boxed{1}$, the $\boxed{4}$'s head) and the collected non-head daughters' *s-list*s ($\boxed{3}$, the $\boxed{4}$'s tail). The second argument is a list where the lists collected in $\boxed{4}$ are shuffled together, i.e., where the members of the lists $\boxed{4}$ are permuted, provided that the relative order of the members within the lists is retained.[4]

This definition of SLCP allows for any surface order, i.e., for total scrambling. Of course, additional constraints on the surface list are needed in order to describe word order of a human language – see §6.3.

6.1.4 Satisfaction of valency requirements

The following constraint shows how valency requirements of the head daughter (HEAD-DAUGHTER|SYNSEM|LOCAL|CATEGORY|VALENCY) are satisfied by being identified with the SYNSEM value of the non-head daughters and, as a result of this satisfaction, being left-out from the valency list of the mother category (SYNSEM|LOCAL|CATEGORY|VALENCY)

(106) In every *non-lexical* sign the head daughter's *valency* list equals the list of non-head daughter's *synsem* objects shuffled with the mothers's *valency* list.

In the AVM notation:

(107) VALENCY PRINCIPLE

non-lexical →

$$\left(\begin{array}{l} \begin{bmatrix} \text{SYNSEM} \mid \text{LOCAL} \mid \text{CATEGORY} \mid \text{VALENCY } \langle\rangle \\ \text{HEAD-DAUGHTER} \mid \text{SYNSEM} \mid \text{LOCAL} \mid \text{CATEGORY} \mid \text{VALENCY } \boxed{1} \\ \text{NONHEAD-DAUGHTERS } \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{collect\_synsems}(\boxed{2}, \boxed{1}) \end{array} \right)$$

Together with lexical specification of valency as including complements, adjuncts and function words, Valency Principle imposes totally flat structure on the derivation tree. All valency requirements of the head daughter ($\boxed{1}$) are satisfied within a single local tree by a list of non-head daughters ($\boxed{2}$). The mother's valency is an empty list. The relation `collect_synsems/2` extracts *synsem* objects from non-head daughters and puts them on a list ($\boxed{1}$), identical to the valency list. The order of non-head daughters as listed in $\boxed{2}$ thus corresponds to the valency specification, not to the surface order.

[3] If traces were used for empty nodes, the 'trace' daughter's *s-list* would be empty, and nothing would be put on the list $\boxed{3}$.

[4] Note that the head daughter contributes a single list into $\boxed{4}$, while in DLCP the head daughter contributes not a list, but rather the members of the list. This corresponds to the difference in the setup of *s-list* and *d-list*: only the latter has always a distinguished non-list member as the governor of the respective subtree.

6.1.5 Phonological realization

The constraint below relates *s-list*s with the actual sequence of forms (PHONOLOGY).

(108) PHONOLOGY PRINCIPLE

$$non\text{-}lexical \rightarrow \left(\begin{bmatrix} \text{PHONOLOGY } \boxed{1} \\ \text{SURFACE } \boxed{2} \end{bmatrix} \wedge \texttt{collect\_phonology}(\boxed{2}, \boxed{1}) \right)$$

The relation `collect-phonology/2` holds between an *s-list* and a list whose members are identified with the members of the lists standing as PHONOLOGY values of the *s-list*, in that order.

6.1.6 The joint effect of the constraints

In Fig. 6.3 on p. 163 the few basic constraints on the type *non-lexical* are conflated into a single schema. The schema includes: the DEEP LIST COMPOSITION PRINCIPLE – see (103), SURFACE LIST COMPOSITION PRINCIPLE – see (105), VALENCY PRINCIPLE – see (107), and PHONOLOGY PRINCIPLE – see (108).

Several caveats are due:

1. In order to make the schema more readable by using as few relations as possible, it covers only local trees with two or more non-head daughters.

2. For similar reasons, the schema only works for content words.[5]

3. Also, the head daughter consists of a single semanteme.[6]

4. The string `collect-phonology` stands for the two-place relation `collect-phonology` (see §A.3 for definition of the relation). To save space, the relation is used here as a function which takes a list as an argument and returns another list whose members are identified with the PHONOLOGY values of the argument list, in that order. Within daughters, this relation is replaced by simple co-indexation.

5. The string `permute` stands for the two-place relation `permute` (see §A.3 for definition of the relation). To save space, the relation is used here as a function which takes a list as an argument and returns another list where the members of the argument are ordered in a different way.

[5] Function word daughters, which have an empty *d-list*, contribute nothing the mother's *d-list*.

[6] Although in a flat structure the mother's *d-list* is composed of the daughters' *d-list*s within a single local tree, the head may be specified in the lexicon as a derived lexical item with dependents, esp. empty nodes.

non-lexical $\rightarrow$

$$
\left[\begin{array}{ll}
\text{PHONOLOGY} & \texttt{collect-phonology}(\boxed{s}) \\
\text{SYNSEM}\,|\,\text{LOCAL} & \left[\begin{array}{l} \text{DEEP}\,|\,\text{TREE}\ \texttt{permute}(\left\langle \boxed{d1}, \boxed{d2}, \ldots, \boxed{dn}\right\rangle) \\ \text{CATEGORY}\,|\,\text{VALENCY}\ \langle\rangle \end{array}\right] \\
\multicolumn{2}{l}{\text{SURFACE}\ \boxed{s} = \boxed{s1} \bigcirc \boxed{s2} \bigcirc \ldots \bigcirc \boxed{sn}} \\
\text{HEAD-DAUGHTER} & \left[\begin{array}{ll} \text{PHONOLOGY} & \boxed{p1} \\ \text{SYNSEM}\,|\,\text{LOCAL} & \left[\begin{array}{l}\text{DEEP}\,|\,\text{TREE}\ \left\langle\boxed{d1}\right\rangle \\ \text{CATEGORY}\,|\,\text{VALENCY}\ \left\langle \boxed{v2},\ldots,\boxed{vn}\right\rangle\end{array}\right] \\ \text{SURFACE} & \boxed{s1}\left\langle\left[\text{PHONOLOGY}\ \boxed{p1}\right]\right\rangle \end{array}\right] \\
\text{DEPENDENT-DAUGHTERS} & \left\langle \begin{array}{l}
\left[\begin{array}{ll} \text{PHONOLOGY} & \left\langle \boxed{p2.1}, \boxed{p2.2}, \ldots, \boxed{p2.n} \right\rangle \\ \text{SYNSEM}\ \boxed{v2} & \left[\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{TREE}\ \boxed{d2}\right] \\ \text{SURFACE}\ \boxed{s2} & \left\langle \begin{array}{l} \left[\text{PHONOLOGY}\ \boxed{p2.1}\right], \\ \left[\text{PHONOLOGY}\ \boxed{p2.2}\right], \\ \ldots\ , \\ \left[\text{PHONOLOGY}\ \boxed{p2.n}\right], \end{array} \right\rangle \end{array}\right], \\
\ldots\ , \\
\left[\begin{array}{ll} \text{PHONOLOGY} & \left\langle \boxed{pn.1}, \boxed{pn.2}, \ldots, \boxed{pn.n} \right\rangle \\ \text{SYNSEM}\ \boxed{vn} & \left[\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{TREE}\ \boxed{dn}\right] \\ \text{SURFACE}\ \boxed{sn} & \left\langle \begin{array}{l} \left[\text{PHONOLOGY}\ \boxed{pn.1}\right], \\ \left[\text{PHONOLOGY}\ \boxed{pn.2}\right], \\ \ldots\ , \\ \left[\text{PHONOLOGY}\ \boxed{pn.n}\right], \end{array} \right\rangle \end{array}\right]
\end{array}\right\rangle
\end{array}\right]
$$

Figure 6.3: A schema conflating some constraints on non-lexical signs

6. The operator symbol $\bigcirc$ is a shorthand for the three-place relation `shuffle` (see §A.3 for definition of the relation). The relation holds between three lists: the first two lists are 'shuffled' together into the third list. More exactly, the third list is a permutation of a list resulting from appending the first two lists, provided that the relative order of members within each of the first two lists is retained. The relation is again used functionally and applies successively to each pair of items conjoined by the operator $\bigcirc$.

7. The order of items in the lists is subject to a number of other constraints, which will be presented in due course. Similarly, some other general constraints apply (such as HEAD FEATURE PRINCIPLE).

6.2 Constraints on *d-list*s

In this section, a set of admissible structures representing expressions at the tectogrammatical level will be outlined without recourse to the other levels. Such a description could work as a stand-alone grammar licensing such a set. However, the description would be heavily underspecified, and as a result the set would be much larger: there would be many tectogrammatical trees which do not correspond to any expression. Leaving the general difficulty of exhaustively enumerating all constraints aside, the main reason is that a detailed tectogrammatical lexicon is needed for such a task.

Instead of attempting to define the set of tectogrammatical structures independently, I will assume that the following constraints on *d-list*s are integrated with constraints of other kinds.

The constraints presented in this section correspond to points 10, 11, 12, and 14 on p. 87.

6.2.1 Signature for TR

First, it is necessary to make clear what kinds of formal objects will be used as components of TR by presenting the relevant part of signature (see Fig. 6.4).[7]

The plain type *list* is used and the more specific type *d-list* and further restrictions on the list elements is expressed by a constraint. The constraint (called 'principle') is presented below in (109). The principle is expressed formally in the AVM notation as an implication: the antecedent is a *deep* object with a specific value of its attribute TREE. The consequent is a relation with the attribute's value as its single argument.

[7]The attribute SENTMOD of *status* would in fact be more appropriately placed within *d-verb*, because it is also relevant for embedded clauses, cf. Hajičová, Panevová, and Sgall (2000, p. 20). Also, the values of CB in *d-node* should not be mere subtypes of *bool*, but should distinguish cases of contrastive topic.

- *top*
  - *deep* — STATUS *status*; TREE *list*
  - *status*
    - *embedded*
    - *unembedded* — SENTMOD *sentmod*; ...
  - *sentmod*
    - *enunc*
    - ...
  - *d-node* — FUN *fun*; CB *bool*; CORE *d-wcl*
  - *fun*
    - *participant*
      - *act*
      - *pat*
      - *addr*
      - *orig*
      - *eff*
      - *id*
      - *mat*
    - *free-modifier*
      - *time*
        - *tsin*
        - *twen*
        - *thl*
        - ...
      - *addr*
      - ...
    - ...
  - *d-wcl* — ...
  - *bool*
    - *yes*
    - *no*
  - *list*
    - *e-list*
    - *ne-list* — FIRST *top*; REST *list*

Figure 6.4: The part of signature relevant for TR

(109) TREE AS LISTS PRINCIPLE (TLP):

$$\begin{bmatrix} \textit{deep} & \\ \text{TREE} & \boxed{1} \end{bmatrix} \rightarrow \texttt{d\_list}(\boxed{1})$$

The crucial part of TLP is the relation `d_list/1`, which is defined in §A.3.7. The relation is satisfied iff the list in the argument position consists of the following member(s): exactly one *d-node* and $0 - n$ lists. Each of those lists (if present) must satisfy `d_list/1` again. If the relation is satisfied for the list in the argument position, that list is a proper representation of dependency tree.

For convenience, I will continue using the 'pseudo-types' *d-list*, *s-list*, *list(phonstring)* etc. for lists constrained by principles of grammar. To rephrase TLP in a more intuitive way, a *d-list* must correspond to (be interpretable as) a projective tree: be composed of exactly one *d-node* and any number of *d-list*s. Also, any formal object within any *d-list*, such as *d-node*, must be well-formed. The structure is exemplified in Fig. 6.5.

The definition of the type *deep* in the signature in Fig. 6.4 guarantees that a TR tree consists of non-overlapping local structures (*list*s), while TLP defined in (109) makes sure that every local structure is a tree, i.e., has a single governor.

Now I am ready to present the constraints.

6.2.2 At least one NB node in a TR

The point 10 says that there is **at least one NB node or grammateme** in the tree. Again, this is expressed formally in the AVM notation as an implication: the antecedent is an object of the type *deep* with STATUS *unemb*edded and a specific value of the attribute TREE, while the consequent is a relation which guarantees that there is at least one non-bound *d-node* in the tree, see (110).

$$
\begin{bmatrix}
\textit{deep} & \\
\textsc{status} & \begin{bmatrix} \textit{unemb} & \\ \textsc{sentmod} & \textit{enunc} \end{bmatrix} \\
\textsc{tree} & \begin{bmatrix}
\textit{ne-list} & \\
\textsc{first} & \begin{bmatrix}
\textit{ne-list} & \\
\textsc{first} & \begin{bmatrix}
\textit{d-node} & \\
\textsc{fun} & \textit{actor} \\
\textsc{cb} & \textit{yes} \\
\textsc{core}\ \boxed{1} & \begin{bmatrix} \textit{d-noun} & \\ \textsc{lemma} & \textit{Máňa} \\ \textsc{d-number} & \textit{sg} \end{bmatrix}
\end{bmatrix} \\
\textsc{rest} & \textit{e-list}
\end{bmatrix} \\
\textsc{rest} & \begin{bmatrix}
\textsc{first} & \begin{bmatrix}
\textit{d-node} & \\
\textsc{fun} & \textit{top} \\
\textsc{cb} & \textit{no} \\
\textsc{core} & \begin{bmatrix} \textit{d-verb} & \\ \textsc{lemma} & \textit{jít} \\ \textsc{d-tense} & \textit{anterior} \end{bmatrix}
\end{bmatrix} \\
\textsc{rest} & \begin{bmatrix}
\textit{ne-list} & \\
\textsc{first} & \begin{bmatrix}
\textit{ne-list} & \\
\textsc{first} & \begin{bmatrix}
\textit{d-node} & \\
\textsc{fun} & \textit{actor} \\
\textsc{cb} & \textit{yes} \\
\textsc{core} & \boxed{1}
\end{bmatrix} \\
\textsc{rest} & \textit{e-list}
\end{bmatrix} \\
\textsc{rest} & \begin{bmatrix}
\textit{d-node} & \\
\textsc{fun} & \textit{patient} \\
\textsc{cb} & \textit{no} \\
\textsc{core} & \begin{bmatrix} \textit{d-verb} & \\ \textsc{lemma} & \textit{tancovat} \\ \textsc{tense} & \textit{posterior} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 6.5: A simplified TR of *Máňa šla tancovat* as a list of lists including the performative node in the first/rest list notation

(110) AT LEAST ONE NON-BOUND NODE PRINCIPLE (ONBP):

$$\begin{bmatrix} \textit{deep} & \\ \text{STATUS} & \textit{unemb} \\ \text{TREE} & \boxed{1} \end{bmatrix} \rightarrow \texttt{nested\_member}([\text{CB } \textit{no}], \boxed{1})$$

6.2.3 The position of the governor in a local tree

The point 11 concerns the position of the root of every local tree with respect to its NB dependents: in every local tree, the **root precedes all NB nodes**. The point 12 concerns the position of the root of every local tree with respect to its CB dependents: disregarding NB grammatemes, in every local tree, either the **root follows all CB nodes**, **or** there are **CB nodes following the root which have a subordinated NB node**.

These two requirements will be formalized as a single principle. In (111) the principle is stated using universal quantifiers.[8, 9]

(111) GOVERNOR POSITION PRINCIPLE (GPP):

$$\begin{bmatrix} \textit{deep} \\ \text{TREE} \ \left\langle \boxed{1} \oplus \langle \textit{d-node} \,|\, \boxed{2} \rangle \right\rangle \end{bmatrix} \rightarrow$$

$$\forall_{\boxed{3}} \Big(\texttt{dependent}(\boxed{3}\textit{d-node}, \boxed{1}) \ \rightarrow \ \boxed{3}[\text{CB } \textit{yes}] \Big)$$
$$\wedge \left(\begin{array}{l} \forall_{\boxed{4}} \Big(\texttt{dependent}(\boxed{4}\textit{d-node}, \boxed{2}) \ \rightarrow \ \boxed{4}[\text{CB } \textit{no}] \Big) \\ \vee \ \forall_{\boxed{5}} \exists_{\boxed{6}} \exists_{\boxed{7}} \left(\begin{array}{l} \Big(\texttt{member}(\boxed{5}, \boxed{2}) \wedge \texttt{member}(\boxed{6}[\text{CB } \textit{yes}], \boxed{5}) \Big) \\ \quad \rightarrow \ \texttt{nested\_member}(\boxed{7}[\text{CB } \textit{no}], \boxed{5}) \end{array} \right) \end{array} \right)$$

The principle says that in the tree (the value of TREE) of every object of type *deep* the following two basic statements must hold for the (possibly empty) lists of left branches ($\boxed{1}$) and right branches ($\boxed{2}$):

1. All left-branching dependents of the root are CB. More specifically: any *d-node* standing in the dependent/2 relation with the list $\boxed{1}$ must be CB.

[8]Note that in the AVM notation used here the constraints should be satisfied by all objects in the model in the sense of an implicit universal quantification over such objects. The quantification binds a variable restricted by the antecedent with the entire formula in its scope: 'for each object, if it is of type *deep* with the value of TREE being as specified, the consequent must hold'. All other variables in the formulas (corresponding to components of the object) are implicitly bound by existential quantifiers with their scope over the entire formula. Wherever universal quantification or narrower scope is intended, explicit notation should be used. For the complete presentation of abbreviatory conventions see Richter (1999, p. 102-112).

[9]The infix operator $\oplus$ is a shorthand for the append relation, which is defined in §A.3: $x \oplus y = z$ is equivalent to $\texttt{append}(x, y, z)$. Here, the append relation is used to pick the part of *d-list* preceding the governor. The angle brackets enclose a list. The head of a list is separated from its tail by vertical bar.

2. All right-branching dependents of the root are NB, unless the CB dependent has a NB subordinate. More specifically, one of the two following cases holds:
   (a) All right-branching dependents of the root are NB: any *d-node* standing in the `dependent/2` relation with the list $\boxed{2}$ must be NB.
   (b) Any right-dependent subtree ($\boxed{5}$) of the root, a member of $\boxed{2}$ which includes a CB *d-node* (as a dependent of the root) must have a subordinate NB *d-node* (i.e. a NB *d-node* standing in the `nested_member/2` relation with the list $\boxed{5}$.

The relation `dependent` looks for an item (its first argument) within a list which is a member of the list specified as its second argument (see §A.3 for definition).

6.2.4 NB nodes follow their sisters

GPP makes sure that the governor is properly ordered at the boundary between its CB and NB dependents, with the exception of one or more CB nodes with a NB subordinate. Such CB nodes are required to follow the governor. In order to handle the case where one or more CB nodes following the governor have any NB sisters, an additional principle is necessary which would guarantee that the NB nodes follow their CB sisters. The principle could be stated in a general way, requiring that any NB node follows its CB sister. However, since GPP handles the order of left dependents, NBLP below will be concerned with right dependents only.

(112) NON BOUND NODES LAST PRINCIPLE (NBLP):

$$\forall\boxed{4}\ \forall\boxed{5}$$
$$\left(\left(\begin{bmatrix}\textit{deep}\\ \text{TREE}\ \boxed{1}\end{bmatrix} \wedge \texttt{right\_subordinates}(\boxed{2}, \boxed{1}) \wedge \texttt{dependents}(\boxed{3}, \boxed{2}) \\ \wedge\ \texttt{member}(\boxed{4}\ [\text{CB}\ \textit{yes}], \boxed{3}) \wedge \texttt{member}(\boxed{5}[\text{CB}\ \textit{no}], \boxed{3})\right) \\ \rightarrow \texttt{precedes}(\boxed{4}, \boxed{5}, \boxed{3})\right)$$

It is the relation `right_subordinates/2` together with the two instances of relation `member/2` which constrain the domain of application of NBLP to the case where there are both CB and NB right dependents. The relation `dependents/2` makes sure that only nodes (immediately) dependent on the governor of $\boxed{1}$ occur in $\boxed{3}$ as an argument of `member/2` and `precedes/3`.

6.2.5 NB sisters are ordered according to SO

The point (14) concerns the order of NB sisters: In every local tree, the (horizontal) order of dependent NB nodes is determined by **systemic ordering** (SO) of their tectogrammatical functions.

The following AVM constraint provides a formalization of the principle.

(113) NON-BOUND NODES SYSTEMIC ORDERING PRINCIPLE (NBSOP):

$$\forall \boxed{4}\ \forall \boxed{5}\ \forall \boxed{6}\ \forall \boxed{0}$$
$$\left(\left(\begin{bmatrix} \textit{deep} \\ \text{TREE } \boxed{1} \end{bmatrix} \land \texttt{right\_subordinates}(\boxed{2}, \boxed{1}) \land \texttt{dependents}(\boxed{3}, \boxed{2}) \right. \right.$$
$$\left. \land\ \boxed{3} = \boxed{4} \oplus \langle\ \boxed{5}[\text{CB } \textit{no}], \boxed{6}[\text{CB } \textit{no}]\ \mid\ \boxed{0}\ \rangle \right)$$
$$\left. \rightarrow \texttt{so\_precedes}\Big(\boxed{5}, \boxed{6}\Big) \right)$$

Again, the principle concerns right dependents ($\boxed{3}$) of the governor of $\boxed{5}$, requiring that any two NB nodes in $\boxed{3}$ must comply with systemic ordering. Any two immediately following NB nodes ($\boxed{5}$ and $\boxed{6}$) must satisfy relation `so_precedes/2` (see §A.3.28). The relation makes sure that the two nodes are ordered according to systemic ordering specific to the language: the first argument's functor precedes the second argument's functor in the SO-ordered list of functors.

6.3 Constraints on the DWO/SWO relation

These constraints relate the tectogrammatical tree (*d-list*) and the morphemic string (*s-list*) within a sign. This section is a follow-up to §4.3, especially to the section on mismatches between deep and surface word order (§4.3.3). The constraints will follow the conclusions concerning surface word order in §4.3.5, which is repeated here for convenience:

General constraints on word order

1. The relative SWO of every two content words corresponds to DWO of the corresponding semantemes, unless any of the cases in the list of Special SWO conditions apply (see below).
2. A function word is ordered adjacently to its host, their order being determined by a syntactic constraint, unless any of the cases in the list of Special SWO conditions apply (see below).
3. For the relative SWO of every two function words in a cluster the list of Special SWO conditions applies.

Special SWO conditions

1. The word ordered first in SWO is the intonation centre of the utterance and corresponds to focus proper.
2. A syntactic constraint requires otherwise.
3. A stress pattern requires otherwise.
4. A word is ordered first in a SWO domain, the domain is larger than that of its corresponding tectogrammatical local tree, and the word corresponds to topic proper or to contrastive topic.

The only type of DWO/SWO mismatch with function words ignored (see §4.3.3) which is not addressed here, is 'deletion due to syntactic constraints' (§4.3.3.5). This phenomenon is handled within the lexical component (by deriving lexical entries with cancelled valency requirements) or by allowing unsaturated signs in the syntactic component (such as embedded infinitival clauses).

6.3.1 Signature for the surface list

See Fig. 6.6 on p. 171.

- *top*
  - *sign* — PHONOLOGY *list*, SYNSEM *synsem*, SURFACE *list*
  - *s-node* — PHONOLOGY *list*, FIELD *field*, I-CENTRE *bool*
  - *field*
    - *matrix-fld*
    - *embedded-fld* — REGION *field*
      - *pre-cl-fld*
      - *cl-fld*
      - *rest-fld*
      - *fin-fld*
      - *dep-fld*
        - *noun-fld*
        - *pp-fld*
        - *adj-fld*
          - *l-adj-fld*
          - *h-adj-fld*
        - *adv-fld*
        - *inf-fld*
        - *aux-fld*
        - *emb-cls-fld*
          - *sconj-cls-fld*
          - *wh-cls-fld*
      - ...

Figure 6.6: The part of signature relevant for surface level

6.3.2 Disjunctive constraint application

Unlike all constraints on the type *non-lexical* presented so far, constraints on the relation between DWO and SWO cannot be applied conjunctively. Instead, they should thought of as parts of a meta-constraint on the *non-lexical* type, in which they appear as disjuncts in the consequent. This meta-constraint can then have the same formal status as other constraints in the grammar.[10]

I will refer to the meta-constraint as the DEEP/SURFACE ORDER PRINCIPLE (DSOP) and define it as in (114), where the abbreviations stand for the individual constraints.

(114) DEEP/SURFACE ORDER PRINCIPLE (DSOP) version 1:
DSOP $\rightarrow$ IOP $\vee$ LDTP $\vee$ NFICP

Without further specification, the disjuncts can apply to any object, including the whole sign. Note, however, that in the summary of word order constraints informally presented above, the 'default' constraint linking SWO with DWO applies to two surface items paired with two deep items, unless other conditions concerning these items are satisfied. Therefore, the whole consequent in (114) should rather apply to the pairs of items selected by the antecedent and the individual disjuncts should be specified as relations with the selected items as their arguments. Crucially, variable bindings set in the antecedent should be preserved in all disjuncts. DSOP can then be schematically presented as in (115).

(115) DEEP/SURFACE ORDER PRINCIPLE (DSOP) version 2:

$$\forall\boxed{5}\forall\boxed{6}\forall\boxed{7}\forall\boxed{8}\left(\begin{array}{l}\left(\begin{array}{l}\begin{bmatrix}\textit{non-lexical} \\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{TREE}\ \boxed{1} \\ \text{SURFACE}\ \boxed{2} \\ \text{HEAD-DAUGHTER}\ \boxed{3} \\ \text{NONHEAD-DAUGHTERS}\ \boxed{4}\end{bmatrix} \\ \wedge\ \texttt{deep\_surf}\left(\boxed{5}, \boxed{6}, \boxed{7}, \boxed{8}, \boxed{3}, \boxed{4}\right)\end{array}\right) \\ \rightarrow\quad \text{IOP} \vee \text{LDTP} \vee \text{NFICP} \vee \ldots\end{array}\right)$$

The relation `deep_surf/6` pairs two distinct *d-nodes* ($\boxed{5}$, $\boxed{6}$) from the local tectogrammatical tree with their *s-node* counterparts ($\boxed{7}$, $\boxed{8}$) by inspecting the head daughter sign ($\boxed{3}$) and the signs in the list of non-head daughters ($\boxed{4}$). Due to the flat derivation structure, all nodes in the local tectogrammatical tree as well as all their surface counterparts are accessible in the daughter signs.

By default, DSOP ignores nodes which have no counterpart at the other level. This concerns tectogrammatical nodes with no overt realization as well as function words. Of course, it also concerns any *s-node*s which do not correspond to *d-node*s in the local tectogrammatical tree. Thus, only items included in the local tectogrammatical tree are ordered by DSOP.

[10] The same solution is used for the lexicon, which is formalized as a single constraint on the lexical type with the individual lexical entries as disjuncts in the consequent.

Since it is not possible in general to find the *s-node* counterpart of a *d-node* by searching through a daughter's *s-list*, each sign's *head* object is appropriate to an attribute S-NODE, whose value is identical with the *s-node* for that sign's head.

6.3.3 DWO and SWO coincide

According to point 1 in §4.3.5.1, "the relative SWO of every two content words corresponds to DWO of the corresponding semantemes, unless any of the cases in the list of Special SWO conditions apply (see below)."

Assuming for a moment that SWO and DWO of content words are always identical, a constraint enforcing the same order on both levels would be as in (116).

(116) IDENTICAL ORDER PRINCIPLE (IOP) version 1:

$$\forall\boxed{5}\ \forall\boxed{6}\ \forall\boxed{7}\ \forall\boxed{8}$$

$$\left(\begin{array}{l}\left(\begin{array}{l}\left(\begin{bmatrix} \textit{non-lexical} \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{DEEP} \mid \text{TREE}\ \boxed{1} \\ \text{SURFACE}\ \boxed{2} \\ \text{HEAD-DAUGHTER}\ \boxed{3} \\ \text{NONHEAD-DAUGHTERS}\ \boxed{4} \end{bmatrix}\right) \\ \wedge\ \texttt{deep\_surf}(\boxed{5}, \boxed{6}, \boxed{7}, \boxed{8}, \boxed{3}, \boxed{4}) \end{array}\right) \\ \quad \rightarrow \begin{pmatrix} \texttt{d\_precedes}(\boxed{5}, \boxed{6}, \boxed{1}) \\ \wedge\ \texttt{precedes}(\boxed{7}, \boxed{8}, \boxed{2}) \end{pmatrix} \end{array}\right)$$

The relation `d_precedes/3` makes sure that a *d-node*, the first argument, precedes another *d-node*, the second argument, in a *d-list*, the third argument. Each *d-node* is either the governor of the local tectogrammatical tree or its immediate dependent.

The relation `precedes/3` holds for two *s-node*s in an *s-list* if the first argument *s-node* precedes the second argument *s-node* in that list, the third argument. There are two things worth mentioning: (i) The constraint does not require that subtrees governed by the *d-node*s are realized continuously. Continuity is guaranteed if *s-nodes* corresponding to the subtree are compacted. (ii) The conjunction in the consequent of the principle guarantees that the principle is applicable no matter whether the description is used for generation or parsing.

Since the assumption about SWO and DWO being always identical is not true, the principle, more precisely the consequent in (116), must be treated as one of the disjuncts in DSOP (115). It is therefore sufficient to define IOP simply as in (117).

(117) Identical Order Principle (IOP) version 2:

$$\begin{pmatrix} \texttt{d\_precedes}(\boxed{5}, \boxed{6}, \boxed{1}) \\ \wedge\ \texttt{precedes}(\boxed{7}, \boxed{8}, \boxed{2}) \end{pmatrix}$$

6.3.4 Left dislocation of topic proper

See 4.3.3.1. Since this kind of SWO/DWO mismatch is optional, no modification of IOP is required.

Contrastive topic can be subsumed under the following constraint as well. However, it would need to be explicitly marked as such in its *d-node* and as the recipient of contrastive stress in the corresponding *s-node*.

Assuming naively that topic proper is identical with the initial context-bound dependent in the local tectogrammatical tree, the corresponding *s-node* is located at the initial position, i.e., in the field *pre-cl-fld*.[11]

For expository purposes, I will pretend again that left dislocation of topic proper is the only principle constraint applicable to pairs of deep and surface nodes:

(118) Left Dislocation of Topic Principle (LDTP) version 1:

$$\forall\boxed{5}\ \forall\boxed{6}\ \forall\boxed{7}\ \forall\boxed{8}$$

$$\left(\begin{array}{l} \left(\begin{array}{l} \begin{bmatrix} \textit{non-lexical} \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{DEEP} \mid \text{TREE}\ \boxed{1} \\ \text{SURFACE}\ \boxed{2} \\ \text{HEAD-DAUGHTER}\ \boxed{3} \\ \text{NONHEAD-DAUGHTERS}\ \boxed{4} \end{bmatrix} \\ \wedge\ \texttt{deep\_surf}(\boxed{5}, \boxed{6}, \boxed{7}, \boxed{8}, \boxed{3}, \boxed{4}) \end{array} \right) \\ \quad \rightarrow\ \exists\boxed{9}\ \exists\boxed{0} \left(\begin{array}{l} \boxed{5}[\text{CB}\ \textit{yes}] \wedge \boxed{1}\langle\, \boxed{9}[\textit{d-list}] \mid \boxed{0}\, \rangle \wedge \texttt{member}(\boxed{5}, \boxed{9}) \\ \wedge\ \texttt{component}(\textit{pre-cl-fld}, \boxed{7}) \end{array} \right) \end{array} \right)$$

The principle maintains that the first from the two *d-nodes* selected in the antecedent ($\boxed{5}$) is the governor of the initial *d-list* ($\boxed{9}$) in $\boxed{1}$ and that somewhere in the corresponding *s-node* ($\boxed{7}$) there is a field *pre-cl-fld*.[12] As in IOP, LDTP does not require that the initial subtree is realized continuously. Continuity is guaranteed if *s-nodes* corresponding to the subtree are compacted.

Now if LDTP is included as one of the disjuncts in the consequent of DSOP (115), and if the condition on contextual boundness of the initial *d-node* is included into the disjunct, only the consequent of DSOP (115) is needed:

[11] This field occurs in finite clauses: *matrix-fld* or *emb-cls-fld*.

[12] The relation `deep_surf/6` is assumed to pick the two *d-nodes* in an order according to DWO, which makes it possible to treat only the first *d-node*. The relation `component/2` holds if the object in the first argument is accessible by following a path of attributes from the root (the outermost bracket) of the second argument.

(119) Left Dislocation of Topic Principle (LDTP) version 2:

$$\left(\exists \boxed{9} \exists \boxed{0} \begin{pmatrix} \boxed{5}[\text{CB } \textit{yes}] \wedge \boxed{1}\langle\ \boxed{9}[\textit{d-list}] \mid \boxed{0}\ \rangle \\ \wedge\ \texttt{member}\Big(\boxed{5}, \boxed{9}\Big) \\ \wedge\ \texttt{component}(\textit{pre-cl-fld}, \boxed{7}) \end{pmatrix} \right)$$

Note that LDTP applies redundantly to topic proper in the initial position.

6.3.5 Non-final placement of the intonation centre

See 4.3.3.2. Again, since this kind of SWO/DWO mismatch is optional, no modification of IOP is required.

I will assume that this constraint concerns focus proper, which will be naively assumed as identical with the final node in the local tectogrammatical tree. This final node can be a dependent or the governor. The surface counterpart of such a node can either appear in accordance with IOP in the final position of the corresponding region, or precede other nodes in that region (or even in higher regions, if not compacted to that region), provided that it receives an appropriate stress.

In order to specify that a given item must be stressed in this way, I employ a naive representation of this property: every *s-node* is appropriate to the binary-valued feature I-CENTRE. As a rough approximation, its value is usually set positive by a constraint on *s-list*s for the final items – if there is no other item with the positive value, or in the lexicon (usually negative for items that can never occur in the intonation centre). The present constraint sets the item's value of I-CENTRE to positive.

(120) Non-Final Intonation Centre Placement Principle (NFICP):

$$\left(\exists \boxed{9} \exists \boxed{0} \left(\begin{pmatrix} \texttt{append}(\boxed{0}, \boxed{6}, \boxed{1}) \\ \vee \begin{pmatrix} \texttt{append}(\boxed{0}, \boxed{9}[\textit{d-list}], \boxed{1}) \\ \wedge\ \texttt{member}(\boxed{6}, \boxed{9}) \end{pmatrix} \end{pmatrix} \\ \wedge\ \boxed{8}\Big[\text{I-CENTRE } \textit{yes}\Big] \right) \right)$$

If the second *d-node* ($\boxed{6}$) is contextually non-bound, the first instance of the relation `append/3` identifies it with the final member of the top *d-list* ($\boxed{6}$), provided that it is a *d-node*, i.e., the governor of the local tree.[13] If the final member of the top list is a dependent, the second instance of the relation `append/3` finds the final subtree ($\boxed{9}$) and the relation `member/2` its governor

[13]The relation `deep_surf/6` is assumed to pick the two *d-node*s in an order according to DWO, which makes it possible to treat only the second *d-node* ($\boxed{6}$).

($\boxed{6}$). The last conjunct sets the I-CENTRE value of the *s-node* counterpart of $\boxed{6}$ to *plus*.[14]

6.3.6 Dislocation due to surface-level constraints

See 4.3.3.3, 4.3.3.4, and 4.3.3.6. This phenomenon results in a surface order which is not determined by any of the DWO/SWO constraints above. Also, this kind of SWO/DWO mismatch is obligatory. Therefore, if one of the pair of nodes (surface or deep) is subject to this kind of dislocation, none of the SWO/DWO constraints should be applied to the pair. The condition making such pairs exempt from the application of the constraints will be specified in the antecedent of DSOP and the appropriate placement of such items will be determined by constraints on SWO using the items' field specifications.

The problem can therefore be reduced to the following question: what kinds of items are exempt from DSOP? The answer is based on the following hypothesis: DSOP applies, whenever surface-level constraints underspecify an item's position in SWO.

For most cases, the notion of surface-level underspecification can be defined as follows: The SWO position of an *s-node* is underspecified by surface-level constraints if its field within a region can be assigned to other *s-node*s within that region, i.e., if in a region the same field can be assigned to multiple *s-node*s.

Rather than enumerating items exempt from DSOP, it seems reasonable to specify items to which DSOP can apply: DSOP applies only to those items, whose order within a region is not determined by surface-level constraints, which means that their field value within a region is one of those values which can be used repeatedly within that region. An example of such a field would be *rest-fld* within *matrix-fld*.

Yet there can be unique positions, fields which can be assigned only to a single item (including a compacted cluster), which can be filled either by surface-level constraints or by DSOP. Such is the case of *pre-cl-fld*, which can be assigned either to a relative/interrogative expression or to an item according to DSOP.[15] Such fields will be added to the list of multiply fillable fields as those which allow the application of DSOP. Since such a field can be assigned only once, an item with the field specified by surface-level constraints will be placed correctly.

The modified DSOP (121) includes the conditions that both of the two surface nodes must be assigned *rest-fld* or *pre-cl-fld*. On the assumption that these two fields behave in the same way irrespective of their region, the region of these fields is not specified. Thus, this version of DSOP will

[14] The assignment of the I-CENTRE value can be redundant with a general constraint relating focus proper and intonation centre.

[15] I am assuming here that the position of relatives/interrogatives is not determined by DWO.

correctly apply to all items whose position is determined by surface-level constraints.

(121) DEEP/SURFACE ORDER PRINCIPLE (DSOP) version 3:

$$
\begin{array}{l}
\forall\boxed{5}\ \forall\boxed{6}\ \forall\boxed{7}\ \forall\boxed{8} \\
\left(\begin{array}{l}
\left(\begin{array}{l}
\begin{bmatrix} \textit{non-lexical} \\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{TREE}\ \boxed{1} \\ \text{SURFACE}\ \boxed{2} \\ \text{HEAD-DAUGHTER}\ \boxed{3} \\ \text{NONHEAD-DAUGHTERS}\ \boxed{4} \end{bmatrix} \\
\land\ \texttt{deep\_surf}(\boxed{5}, \boxed{6}, \boxed{7}, \boxed{8}, \boxed{3}, \boxed{4}) \\
\land\ \texttt{component}(\begin{bmatrix} (\textit{rest-fld} \lor \textit{pre-cl-fld}) \\ \text{FIELD}\ \boxed{9} \end{bmatrix}, \boxed{7}) \\
\land\ \texttt{component}(\begin{bmatrix} (\textit{rest-fld} \lor \textit{pre-cl-fld}) \\ \text{FIELD}\ \boxed{9} \end{bmatrix}, \boxed{8})
\end{array}\right) \\
\quad \rightarrow \quad \text{IOP} \lor \text{LDTP} \lor \text{NFICP}
\end{array}\right)
\end{array}
$$

The relation `component/2` holds between an object (the second argument) and its subpart (the first argument). Here it is used to make sure that the *s-node*s are assigned *rest-fld* or *pre-cl-fld* within the same region.[16]

The complete DSOP is presented in §A.2.

6.4 General constraints on *s-list*s

These constraints are adopted from Penn (1999b) and Richter (2000) with only a few necessary modifications.[17]

6.4.1 Matrix compaction

All *s-nodes* in every *s-list* must lie within a single *matrix-fld* region, i.e., their FIELD value is or the REGION path terminates in *matrix-fld*, co-indexed with all other *matrix-fld* values within the *s-list*. In other words, in every *s-node* in every *s-list* there is an object *matrix-fld*. Furthermore, in every sentence, all the values *matrix-fld* are a single object.

The following principle expresses exactly that in two statements, using RSRL's restricted quantification over the subparts of an object, i.e., over values of all attribute paths rooted in the object.

[16]It is not possible to specify the *s-node*s simply as objects with *rest-fld* or *pre-cl-fld* as the value of their attribute FIELD: these fields can be embedded more deeply in the structure if they contain other fields.

[17]I am omitting the equivalent of Penn (1999b)'s Relevance principle, which has no linguistic significance. It only makes sure that the REGION paths are not arbitrarily long due recursive regions, but correspond to existing regions.

(122) MATRIX COMPACTION PRINCIPLE (MCP):

$$\textit{s-node} \rightarrow \exists_{\boxed{1}}\ \boxed{1}[\textit{matrix-fld}]$$

$$[\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{DEEP} \mid \text{STATUS}\ \ \textit{unembedded}\ ] \rightarrow \forall_{\boxed{1}}\ \forall_{\boxed{2}} \left(\left(\boxed{1}[\textit{matrix-fld}] \wedge \boxed{2}[\textit{matrix-fld}] \right) \rightarrow \boxed{1} = \boxed{2} \right)$$

6.4.2 Planarity

This is the equivalent of the condition of projectivity: occupants of a multiply-fillable field (such as *rest-fld*) cannot be made discontinuous by some other occupants with a different field specification.

(123) PLANARITY PRINCIPLE:

$$\forall_{\boxed{1}}\ \forall_{\boxed{2}}\ \forall_{\boxed{3}}\ \forall_{\boxed{4}}\ \forall_{\boxed{5}} \left(\begin{pmatrix} \texttt{s\_list}(\langle \boxed{1}, \boxed{2}, \boxed{3} \mid \textit{list} \rangle) \\ \wedge\ \texttt{region}(:, \boxed{4}) \\ \wedge\ \texttt{topo\_field}(\boxed{1}, \boxed{4}, \boxed{5}) \\ \wedge\ \texttt{topo\_field}(\boxed{3}, \boxed{4}, \boxed{5}) \end{pmatrix} \rightarrow \texttt{topo\_field}(\boxed{2}, \boxed{4}, \boxed{5}) \right)$$

The principle in (123) formalizes the notion of planarity by requiring that in every *s-list* with the smallest region $\boxed{4}$ and at least three *s-node*s, where the first and the third *s-node* are assigned the same field $\boxed{5}$ relative to $\boxed{4}$, the middle *s-node* must also be assigned the field $\boxed{5}$.

The principle uses three relations whose definitions are given in §A.3. The relation `s_list/1` simulates the *s-list* type. By letting the principle apply to a list, recursive search through the list is performed by the principle being applied successively from left to right to all sublists with at least 3 members. The relation `region/2` holds about the smallest region (the second argument) that contains the *s-node*s on the *s-list* (the first argument).

The relation `topo_field/3` holds about the relative field (the third argument) of an *s-node* (the first argument) with respect to a region (the second argument).[18]

6.4.3 Topological order

This constraint might be called *primus inter pares* among the constraints on surface word order, as it determines the order of *s-nodes* within a region by using an independently specified order of fields for that region and imposing that order on relative fields.

[18] See above in §5.3.3.2 for the definition of the notions of 'field relative to region' and 'relative field'.

(124) TOPOLOGICAL ORDER PRINCIPLE (TOP):

$$\begin{bmatrix}\text{SURFACE } \boxed{1}\textit{ne-list}\end{bmatrix} \rightarrow \exists_{\boxed{2}}\ \texttt{region}(\boxed{1}, \boxed{2}) \wedge \texttt{topo\_order}(\boxed{1}, \boxed{2})$$

The principle applies to each non-empty *s-list*. The relation `region/2` identifies the smallest region ($\boxed{2}$) common to all *s-node*s within the *s-list* ($\boxed{1}$) and the relation `topo_order/2` applies the appropriate order relative to the region. Note that the relation `region/2` also compacts the items in the *s-list* into the region $\boxed{2}$.[19]

The core of the principle consists in the definition of the relation, which is – due to its complexity – again presented separately in §A.3. The linguistic content of the definition will be presented in a less opaque form as tables below in §6.5.

6.4.4 Field existence

The following constraint makes sure that a certain field is present within a given region.

(125) FIELD EXISTENCE PRINCIPLE (FEP):

$$\begin{array}{l}\left(\begin{array}{l}\begin{bmatrix}\text{SURFACE } \boxed{1}\textit{ne-list}\end{bmatrix}\\ \wedge\ \texttt{region}(\boxed{1}, \boxed{2})\\ \wedge\ \texttt{field\_existence}(\boxed{2}, \boxed{3})\end{array}\right)\\ \rightarrow \exists_{\boxed{4}} \left(\begin{array}{l}\texttt{member}(\boxed{4}, \boxed{1})\\ \wedge\ \texttt{topo\_field}(\boxed{4}, \boxed{2}, \boxed{3})\end{array}\right)\end{array}$$

If there is a sign with a non-empty *s-list* $\boxed{1}$ whose smallest region $\boxed{2}$ satisfies, together with a field $\boxed{3}$, the relation `field_existence/2`, then there must be an *s-node* $\boxed{4}$, which is assigned the field and the region. The relation `field_existence/2` enables to separately enumerate fields which are obligatory relative to a region.[20]

Linguistically, the core of the principle consists again in the definition of the relation, which is presented in §A.3. The definition is obvious once all obligatory fields relative to regions are found. Linguistic content of the definition is presented below in S6.5, together with linear precedence tables.

---

[19] The compacting effect of TOP is the reason why the principle is allowed to apply also to singleton lists, where the ordering effect makes no sense.

[20] This is a generalized version of Penn's principle bearing the same name. The latter is applied to the domain list in complete clauses (main or subordinate) and requires the presence of at least one *pre-cf* object. The present version applies to any syntactic object. Another difference is that the general relation `exist/2` is replaced here with the more restrictive `member/2`.

6.4.5 Field uniqueness

The following constraint makes sure that a certain field is present within a given region at most once.

(126) FIELD UNIQUENESS PRINCIPLE (FUP):

$$\begin{pmatrix} \begin{bmatrix}\text{SURFACE } s\end{bmatrix} \\ \wedge\ \texttt{region}(s,r) \\ \wedge\ \texttt{field\_uniqueness}(r,f_1,f_2) \end{pmatrix} \rightarrow \forall n_1 \forall n_2 \begin{pmatrix} \begin{pmatrix} \texttt{member}(n_1,s) \wedge \texttt{member}(n_2,s) \wedge \neg\, n_1 = n_2 \\ \wedge\ \texttt{topo\_field}(n_1,r,f_1) \\ \wedge\ \texttt{topo\_field}(n_2,r,f_2) \end{pmatrix} \\ \rightarrow\ f_1 = f_2 \end{pmatrix}$$

If there is a non-empty *s-list* with a smallest region r which satisfies, together with the fields f_1 and f_2, the relation `field_uniqueness/3`, then for all pairs of distinct *s-node*s n_1 and n_2 which are on the *s-list*, whose region is r, and whose fields are f_1 and f_2, the fields are identical. The relation `field_uniqueness/3` enables to separately enumerate fields which can occur at most once relative to a region.[21]

The relation `field_uniqueness/3` is presented in §A.3. The principle only makes sense if the second and the third arguments of the relation are instantiated to the same field type. Linguistic content of the relation is given below in §6.5, together with linear precedence tables.

In cases when the principles TOP, FEP and FUP should apply only to parts of *s-list* as the value of SURFACE rather than to the whole *s-list*, I will use a relation `region_setup/2`, which simply invokes the three principles and applies them to the first argument, the list, identifying the second argument with its region (see A.3).

6.5 Some constraints specific to Czech

I have no ambition to describe the subtleties of word order in Czech exhaustively. The constraints below represent only a fraction of what would be needed in a realistic grammar, both in terms of coverage and detail. My aim here is rather modest: to show that the theoretical and formal framework can serve such a purpose. Admittedly, by not including all phenomena and details, there is always the risk of a stumbling block being passed unnoticed.

[21] This is again a generalized version of Penn's principle bearing the same name. The condition on uniqueness can be stated for any *s-list*, therefore it does not need to apply to a whole sign.

First, I will make a few general remarks and then give examples of some common cases of word order as specified by surface-level constraints. Finally, I will show how function words can be ordered.

The order of *s-node*s is specified jointly by surface-level constraints and SWO/DWO constraints (presented above in §6.3). The latter apply whenever the former result in underspecification.

Surface-level constraints impose an order on *s-node*s by using two notions, which are part of grammar: order of fields within a region and compaction of fields to a region. The order of *s-node*s must correspond to the defined order of fields and the way they compact into regions.

The imposition of surface order is made possible by each *s-node* being assigned a relative topological field. The field is specified as a part of a topological region, which in turn can be specified as a field within a higher region. Thus, the position of each field is determined by a path of regions terminating in *matrix-fld*.

The assignment of a field to *s-node* is conditioned by several factors: lexicon, DWO/SWO constraints, and surface-level constraints, including those relating the local derivation tree with the corresponding *s-list*. Whenever possible, I will restrict myself to constraints of a single kind, namely those which specify the order and number of fields within a region, and which are formalized by three principles presented above: the principles of topological order (124), field existence (125) and field uniqueness (126), and definitions of the corresponding relations `topo_order/2`, `field_existence/2` and `field_uniqueness/3` in §A.3. These constraints can be equivalently expressed as tables.

6.5.1 Ordering a simple clause

6.5.1.1 The setup of *matrix-fld*

In the following, I will assume the setup of the top region as in Table 6.1. The fields in the Fields column compact to the region specified in the leftmost Region column, in the order which is indicated in the Order column. The column Occupancy shows how many times a field can occur within the region. It is important to remember that a field may include a number of compacted *s-node*s.

The facts presented informally in this table and other tables below are expressed formally by the principles of topological order – TOP (124), field existence – FEP (125) and field uniqueness – FUP (126) and definitions of the corresponding relations `topo_order/2`, `field_existence/2` and `field_uniqueness/3` in §A.3.[22]

The signature in (127) is the one assumed for the fields in Table 6.1.

[22]The relations are defined only as illustrations for a subset of facts.

Region	Field	Order	Occupancy
matrix-fld	*pre-cl-fld*	1	1
	cl-fld	2	≤1
	rest-fld	3	any
	fin-fld	4	≤1

Table 6.1: Fields within the top region

(127) *field*
matrix-fld
embedded-fld REGION *field*
pre-cl-fld
cl-fld
rest-fld
fin-fld

Note that I am introducing a hierarchy in the signature of topological fields. However, according to (127), all maximal field types except *matrix-fld* are subtypes of *embedded-fld*. A more elaborate hierarchical structure will be introduced later. The signature of fields does not represent the setup of regions, which is the task of the three principles mentioned above and which is informally represented by the tables with facts about order and occupancy.

6.5.1.2 A one-word sentence

I will start with a trivial example of a simple sentence consisting of a single verb, such as:

(128) Hoří!
[it] burns
'Fire!'

The verb emerges from the lexicon without any valency requirement and with a single *s-node*. Its field is lexically underspecified as *pre-cl-fld* or *rest-fld*. Due to the first clause of Matrix Compaction Principle (122), there must be a *matrix-fld* within every *s-node*, and the only possibility for a *matrix-fld* to occur within the *s-node* for *hoří* is as the region of one of the two disjunctive field specifications. There are three additional general constraints on *s-lists* to be satisfied:[23] (i) Topological Order Principle (124), which does not order anything here, but merely states (redundantly) that the region of both *pre-cl-fld* and *rest-fld* is *matrix-fld*, as defined by `topo_order/2`. (ii) The Field Existence Principle (125) applies because the `field_existence/2` relation

[23]Planarity Principle (123) does not apply to an *s-list* with less than three items.

is defined as being satisfied for *matrix-fld* as the first argument and *pre-cl-fld* as the second argument. In effect, the principle requires that there must be a *pre-cl-fld* inside *matrix-fld*. (iii) The Field Uniqueness Principle (126) applies because the `field_uniqueness/3` relation is defined as being satisfied for *matrix-fld* as the first argument and *pre-cl-fld* as the second and third argument. In effect, the principle requires that there must be at most one *pre-cl-fld* inside *matrix-fld*.

The joint effect of MCP, FEP and FUP is that there is exactly one obligatory field inside the obligatory region *matrix-fld*, namely *pre-cl-fld*. Together with the fact that there is only one *s-node* in our *s-list*, the *s-node* of *hoří* will look as follows:

(129) $\begin{bmatrix} \textit{s-node} & \\ \text{PHONOLOGY} & \langle \text{hoří} \rangle \\ \text{FIELD} & \begin{bmatrix} \textit{pre-cl-fld} & \\ \text{REGION} & \textit{matrix-fld} \end{bmatrix} \\ \text{I-CENTRE} & \textit{yes} \end{bmatrix}$

The positive specification of intonation centre is due to the constraint requiring that there must be at least one *s-node* which is part of the intonation centre of an *s-list* (cf. footnote 14 on p. 176). The SWO/DWO constraints do not apply to single-item *s-lists* and the other general SWO constraints apply as intended.

6.5.1.3 A two-words sentence

Extending the trivial example slightly, the sentence in (130) corresponds to an *s-list* consisting of two nodes:

(130) Děti spí.
children sleep
'The children are sleeping.'

The sign representing the sentence obtains essentially by applying the 'backbone constraints'. Due to Surface List Composition Principle (105), its *s-list* consists of two items.

The field of the *s-node* for *spí* is specified in the lexicon in the same way as *hoří*, i.e., as *pre-cl-fld* or *rest-fld*. MCP again requires that the region is *matrix-fld*. However, the *s-node* for *spí* is not the only item on the *s-list*, so the choice between *pre-cl-fld* and *rest-fld* cannot be resolved in the same way.

The field of the *s-node* for *děti* is lexically specified as *noun-fld*, which can compact to the same regions as the verb: *pre-cl-fld* or *rest-fld*.[24] If the

[24] See Tables 6.2 and 6.4 below.

noun were itself modified, then the field *noun-fld* would compact with all its modifiers into a larger *noun-fld* due to the relation `region/2` in TOP (124) and the corresponding relation `topo_order/2`, partially specifying the setup of *noun-fld* (for the complete specification, FUP and FEP must be applied). TOP is also responsible for the trivial compaction of *noun-fld* into either *pre-cl-fld* or *rest-fld*.

According to the first clause of MCP, the top region is again *matrix-fld*. According to its second clause, the *matrix-fld* fields of both *děti* and *spí* are the same object, i.e., the *s-node*s compact within the top region.

There are two possibilities: either Identical Order Principle (117) applies with *spí* as the intonation centre and focus proper, or Non-Final Intonation Centre Placement Principle (120) applies with *děti* as the intonation centre and focus proper. In either of them *pre-cl-fld* must be filled by *děti*, due to Phonology Principle and the properties of the field *pre-cl-fld*. Because this field accommodates exactly one occupant (as required by FEP and FUP), the middle-level region in *s-node* for *spí* must be assigned the only remaining option: *rest-fld*.

The *s-list* corresponding to the first possibility with *spí* as the intonation centre and focus proper is shown in (131).[25]

(131)
$$\left\langle \begin{bmatrix} \textit{s-node} \\ \textsc{phonology}\ \langle \textit{děti} \rangle \\ \textsc{field} \begin{bmatrix} \textit{noun-fld} \\ \textsc{r} \begin{bmatrix} \textit{pre-cl-fld} \\ \textsc{r}\ \boxed{1}\ \textit{matrix-fld} \end{bmatrix} \end{bmatrix} \\ \textsc{i-centre}\ \textit{no} \end{bmatrix}, \begin{bmatrix} \textit{s-node} \\ \textsc{phonology}\ \langle \textit{spí} \rangle \\ \textsc{field} \begin{bmatrix} \textit{rest-fld} \\ \textsc{r}\ \boxed{1} \end{bmatrix} \\ \textsc{i-centre}\ \textit{yes} \end{bmatrix} \right\rangle$$

A very similar picture can be shown if the verb has more dependents. Each dependent of any word class is eventually assigned via TOP two possible regions: *pre-cl-fld* or *rest-fld*. The initial position is licensed by the item being either topic proper or focus proper.

There is one more field optionally available within *matrix-fld*, which is situated to the right of *rest-fld*: *fin-fld*. This field is used for extraposed and/or phonologically heavy dependents, such as embedded clauses, and can be multiply filled. Being exempt from DSOP, this field is assigned by construction-specific constraints.

6.5.2 Going deeper: nominal groups

We have seen one possible non-verbal filler of *rest-fld* or *pre-cl-fld*, namely an item which is assigned the field *noun-fld*. Such an item may be a single *s-node*, as in (131), or more *s-node*s, which are compacted into a single

[25]The attribute REGION is abbreviated as R.

region of *noun-fld*. Thus, items which consist of a single *s-node*, or which consist of multiple compacted *s-node*s, correspond to what is sometimes called *phenogrammatical constituents*.[26]

Table 6.2 shows fields which can occur in regions *pre-cl-fld* and *rest-fld*.

Region	Field
pre-cl-fld $\vee$ *rest-fld*	*noun-fld*
	pp-fld
	adj-fld
	adv-fld
	emb-cls-fld

Table 6.2: Fields for dependents, version 1

There are no Order and Occupancy columns in the table, because it is difficult to represent the relevant facts within the format: only a single field may occupy *pre-cl-fld*, while in *rest-fld*, where multiple fields can occur, no order and occupancy limit are stipulated. In order to represent the facts more consistently, the signature of fields in (127) can be extended as in Fig. 6.7.

```
field
    matrix-fld
    embedded-fld        REGION field
        pre-cl-fld
        cl-fld
        rest-fld
        fin-fld
        dep-fld
            noun-fld
            pp-fld
            adj-fld
            adv-fld
            inf-fld
            aux-fld
            emb-cls-fld
                sconj-cls-fld
                wh-cls-fld
```

Figure 6.7: Signature of topological fields, version 2

Note that *dep-fld* also subsumes *inf-fld* (for infinitival clauses), whose setup will be presented below in Table 6.5 on p. 200, and *aux-fld*, which is lexically assigned to non-clitic auxiliaries, esp. the future auxiliary and

[26] See Penn (1999a) for a brief history of the term.

modals. Now the setup of *pre-cl-fld* and *rest-fld* can be stated more concisely as in Table 6.3.

Region	Field	Order	Occupancy
pre-cl-fld	*dep-fld*	1	1
rest-fld	*dep-fld*	any	any

Table 6.3: Fields for dependents, version 2

Organizing fields as sister subtypes in a hierarchy is useful whenever the fields may occur in the same context, yet differ in their internal structure.

Table 6.4 shows a possible setup of *noun-fld*. The region consists of a single obligatory field *noun-fld*, which is flanked on both sides by a number of optional fields. There is only a single left field for 'light adjectival phrases', consisting of adjectives premodified by adverbs. Following the governor, a range of field types is allowed, including postmodifying nouns and 'heavy adjectival phrases'. Several fields are potentially recursive, although recursion is restricted by syntax.

Each word class has its standard lexically specified field assignment: *noun-fld*, *adj-fld*, *adv-fld*, *prep-fld*. Lexical items usually compact with members of the same syntactic paradigms into regions bearing an identical name. In either case, the fields identify constituency of surface strings. Strings corresponding to adjectival groups are classified as 'light' (which are allowed to premodify) and 'heavy'.

According to what has been presented so far, the only way of compacting fields is by applying the general constraints on *s-list*s to the value of the attribute SURFACE. This means that only *s-list*s corresponding to signs may be compacted this way.

For example, due to flat derivation structure, *pp-fld* cannot be compacted according to Table 6.4, because the *s-list* corresponding to the PP would have to include exactly two items, a *prep-fld* and a *noun-fld*, in that order. This can happen only if the 'prepositional object' is a bare noun. In all other cases there is no way of compacting the nominal group following the preposition, because *noun-fld* would have to be a region common to all *s-node*s in the *s-list*. It is not, because of the preposition. However, as will be shown below in §6.5.5, the fact that a PP cannot compact in the way other syntactic units do is in fact desirable.

The signature including the new subtypes of *field* is now as in Fig. 6.8.

The following two points summarize the tools for imposing surface order proposed so far:

1. Deep/Surface Order Principle, which mediates between deep and surface word order. This principle applies only to pairs of content words

Region	Field	Order	Occupancy
noun-fld	*l-adj-fld*	1	any
	noun-fld	2	≥1
	emb-cls-fld	3	≤1
	h-adj-fld	≥4	any
	pp-fld	≥4	any
pp-fld	*prep-fld*	1	1
	noun-fld	2	1
l-adj-fld	*adv-fld*	1	any
	adj-fld	2	1
h-adj-fld	*adv-fld*	1	any
	adj-fld	2	1
	pp-fld	≥3	any
	emb-cls-fld	≥3	any
	noun-fld	≥3	any
adv-fld	*adv-fld*	1	≤1
	adv-fld	2	1

Table 6.4: Fields for nominal groups

in a local tectogrammatical tree whose corresponding *s-node*s are assigned *rest-fld* or *pre-cl-fld* within the region corresponding to the tree. The principle relates deep word order with the surface order of compacted items including such *s-node*s, taking into account the property of being intonation centre.

2. The principles of Topological Order, Field Existence and Field Uniqueness apply to *s-list* as a value of SURFACE. They impose an order on fields (phenogrammatical constituents) relative to the lowest region common for all *s-node*s within the *s-list*.

These tools interact in order to determine word order in regular cases where all subtrees are realized continuously. The following sections will be concerned with some constructions involving discontinuities.

6.5.3 Discontinuity: a bigger problem than one might think

If it were the case that members of a syntactic paradigm always compact, then – with a suitable arrangement of syntactic rules – the surface string could be simply read off the leaves of the derivation tree and no *s-list* would be necessary. Unfortunately, there are a number of phenomena involving discontinuous realization of syntactic paradigms (see §4). I will start with a simple example (54), repeated here as (132).

field
- *matrix-fld*
- *embedded-fld* REGION *field*
  - *pre-cl-fld*
  - *cl-fld*
  - *rest-fld*
  - *fin-fld*
  - *dep-fld*
    - *noun-fld*
    - *pp-fld*
    - *adj-fld*
      - *l-adj-fld*
      - *h-adj-fld*
    - *adv-fld*
    - *inf-fld*
    - *aux-fld*
    - *emb-cls-fld*
      - *sconj-cls-fld*
      - *wh-cls-fld*

Figure 6.8: Signature of topological fields, version 3

(132) menší vesnice než Lhota
smaller village than Lhota
'a smaller village than Lhota'

The adjectival group *menší než Lhota* is split by the governor of the whole noun group. The field assignments are as follows: *adj-fld* for *menší*, *noun-fld* for *Lhota*, and *sconj-compar-fld* for *než*. The region of *noun-fld* should compact according to the specification in (133) with *sconj-compar-fld* into *compar-base-fld*.[27]

(133)

Region	Field	Order	Occupancy
compar-base-fld	*sconj-compar-fld*	1	1
	noun-fld	2	1

Similarly as in the case of *pp-fld*, compaction according to (133) cannot occur unless *noun-fld* is a single *s-node*. In order to make the compaction possible, I am going to introduce the following rather ad-hoc constraint, which is used

[27]The region *compar-base-fld* should also cover bases of comparison governed by other parts of speech. If the base of comparison is a clause, its word order interacts with the presence of the conjunction and the whole region should be viewed as a subtype of *sconj-cls-fld* and the conjunction be placed at a position equivalent to *sconj-fld*. I will leave this for further research, as well as the question whether there are also comparative adverbial groups made discontinuous by a verb.

to compact only this specific region.[28]

(134) Base-of-Comparison Compaction (BCC):

$$\left(\begin{bmatrix}\text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CATEGORY}\,|\,\text{HEAD } \textit{noun}\\ \text{NONHEAD-DAUGHTERS } \boxed{1}\\ \text{SURFACE } \boxed{2} \bigcirc \boxed{3}\end{bmatrix} \atop \wedge\ \texttt{member}\left(\boxed{0}\begin{bmatrix}\text{SURFACE } \boxed{3}\left\langle\left[\text{FIELD } \textit{sconj-compar-fld}\right]\right\rangle\end{bmatrix}, \boxed{1}\right)\right)$$
$$\rightarrow \exists_{\boxed{4}}\ \texttt{region\_setup}(\boxed{2}, \boxed{4}\textit{noun-fld})$$

The constraint applies to signs with a non-head daughter ($\boxed{0}$) whose *s-list* is a singleton list with an *s-node* corresponding to the conjunction ($\boxed{3}$). Then all the other *s-node*s of the *s-list* ($\boxed{2}$) compact to *noun-fld*.

The resulting *s-list* consists of two compacted items, *sconj-compar-fld* and *noun-fld*, which is exactly the setup of *compar-base-fld* as required in (133). This means that the two items compact into *compar-base-fld* due to the general SWO principles.[29] The result is shown in (135).

(135)
$$\left\langle \begin{bmatrix}\textit{s-node}\\ \text{P } \langle \textit{než} \rangle\\ \text{F } \begin{bmatrix}\textit{sconj-compar-fld}\\ \text{R } \boxed{1}\ \textit{compar-base-fld}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\textit{s-node}\\ \text{P } \langle \textit{Lhota} \rangle\\ \text{F } \begin{bmatrix}\textit{noun-fld}\\ \text{R } \boxed{1}\end{bmatrix}\end{bmatrix} \right\rangle$$

If *menší* (= *adj-fld*) is not specified as being able to compact to a region which would include *než Lhota* (= *comp-base-fld*, such a region could be *h-adj-fld*), then it trivially compacts to *l-adj-fld* as its single field, and then it can compact to *noun-fld* as a premodifier of *vesnice*.

The string *než Lhota* should also compact to the same region *noun-fld*, which is what happens if *compar-base-fld* is included as a field of that region. The line for *compar-base-fld* in (136) extends Table 6.4.

(136)

Region	Field	Order	Occupancy
noun-fld	...	...	...
	compar-base-fld	≥ 4	≤ 1
	...	...	...

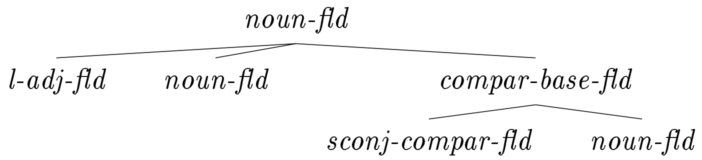
[28] In syntax, a corresponding lexically derived valency for a conjunction of this type is also needed.

[29] It would be possible to compact *sconj-compar-fld* and *noun-fld* into *compar-base-fld* only by (134), without invoking the general principles. This could be done by extending the consequent: the attribute REGION of the fields *sconj-compar-fld* and *noun-fld* should be identical and equal to *compar-base-fld*.

(137)
$$\left\langle \begin{matrix} \begin{bmatrix} \textit{s-node} \\ \textrm{P}\ \langle \textit{menší} \rangle \\ \textrm{F} \begin{bmatrix} \textit{l-adj-fld} \\ \textrm{R}\ \boxed{1}\ \textit{noun-fld} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} \\ \textrm{P}\ \langle \textit{vesnice} \rangle \\ \textrm{F} \begin{bmatrix} \textit{noun-fld} \\ \textrm{R}\ \boxed{1} \end{bmatrix} \end{bmatrix}, \\ \begin{bmatrix} \textit{s-node} \\ \textrm{P}\ \langle \textit{než} \rangle \\ \textrm{F} \begin{bmatrix} \textit{sconj-compar-fld} \\ \textrm{R}\ \boxed{2} \begin{bmatrix} \textit{compar-base-fld} \\ \textrm{R}\ \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} \\ \textrm{P}\ \langle \textit{Lhota} \rangle \\ \textrm{F} \begin{bmatrix} \textit{noun-fld} \\ \textrm{R}\ \boxed{2} \end{bmatrix} \end{bmatrix} \end{matrix} \right\rangle$$

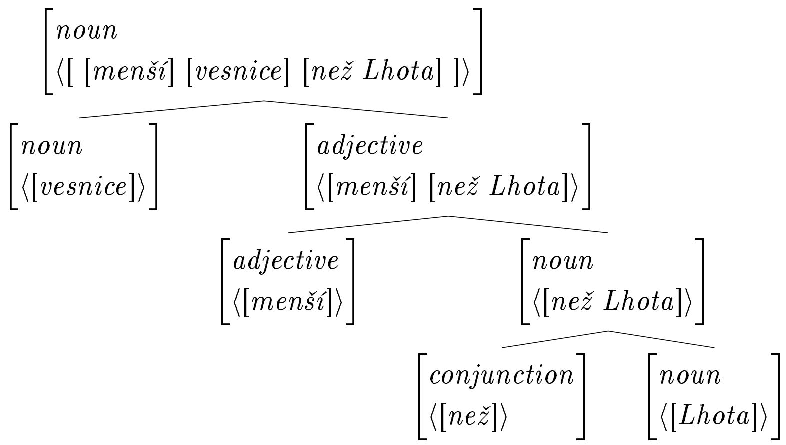
In (138) the *s-list* for the whole noun group is shown. The hierarchy of fields is represented as a tree in (139).

(138)



Finally, (139) shows the derivation tree with *s-lists*. Compaction of *s-nodes* is denoted by brackets. Note that in the sign standing for the adjectival group *menší než Lhota* the adjective does not compact with the noun.

(139)



The remaining issue concerns a variant of (132) shown in (140). Both variants are equally acceptable.

(140) vesnice menší než Lhota
village smaller than Lhota
'a village smaller than Lhota'

Here no discontinuity arises, so *menší* should compact with *než Lhota*. However, using fields in the way shown above, the inclusion of *compar-base-fld* as a field of the region *h-adjp-fld* would make the other variant impossible, because *compar-base-fld* would always compact with *adj-fld* before it could become a field of the region *noun-fld*.

Instead, I will introduce a disjunctive constraint on the way *adj-fld* is combined with *compar-base-fld*. The constraint can be depicted schematically as (141):

(141) $[\mathit{adj\text{-}fld}\ \mathit{compar\text{-}base\text{-}fld}] \Rightarrow [\mathit{h\text{-}adj\text{-}fld}] \vee [\mathit{l\text{-}adj\text{-}fld}\ \mathit{compar\text{-}base\text{-}fld}]$

In words, if *adj-fld* is followed by *compar-base-fld*, then either both compact to *h-adj-fld*, or the former to *l-adj-fld* and the latter stays intact. Now *h-adj-fld* can be specified as including the field *compar-base-fld*, as in (142).

(142)

Region	Field	Order	Occupancy
h-adj-fld	...	...	...
	compar-base-fld	≥ 3	≤ 1
	...	...	...

The disjunctive constraint is formalized as (143).

(143) Adjective and Base-of-Comparison Compaction (ABCC):

$$\left(\begin{array}{l}[\text{SURFACE}\ \boxed{1}]\\ \wedge\ \mathtt{append}(\boxed{2}, \boxed{3}, \boxed{1})\\ \wedge\ \mathtt{region}(\boxed{2}, \boxed{4}\mathit{adj\text{-}fld})\\ \wedge\ \mathtt{region}(\boxed{3}, \boxed{5}\mathit{compar\text{-}base\text{-}fld})\end{array}\right) \rightarrow \left(\begin{array}{l}\mathtt{region}(\boxed{1}, \mathit{h\text{-}adj\text{-}fld})\\ \vee\ \exists_{\boxed{6}} \left(\begin{array}{l}\boxed{4}\begin{bmatrix}\mathit{l\text{-}adj\text{-}fld}\\ \text{REGION}\ \boxed{6}\mathit{noun\text{-}fld}\end{bmatrix}\\ \wedge\ \boxed{5}[\text{REGION}\ \boxed{6}]\end{array}\right)\end{array}\right)$$

In each sign whose *s-list* ($\boxed{1}$) consists of two lists: $\boxed{2}$ compacting to *adj-fld* and $\boxed{3}$ compacting to *compar-base-fld*, either the whole *s-list* compacts to *h-adj-fld* or $\boxed{2}$ and $\boxed{3}$ compact in the next higher region to *noun-fld* ($\boxed{6}$), while $\boxed{2}$ is specified as *l-adj-fld* ($\boxed{4}$) and $\boxed{3}$ remains *compar-base-fld* ($\boxed{5}$). The order of the two items ($\boxed{2}$ and $\boxed{3}$) in the resulting *s-list* is determined by the setup of *h-adj-fld* or $\boxed{6}$*noun-fld*.

The *s-list* corresponding to (140) is shown in (144).

(144)

$$\left\langle \begin{array}{l} \begin{bmatrix} \textit{s-node} \\ \text{P}\ \langle \textit{vesnice} \rangle \\ \text{F}\ \begin{bmatrix} \textit{noun-fld} \\ \text{R}\ \boxed{1}\,\textit{noun-fld} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} \\ \text{P}\ \langle \textit{menší} \rangle \\ \text{F}\ \begin{bmatrix} \textit{l-adj-fld} \\ \text{R}\ \boxed{2}\begin{bmatrix} \textit{l-adj-fld} \\ \text{R}\ \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \\ \begin{bmatrix} \textit{s-node} \\ \text{P}\ \langle \textit{než} \rangle \\ \text{F}\ \begin{bmatrix} \textit{sconj-compar-fld} \\ \text{R}\ \boxed{3}\begin{bmatrix} \textit{compar-base-fld} \\ \text{R}\ \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{s-node} \\ \text{P}\ \langle \textit{Lhota} \rangle \\ \text{F}\ \begin{bmatrix} \textit{noun-fld} \\ \text{R}\ \boxed{3} \end{bmatrix} \end{bmatrix} \end{array} \right\rangle$$

The hierarchy of fields is shown in (145).

(145)

- *noun-fld*
  - *noun-fld*
  - *l-adj-fld*
    - *l-adj-fld*
    - *compar-base-fld*
      - *sconj-compar-fld*
      - *noun-fld*

6.5.4 Long-distance dependencies

Similarly as English, Czech allows extraction from clauses embedded to an arbitrary depth. The usual approach in constraint-based grammars is to use a mechanism identifying the 'gap' with its 'filler', or a valency requirement with its satisfier positioned at a certain (usually initial) position. This 'slash feature passing' approach has been supported by empirical facts from some languages (cf. Bouma, Malouf, and Sag (1998)), showing that constructions from which something is extracted exhibit a special kind of behaviour, which is accounted for in a natural way exactly by the slash feature mechanism.

While this approach to long-distance dependencies could be adopted here as well, I will rather explore the alternative of ordering only *s-node*s, while leaving all other features of the expression in place.

I have already suggested that examples of topicalization could be solved by Left Dislocation of Topic Principle (119) on p. 175, a component of DSOP.[30] If wh- expressions, whose clause- or sentence-initial position is obligatory, are not subsumed by topicalization, then they should be treated here.

A region common to relative and wh- interrogative clauses (*wh-cls-fld*) has already been introduced. Its setup is given below (146).[31]

[30] Although the principle can only assign *pre-cl-fld* to an item which is present as a node in the local tectogrammatical tree, once the field is assigned, the item can be compacted to a higher region.

[31] It seems that a common region type for wh- clauses, both relative and wh- interrog-

(146)

Region	Field	Order	Occupancy
wh-cls-fld	*pre-cl-fld*	1	1
	cl-fld	2	≤1
	rest-fld	3	≥1
	fin-fld	4	≤1

Note that the only point in which the setup of *wh-cls-fld* differs from that of *matrix-fld* is the occupancy of *rest-fld*. As a subtype of *emb-cls-fld*, *wh-cls-fld* covers all embedded finite clauses without subordinating conjunction. If wh-expression is assigned *pre-cl-fld*, a wh- question with the wh- expression in the initial position of the whole sentence is – from the topological viewpoint – indistinguishable from the main clause, and therefore it will be treated as *matrix-fld*.

With the setup of *wh-cls-fld* defined as above, the field of every relative or interrogative expression can be specified as *pre-cl-fld*.[32] Because such an item can only occupy a *pre-cl-fld* field, it should 'climb' through *s-list*s of embedded syntactic units until it reaches its proper place.[33]

In order to release a *pre-cl-fld* while letting the rest of the *s-list* compact according to the tables, the following constraint is necessary:

ative, cannot be used, because they have different distribution. However, the distinction should be a matter of syntax rather than surface word order. Since relative and wh- interrogative clauses must be properly embedded, there is no risk of a relative expression compacting to an interrogative clause and vice versa.

[32]Here I am assuming the use of non-local attributes REL and QUE, and the necessary grammar principles in order to relate the presence of an embedded wh- element with the assignment of *pre-cl-fld* to the whole expression, as in (i).

(i) Přišel ten pán, [s manželem dcery jehož první ženy] jsi
came the gentleman with husband daughter$_{gen}$ whose first wife$_{gen}$ AUX-2SG
včera mluvila.
yesterday talked
The gentleman came with whose first wife's daughter's husband you spoke yesterday.

Furthermore, I assume here and below that the setup of *pre-cl-fld* does not prevent the region from accommodating other occupants than subtypes of *dep-fld*. Probably the setup of *pre-cl-fld* will eventually have to be specified negatively, with only a few fields excluded: *cl-fld*, *rest-fld* and *fin-fld*. However, *pre-cl-fld* should still hold only a single occupant, unless a specific conditions are satisfied (as in the case of multiple fronting).

[33]Restrictions on the the placement of wh- expressions in a relative or wh- interrogative clause are supposed to be the matter of syntactic rather than SWO constraints.

(147) PARTIAL COMPACTION, version 1:

$$\left(\begin{array}{l}[\textsc{surface}\ \boxed{1}]\\ \wedge\ \mathtt{\textbf{shuffle}}(\boxed{2}, \boxed{3}, \boxed{1})\\ \wedge\ \mathtt{\textbf{region}}(\boxed{2}, \textit{pre-cl-fld})\end{array}\right) \rightarrow \exists_{\boxed{4}} \left(\begin{array}{l}\mathtt{\textbf{region\_setup}}(\boxed{3}, \boxed{4})\\ \vee\ \mathtt{\textbf{region}}(\boxed{1}, \boxed{4})\end{array}\right)$$

The constraint applies to every sign in whose *s-list* ($\boxed{1}$) there is an item ($\boxed{2}$) compacted to *pre-cl-fld*. Then the remaining part of *s-list* ($\boxed{3}$) compacts to some region $\boxed{4}$.

In Fig. 6.9 the structure of fields corresponding to (148) is shown as a tree. For space reasons, the field types are abbreviated.

(148) Koho jsi říkal, že Marie myslela, že Pavel pozve?
who AUX-2SG said that Mary thought that Paul invites
'Who did you say Mary thought Paul would invite?

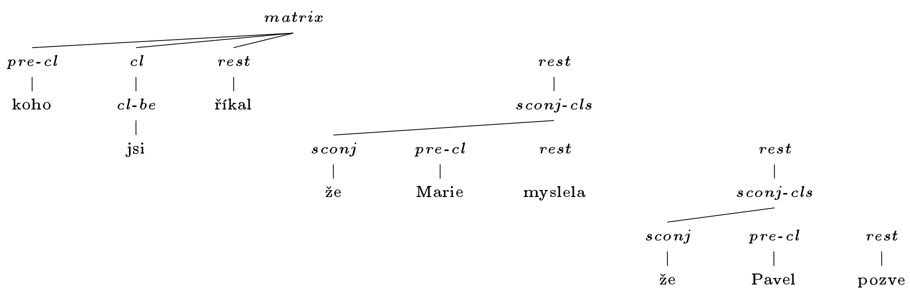


Figure 6.9: Fields in a sentence with long-distance dependency

6.5.5 More discontinuities: split PPs

Similarly as Polish and Serbo-Croatian, Czech allows splitting of PPs. This occurs most easily if the item immediately following the preposition is an interrogative or relative expression, as in (149b). As (149a) shows, the splitting is optional.[34]

(149) a. O jakou soutěž se jedná?
about what competition REFL is talked about
'What kind of competition is it?'

[34] The cases without preposition, as in (i), are considered as cases of general long-distance dependency.

(i) Jakou myslíš soutěž?
what think-SG2 competition
'What kind of competition do you have in mind?'

b. O jakou se jedná soutěž?
about what REFL involves competition

The interrogative or relative item can be embedded, as shown in (150b). There seems to be a condition that the expression is a dependent of the governing noun in the final position; this condition is not satisfied in (150c).[35] Finally, (150d) shows that the expression need not include an interrogative/relative item, although in such a case the expression must be stressed in order to improve its acceptability.

(150) a. O jak dotovanou soutěž se jedná?
about how financed competition REFL is talked about
'How financed competition is it?'
b. O jak dotovanou se jedná soutěž?
about how financed REFL is talked about competition
c. *O jak se jedná dotovanou soutěž?
about how REFL is talked about financed competition
d. ?O velmi dobře dotovanou se jedná soutěž.
about very well financed REFL is talked about competition

'It is a very well financed competition.'

Additionally, the split PP can be subject to unbounded dependency, as in (151).[36]

(151) O jakou jsi myslel, že se jedná
about what AUX-2SG thought that REFL is talked about
soutěž?
competition
'What kind of competition did you think it was?'

Solutions for similar examples from Polish and Serbo-Croatian have been presented by Kupść (2000, §2.4.2) (see Fig. 5.7) and Penn (1999b). In the example in Fig. 5.7 on p. 139 (*w dużym mieszka domu*), the preposition is compacted with the following item (an adjective) only when the whole PP becomes a part of a clause and the noun is free to be ordered independently. Penn (1999b) uses a principle applying to signs for NPs and PPs with disjunctive statements, which compact the domain objects of the phrase either to *pre-cf* (pre-clitic field) or to *rf* (rest field) in the clause. According to the third option the first *prosodic word*[37] compacts to *pre-cf* and the rest

[35] The unacceptability of (150c) was observed by Karel Oliva (p.c.).
[36] Sentences where the noun is more deeply embedded are still grammatical, but it is difficult to find some which do not sound awkward.
[37] Prosodic word is identified by using a parallel structure for prosodic constituency, with a separate list of domain objects. Domain objects can be compacted into prosodic

to *post-cf* (post-clitic field) of the next higher region, a matrix clause or an embedded finite clause.

As the present proposal concerning surface order is founded on Penn's approach, I will consider only his solution, which can be adopted with a few modifications:

1. Domain objects should not be compacted too soon, otherwise the item following preposition would not be available for compaction with the preposition. This condition is satisfied if either no region is defined which consists of a preposition and a nominal group, or if such a region cannot be built, as in our case, where the flat derivation structure does not allow for compacting a *noun-fld* with a preceding preposition, except when the noun is bare. Such a region can only be formed by one of the disjuncts in the principle. On the other hand, because of the flat structure, compaction of only a part of the nominal group without the preposition does not prevent compaction of the preposition with a following segment into *pre-cl-fld*.

2. According to Penn (1999a), long-distance dependencies are handled by a modified slash feature mechanism. Thus, the initial segment of the PP can compact in the next higher clause, together with the rest of the PP. However, here we need to underspecify the position of the initial segment, similarly as the position of wh- expressions.

3. In Czech, the equivalent of *post-cf* (post-clitic field) seems to be rather the clause-final field, if any.

4. The final point concerns the role of prosody. Examples in (150) suggest that – unlike in Serbo-Croatian – the expression following preposition is a syntactic rather than prosodic unit. Thus, if there is any involvement of prosodic factors here at all, it is restricted to procliticizing preposition to the syntactic unit.[38]

The solution should be in line with the approach pursued so far, which was based solely on defining the setup of regions. Similarly as in the case of discontinuous adjectival group above, it is possible to define two alternative regions: *pre-cl-fld* as the PP-initial region or *pp-fld* as the region compacting preposition with the nominal group. With the initial region compacting into *pre-cl-fld*, the remainder part can be assigned *fin-fld*, which would position the noun at the end of the clause by surface-level rules, or *noun-fld*, which would compact to *rest-fld* and determine its position by DSOP (121). If

constituents which correspond to prosodic words.

[38]Recall that I assume no separate representation of prosodic structure. Admittedly, this makes it difficult to distinguish prosodic and syntactic factors responsible for a specific phenomenon.

the noun is the rightmost dependent of the verb, IOP (117) would place its surface counterpart correctly as the last *rest-field*. In the following formalization of PP compaction I will assume that the position of the remainder part of PP is not fixed to *fin-fld*.[39]

(152) PP COMPACTION:

$$
\left(\begin{array}{l}
\begin{bmatrix}
\text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CATEGORY}\,|\,\text{HEAD } \textit{noun} \\
\text{NONHEAD-DAUGHTERS } \boxed{1} \\
\text{SURFACE } \boxed{2}\langle[\text{FIELD } \textit{prep-fld}]\rangle \bigcirc \boxed{3} \bigcirc \boxed{4}
\end{bmatrix} \\
\wedge\ \texttt{member}([\text{SURFACE } \boxed{2}], \boxed{1}) \\
\wedge\ \texttt{member}([\text{SURFACE } \boxed{3}], \boxed{1})
\end{array}\right)
$$

$$
\rightarrow \exists_{\boxed{5}} \left(\begin{array}{l}
\left(\begin{array}{l}
\texttt{append}(\boxed{2}, \boxed{3}, \boxed{5}) \\
\wedge\ \texttt{region}(\boxed{5}, \textit{pre-cl-fld}) \\
\wedge\ \texttt{region\_setup}(\boxed{4}, \textit{noun-fld})
\end{array}\right) \\
\vee \left(\begin{array}{l}
\texttt{shuffle}(\boxed{3}, \boxed{4}, \boxed{5}) \\
\wedge\ \texttt{region\_setup}(\boxed{5}, \textit{noun-fld})
\end{array}\right)
\end{array}\right)
$$

If there is a sign headed by *noun* with a non-head daughter whose singleton *s-list* ($\boxed{2}$) contains an *s-node* corresponding to preposition, and with another non-head daughter whose *s-list* is $\boxed{3}$, then either these two daughters ($\boxed{5}$) compact to *pre-cl-fld* and the rest of the mother's *s-list* ($\boxed{4}$) compacts to *noun-fld*, or the second daughter ($\boxed{3}$) compacts with the rest to *noun-fld*.[40]

As stated, the *pre-cl-fld* can occur within the current finite clause. In order to let it being extracted to a higher clause, the Partial Compaction constraint (147) must allow 'uncompacting' of *pre-cl-fld*s from the current finite clause, which is precisely what it does.

6.5.6 More fields

There are some more fields needed in addition to those presented so far. One area which needs to be investigated in more detail are verbal complexes and non-finite verb projections. Non-finite verb projections are good candidates for receiving fields of their own so that clitics which do not climb the whole way into the clitic field of a finite clause could be ordered properly. Then, if

[39]This solution could be supported by the marginal acceptability of (i), with the noun being positioned non-finally, which might suggest that a surface-level rule insisting on the final position is not involved here.

(i) ??O jakou se soutěž jedná?
about what REFL involves competition

[40]Unfortunately, because of the flat derivation tree, the second daughter cannot be specified with respect to its position within the hypothetical *noun-fld*, so the preposition can be compacted with any non-head daughter.

it is the case that items in verbal complexes compact, a dedicated field for both finite and non-finite verb projections is needed.

In order to account for infinitival clauses, I will introduce a new field (and region) *inf-fld*, which is lexically assigned to infinitives. To find an adequate setup of the field, I will examine word order variations in an embedded infinitival clause.

In (153) the clitic is in the initial position of the infinitival clause, the verb can be positioned arbitrarily (indeed, with FSP effects).[41] Note that the infinitive may be preceded or followed by all of its dependents, or only some of them.

(153) a. Je otrava *si* pobrukovat celý den stejnou písničku.
is bore REFL purr whole day same song
'It is boring to purr the same song all day long'
b. Je otrava *si* celý den pobrukovat stejnou písničku.
c. Je otrava *si* celý den stejnou písničku pobrukovat.

The same infinitival clause cannot precede the main clause, because the clitic cannot occur in a sentence-initial position (154).

(154) a. **Si* pobrukovat celý den stejnou písničku je otrava.
b. **Si* celý den pobrukovat stejnou písničku je otrava.
c. **Si* celý den stejnou písničku pobrukovat je otrava.

Other orderings are possible if the clitic precedes or immediately follows the infinitive, as in (155) and (156).

(155) a. Je otrava pobrukovat *si* stejnou písničku celý den.
b. Je otrava celý den *si* pobrukovat stejnou písničku.
c. Je otrava celý den *si* stejnou písničku pobrukovat.

(156) a. Pobrukovat *si* stejnou písničku celý den je otrava.
b. Celý den *si* pobrukovat stejnou písničku je otrava.
c. Celý den *si* stejnou písničku pobrukovat je otrava.

A clitic may not follow the infinitive non-adjacently (157a) and (158a).

(157) a. *Je otrava pobrukovat stejnou písničku *si* celý den.
b. ?Je otrava celý den pobrukovat *si* stejnou písničku.

[41]The clitic cannot climb from the infinitival clause into the main clause:

(i) a. *Je *si* otrava pobrukovat celý den stejnou písničku.
is REFL bore purr whole day same song
(intended:) 'It is boring to purr the same song all day long'
b. Tomáš *si* musí pobrukovat celý den stejnou písničku.
Tom REFL must purr whole day same song
'Tom can't stop purring the same song all day long.'

c. Je otrava celý den stejnou písničku *si* pobrukovat.

(158) a. *Pobrukovat stejnou písničku *si* celý den je otrava.
b. ?Celý den pobrukovat *si* stejnou písničku je otrava.
c. Celý den stejnou písničku *si* pobrukovat je otrava.

It may seem according to (159) that a clitic may not be positioned sentence-finally, but the true reason for the unacceptability consists again in that a clitic may not follow the infinitive, if it is not adjacent to it.[42]

(159) a. *Je otrava pobrukovat stejnou písničku celý den *si*.
b. *Je otrava celý den pobrukovat stejnou písničku *si*.
c. ??Je otrava celý den stejnou písničku pobrukovat *si*.

As (160c) and (161) show, again the clause-final position of the clitic is not the reason why (160a) and (160b) is out. The reason is that the clitic does not precede the infinitive nor follows it immediately.

(160) a. *Pobrukovat stejnou písničku celý den *si* je otrava.
b. *Celý den pobrukovat stejnou písničku *si* je otrava.
c. ?Celý den stejnou písničku pobrukovat *si* je otrava.

(161) Vyklánět se je nebezpečné.
to lean out REFL is dangerous
'It is dangerous to lean out.'

Finally, the examples in (162) show that there is a strong preference for clitics to either precede or follow the infinitive, rather than both precede and follow.

(162) a. Vrátný nám slíbil si ho obstarat hned zítra.
porter us promised REFL it provide immediately tomorrow
'The porter has promised us to provide it for himself first thing tomorrow.'
b. Vrátný nám slíbil obstarat si ho hned zítra.
c. ?*Vrátný nám slíbil si obstarat ho hned zítra.
d. ?*Vrátný nám slíbil ho obstarat si hned zítra.

The region of *inf-fld* could therefore be defined as in Table 6.5.[43]

[42] Cf. (i).

(i) Je snadné usmát se.
is easy smile REFL
'It is easy to make a smile.'

[43] This specification of *inf-fld* contradicts Planarity Principle, because in between items assigned *rest-fld* an item assigned a different field may occur. The problem can be solved by introducing three new fields corresponding to the three positions of *rest-fld* in *inf-fld* as subtypes of *rest-fld*, along with a fourth filed covering the other options. Since this

Region	Field	Order	Occupancy
inf-fld	*rest-fld*	1	any
	cl-fld	2∨5	≤1
	rest-fld	3	any
	inf-fld	4	1
	rest-fld	6	any

Table 6.5: The region of infinitival clause

The optionality of *cl-fld* enables optional clitic climbing according to §6.5.7.3.

Finally, it is necessary to add the constraint prohibiting clitics from standing at the very beginning of a sentence, irrespectively of their level of embedding.

(163) CLITICS NOT FIRST

$$\begin{bmatrix} \text{SYNSEM} \mid \text{LOCAL} \mid \text{DEEP} \mid \text{STATUS} & \textit{unembedded} \\ \text{SURFACE} \ \langle \boxed{1} \mid \boxed{0} \rangle & \end{bmatrix} \\ \rightarrow \neg \ \boxed{1}[\text{FIELD} \mid \text{REGION} \ \textit{cl-fld}]$$

6.5.7 Function words

In this section, I will return again to the overview of mismatches between surface and deep order, this time to the part concerning function words (§4.3.4).[44]

6.5.7.1 Function words adjacent to their hosts

See §4.3.4.1. The example (56), repeated here as (164), suggests that there is a region of deverbal nouns, which always includes a clitic field immediately following the noun itself.

(164) otcovo zachmuření se
father's frowning REFL
'father's frown'

Other examples of this kind can be solved in a similar way.

issue has no consequences elsewhere, I will keep ignoring it.

[44] Certain phenomena involving some function words (e.g., prepositions) have already been discussed in the preceding sections.

6.5.7.2 Function words within local domain

See 4.3.4.2. The only difference as compared to the previous case is in a different specification of the order for the given field. This applies to prepositions, conjunctions (both compacting in the initial position to the region of their host, except for the phenomenon of PP splitting), past and conditional auxiliaries (compacting to the nearest clitic field), as well as other clitics hosted by a finite verb.

6.5.7.3 Function words outside local domain

See §§4.3.4.3, 4.3.4.4, and 4.3.4.5. A typical example of this sort is clitic climbing. If the *s-node* of a clitic (a reflexive particle) is lexically specified as in (165), then the clitic should compact with other clitics in the region of a finite clause if it does not compact within a *inf-fld* along the path from the governor up to the finite clause.

(165) $$\begin{bmatrix} \textit{s-node} \\ \text{PHONOLOGY } \langle \textit{se} \rangle \\ \text{FIELD } \begin{bmatrix} \textit{cl-rfl-fld} \\ \text{REGION } \textit{cl-fld} \end{bmatrix} \\ \text{I-CENTRE } \textit{no} \end{bmatrix}$$

Clitic climbing must be allowed by extending the Partial Compaction constraint to clitics. For clitics, as compared with *pre-cl-fld*s, an additional restriction is needed, as clitics may not climb beyond a finite clause.[45]

(166) PARTIAL COMPACTION, version 2:

$$\begin{pmatrix} [\text{SURFACE } \boxed{1}] \\ \wedge\ \texttt{shuffle}(\boxed{2}, \boxed{3}, \boxed{1}) \\ \wedge \begin{pmatrix} \texttt{region}(\boxed{2}, \textit{pre-cl-fld}) \\ \vee \begin{pmatrix} \texttt{region}(\boxed{2}, [\text{REGION } \textit{cl-fld}]) \\ \wedge\ \neg\ [\ \text{S} \mid \text{L} \mid \text{C} \mid \text{A-MORPH } \textit{finite}]) \end{pmatrix} \end{pmatrix} \end{pmatrix}$$
$$\rightarrow \exists_{\boxed{4}} \begin{pmatrix} \texttt{region\_setup}(\boxed{3}, \boxed{4}) \\ \vee\ \texttt{region}(\boxed{1}, \boxed{4}) \end{pmatrix}$$

Clitic climbing is optional: the clitic may either stay within the clitic region of its host or climb only half way to the clitic region of a less deeply embedded infinitive. In order to allow for these various options, the clitic field in

[45]The string S|L|C stands for SYNSEM|LOCAL|CATEGORY. The attribute A-MORPH stands for "analytical morphology". An item may have a different form as a separate word and as a complex form interpreted together with function words. Thus, an item with the head value specified as *infinitival* may have the attribute A-MORPH set to *finite*. (I am assuming that the type *head* is partitioned roughly as proposed by Przepiórkowski (1999a, p. 420).)

infinitives is optional (see Table 6.5 on p. 200). The clitic is then positioned in a clitic region nearest to its host.

Closely connected with clitic climbing is the phenomenon of haplology of reflexive particles and pronouns. I will postpone the discussion of this issue to the section devoted specifically to clitics climbing and haplology (§7.3).

6.5.7.4 Adjacency violation due to a split node

See 4.3.4.6. In order to properly describe the position of the future auxiliary in (63), repeated here as (167), it is necessary to represent the fact that it is the tense grammateme of posteriority, which is the only non-bound element in the sentence.

(167) Chodit já tam <u>budu</u>.
to go I there AUX
'I WILL be going there.'

I have not provided an appropriate signature for representing the property of contextual boundness specific to grammatemes, so I will only sketch a possible solution.

For content words, SWO and prosodical regularities conditioned by DWO and the CB property are formalized by constraints relating *d-list* and *s-list*, more specifically, their members corresponding to content words. By operating on pairs of corresponding deep and surface nodes, an item which must be ordered according to DWO and CB rather than by surface-level constraints must be on both lists. This means that the function word *budu* should be represented by a node of its own in the tectogrammatical tree, if its CB or DWO properties differ from those if its host.

This solution is not very appealing for at least two reasons: (i) it does not correspond to the theoretical premises about tectogrammatical representation, and (ii) it introduces an otherwise unmotivated distinction in a very important representational issue.

Another possibility would presuppose that even DWO/CB-conditioned word order regularities can be stated from *s-node*s. This would mean that *s-node*s would have to include tectogrammatical information and that the horizontal order of nodes in the tectogrammatical tree would not be relevant for SWO: the effects of DWO on SWO would be a part of *s-list*-specific constraints. This solution seems to be even less satisfactory.

Finally, there might be a way of relating not only *d-node*s but also grammatemes with their corresponding *s-node*s, while imposing an appropriate DWO-based ordering and/or prosodic marking. I will leave this issue for further research.

6.6 Conclusions

This chapter started with the definition of general principles on the setup of non-lexical signs, *d-list*s and *s-list*s, including constraints of the DWO/SWO relation. Next, a number of surface-level constraints specific to Czech have been suggested.

The empirical facts and theoretical premises presented in §4 and elsewhere have been formalized in a way which allows for the interaction of factors determining surface word order, where FSP determines surface ordering when it is underspecified by other constraints. This is achieved by applying a few general and some construction-specific constraints to recursive topological fields, tectogrammatical representation and – in some cases – also to morphosyntactic properties of the signs involved.

Even though there were a few examples included, the chapter was rather formal. The following chapter will shift the balance to the empirical side.

Chapter 7

A closer look at clitics

Word order properties of Czech clitics is a phenomenon which defies straightforward solutions. There are at least two sources of the complexity:

(i) Several factors – syntactic, prosodical, discourse-based, stylistic – interact to determine their position. The integration of constraints originating in different corners of a language system should be supported by an appropriate theoretical and formal framework.

(ii) Only some generalizations concerning their ordering behaviour can be expressed by strict rules, while many other properties have to be stated as mere preferences.[1] To model preferences or tendencies by the symbolic apparatus of theoretical linguistics is not easy and will not be attempted here.[2]

The coverage of clitics in modern descriptive grammars of Czech is rather sketchy (Šmilauer, 1966, p. 67–68), (Dokulil et al., 1986, p. 154–156), (Daneš, Grepl, and Hlavsa, 1987, p. 604–605); a slightly more detailed description is presented in (Karlík, Nekula, and Rusínová, 1995, p. 648–651). As the attention to the phenomenon has been growing in recent years, all of the above accounts have been shown to suffer from various inconsistencies and omissions. At the same time, the complexity of the topic has become obvious (Avgustinova and Oliva, 1995; Esvan, 2000; Svoboda, 2000; Toman, 2000; Uhlířová, 2001; Toman, 2001; Svoboda, 2001; Oliva, 2001).

I will tackle the difficult issue only to the extent necessary to explain its integration into the framework, and the linguistic data – although not trivial – will be presented mainly in order to show the interaction of constraints

[1] Phenomena which can be described by strict rules include the order of reflexives, auxiliaries and personal pronouns of distinct cases within a single clitic cluster. Phenomena which can be described only by preferential statements include clitic climbing or the specification of the second (Wackernagel) position.

[2] The need for a formal framework better suited to the reality of language and linguistic methodology has been voiced recently by Sgall (2001).

governing the deep and surface word order. I will deal mainly with the issue of clausal clitics (i.e. 2P clitics, 2nd or Wackernagel position clitics) and I will assume 'clitics' to mean clausal (or sentential) clitics, unless explicitly said otherwise.

First, I will identify relevant Czech clitics, introducing the usual distinction between constant and inconstant clitics. Next, the question of what actually the second position is will be asked and some answers suggested. In the following, clitic climbing and haplology will be discussed and the issue concerning the order of clitics within a clitic cluster will be raised.

Most topics in this section consist of a data presentation part followed by an attempt to integrate the phenomenon into the formal description. Admittedly, these attempts are sometimes of a speculative nature and not very conclusive. This is partially due to the inconclusive evidence of the phenomena, partially to the insufficient range of descriptive tools available.

The section is concluded by a summary of observations – a checklist of phenomena with specifications of their status in the formal description.

7.1 Czech clitics identified

The claim that Czech clitics will be identified here may sound foolhardy or even presumptuous: to define the class of Czech clitics is an issue in itself. Nevertheless, for the purpose of the subsequent exercise of stating their word order properties, I will need at least some definition, and I will therefore adopt a rough distinguishing criterion.

Czech clitics are usually characterized as unstressed words which form larger prosodical segments with their immediately preceding stressed neighbour, the first stressed item in the clause. In other words, they are characterized as clausal (sentential) enclitics to the initial stressed item.[3]

However, to define Czech clitics as dependent and unstressed items is not correct. It is easy to argue, e.g. with Esvan (2000, p. 144) or Uhlířová (2001, p. 144), against such a definition on the grounds of examples such as (168a) and (168b), where an uncontroversially clitical element does bear some stress, with the conclusion that Czech clausal clitics must be defined by their 'position within the clause', rather than by their inability to bear stress.[4]

[3]Karlík, Nekula, and Rusínová (1995, p. 647) remark (emphasis of the authors): "The rhythm of a sentence is influenced mainly by the **position** of monosyllabic, less frequently disyllabic unstressed **clitics**."

[4]In the following examples, relevant clitics are set in italics. This may be important, since for space reasons, only some examples are glossed in this chapter.

(168) a. Prosím vás, v tom zmatku, co tam je, *se* *mu* chlapec
please you in the turmoil that there is REFL him boy
klidně mohl ztratit.
easily could lose
'Well, in the turmoil there he might easily have lost the boy.'

b. Tohle, jak víte, *jsem* *se* *vám* snažil povědět už
this as know AUX-1SG REFL to you tried say already
včera.
yesterday
'As you know, I tried to tell you this already yesterday.'

In fact, in such examples (see also (181c) below) clitics occur after a pause and tend to procliticize with the following expression. It is not even necessary for a clitic to be part of a larger prosodical segment, as in (169).

(169) My všichni, co spolu chodíme, *bychom*, jak říká Zilvar z
We all, that together walk, AUX, as says Zilvar from
chudobince, měli držet za jeden provaz. [KOCL]
poorhouse, should hold by one rope.
'As Zilvar from the poorhouse says, all of us friends should be thick as thieves.'

Czech clausal clitics can therefore be only very roughly characterized as words occupying the '2nd position'. More will be said later about the nature of the 2nd position, called equivalently 'clitical environment', 2P, the post-initial or Wackernagel position. This position can be filled by more than one clitic: if there are more clitics within the same clause, originating within the clause or within embedded clauses, they form a continuous cluster.

What kinds of words can be considered as candidates to clitic-hood? There are some word forms which occur only in the clitical environment, their traditional name being **constant clitics**. Many other forms can stand either in the clitical environment or elsewhere, and they are traditionally called **inconstant clitics**.

Among Czech clitics, a prominent group is represented by a subset of forms of personal pronouns. The so-called weak forms are all constant clitics: they cannot be stressed or coordinated and cannot occur outside the clitical environment. The sentences in (170) work as a test to classify dative personal pronouns as constant clitics, inconstant clitics and non-clitics (strong forms). The pronouns occupying the 2nd position in (170a) are either constant or inconstant clitics. The same position in (170b) is unacceptable for non-clitics. The focussed pronouns in (170c) are either inconstant clitics or non-clitics. The same position is unacceptable for constant clitics (170d). The test shows that many forms of dative personal pronouns are inconstant clitics, while only four dative pronouns are constant clitics (170d).

(170) a. Dáša *mně/mi/ti/jí/mu/nám/vám/jim/si* ho poslala doporučeně.

b. *Dáša *tobě/jemu/sobě* ho poslala doporučeně.
c. Dáša ho poslala doporučeně jenom mně/tobě/jí/jemu/nám/vám/jim/sobě.
d. *Dáša ho poslala doporučeně jenom mi/ti/mu/si.

Table 7.1 on p. 209 lists all relevant forms of personal pronouns.[5]

Since none of the constant clitics can be placed outside the 2nd position and most of them are monosyllabic and cannot bear stress, their status as a clitic is beyond doubt. On the other hand, it is much easier to challenge the status of inconstant clitics. Because I am interested in the investigation to what extent the framework is able to cope with various word order phenomena, I will adopt a rather rough attitude and admit as a clitic all forms which can be placed within a clitic cluster. Then, the only difficulty is in finding the end of a cluster. The strategy will be to treat as clitics only those forms which satisfy at least one of the following conditions: (i) a clitic cannot occur outside the 2nd position, or (ii) a clitic can be followed by another clitic within a clitic cluster.

7.1.1 Constant clitics

None of the types below occurs outside the clitical environment.[6]

- conjunction *-li* 'whether'[7]
- forms of *být* as the past tense auxiliary: *jsem, jsi, jsme, jste*;

[5] With a preposition, a non-weak form is used. After a preposition, the initial *j-* is replaced with palatalized *n-*: *jí* $\rightarrow$ *bez ní*, *jemu* $\rightarrow$ *k němu*.

[6] Additionally, Karlík, Nekula, and Rusínová (1995, p. 649) present the past conditional auxiliary forms of *být* as 2P constant clitics: *byl, byla, bylo, býval, bývala, bývalo*. I am not ready to accept those without further evidence, because the form in (i) is clearly not in the 2nd position:

(i) Pavel *by* včera *byl* rád přišel.

These forms could be treated as inconstant clitics, but examples such as (ii) do not provide very good evidence that *byl býval* is placed inside a clitic cluster, so I will assume that these forms are not clitics.

(ii) Já *bych mu to tam byl býval přece* koupil. [KAR:649]

Note that in the above example I am marking by italics (Karlík, Nekula, and Rusínová, 1995)'s assumption about what constitutes the clitic cluster, including the 'modification particle' *přece*.

[7] On the basis of examples such as (i), Avgustinova and Oliva (1995) argue that *-li* is not a clausal clitic, but rather a word clitic:

(i) Lásce-*li* své *se* v žití budeš protiviti, žebrákem půjdeš světem.

NUMBER	*sg*					
PERSON	*1st*	*2nd*	*3rd*			
GENDER			*f*	*mi*	*ma*	*n*
nom	já	ty	ona	on		ono
gen	mne mě	**tebe** *tě*	jí	*ho*	**jeho** *ho*	
dat	mně *mi*	**tobě** *ti*	jí	*mu*	**jemu** *mu*	
acc	mne mě	**tebe** *tě*	ji	*ho*	**jeho** *jej* *ho*	je *ho*
ins	mnou	tebou	jí	jím		

NUMBER	*pl*					
PERSON	*1st*	*2nd*	*3rd*			*refl*
GENDER			*f,mi*	*ma*	*n*	
nom	my	vy	ony	oni	ona	–
gen	nás	vás	jich			**sebe** ?*se*
dat	nám	vám	jim			**sobě** *si*
acc	nás	vás	je			**sebe** *se*
ins	námi	vámi	jimi			sebou

Table 7.1: Forms of personal pronouns used as constant clitics (weak forms, set in *italics*), only as non-clitics (strong forms, set in **bold**), and as either clitics or non-clitics (inconstant clitics, set in plain type).

- forms of *být* as the present conditional auxiliary: *bych, bys, by, bychom, byste*;[8]

- *s* – contracted form of *být* as the past tense auxiliary, as the periphrastic passive auxiliary or as the copula;[9]

- the forms *se* and *si* as reflexive particles or reflexive pronouns (respectively accusative/genitive and dative)

- weak forms of personal pronouns: *mi, ti, tě, mu, ho* (see Table 7.1)

7.1.2 Inconstant clitics

- those forms of personal pronouns which are neither weak nor strong (see Table 7.1)

  See also (254) and (253) below for examples involving pronouns in the nominative and instrumental case within a clitic cluster. In fact, it is not clear whether pronouns in the instrumental case do behave as clitics, because they occur at the end of the clitic cluster with no constant clitics clitics following them. However, I will assume that they are inconstant clitics, by analogy with the other pronominal forms.

- non-negated forms of *být* as the periphrastic passive auxiliary and as the copula: *jsem, jsi, je, jsme, jste, jsou*;[10]

- *však* as a conjunction 'however'

However, in (ii) *-li* would be difficult to interpret as a word clitic.

(ii) Vstanu a obléknu-*li* se, je tím vyčerpán můj příděl energie pro zbývající den. [MD:27/01,p.15]

I will assume that at least in some cases *-li* is a clausal clitic.

[8] The forms *abych, abys, abychom, abyste* and *kdybych, kdybys, kdybychom, kdybyste* are contractions of a subordinating conjuction and the conditional auxiliary and occupy the initial position.

[9] By an orthographical convention the form *s* is attached to the preceding host: *namalovalas to hezky*, *kterous myslel?*, *napsal sis to?*. An apostrophe can be inserted, cf. [SVO(12,13)]: *Dědovi's to tam neměl dnes nosit.* and *Té holce, co byla včera na večírku, 's to tam neměl dnes nosit.*

[10] I will assume that copula is an inconstant clitic in the light of the examples below (i).

(i) a. Jedinou radostí *jsou mu* dopisy z domova. [SYN]
  b. Já *jsem ti* tuhle spokojeně doma, a najednou ... [KOCL]
  c. Nakonec *je ti ho* skoro líto.
  d. A teď *je ho tam* taková spousta. [SYN]
  e. To *je mu* podobný. [SYN]

Note that most of the examples involve extracting of the pronoun from an embedded constituent into the clausal clitic cluster.

Here, the crucial example is (171c), where *však* is a part of the clitic cluster.

(171) a. Tomáš *se ho však* nebojí.
b. *Tomáš *se však ho* nebojí.
c. Tomáš *však se ho* nebojí.
d. *Však *se ho* Tomáš nebojí. (však ≠ vždyť)
e. *Tomáš *se ho* nebojí však.
f. Včera *se ho* Tomáš však naštěstí nebál.

Example (171d) is out only when *však* is used as a conjunction, not in its particle reading, in which it can be replace by *vždyť*.

- the "short" adverbs *už* 'already' and *prý* 'allegedly'

  Unlike *tu*, a constant clitic, these two adverbs can occur outside the 2nd position, but they satisfy our test of clitic-hood in that they can be followed within a clitic cluster by another clitic:

(172) a. Tomáš *se ho už/prý* nebojí.
b. Tomáš *se už/prý ho* nebojí.
c. Tomáš *už/prý se ho* nebojí.
d. Už/Prý *se ho* Tomáš nebojí.
e. *Tomáš *se ho* nebojí už/prý.
f. Včera *se ho* Tomáš už/prý naštěstí nebál.

7.1.3 Candidates of clitic-hood ruled out

Karlík, Nekula, and Rusínová (1995) include the following forms as inconstant clitics, but according to our test they are ruled out.

- the "short" adverb *tu* 'here'

  Abstracting away from some Czech dialects, *tu* seems to be a weak adverb with a strong variant *tady*, similarly as some personal pronouns, and it could even be considered as a constant clitic.[11] However, not only that the examples below do not show convincingly that *tu* is a constant clitic, they do not provide enough evidence for *tu* as an inconstant clitic.

(173) a. Tomáš *se ho tu* nebojí.
b. *Tomáš *se tu ho* nebojí.

[11] As a constant clitic occurring at the end of a clitic cluster, *tu* would be useful in tests: items preceding *tu* up to the left boundary of the cluster would be – by our criterion – clitics. As a result, a number of additional clitics would pass the test of clitic-hood, such as the pronoun *to* and pronominal PPs.

c. *Tomáš *tu se ho* nebojí.
d.??Tu *se ho* Tomáš nebojí.
e. *Tomáš *se ho* nebojí tu.
f. ??Včera *se ho* Tomáš tu naštěstí nebál.

Consider also (174):

(174) Ona *ti s tím dnes prý tu už* nepomůže.

- pronoun *to* 'it'

  Examples (175a)–(175f) do not attest *to* as a clitic. Only (175g) would make the difference, if *tu* would be a constant clitic.

(175) a. Věra *se mu to* snaží rozmluvit.
b. *Věra *se to mu* snaží rozmluvit.
c. *Věra *to se mu* snaží rozmluvit.
d. To *se mu* Věra snaží rozmluvit.
e. Věra *se mu* snaží rozmluvit *(právě) to.
f. Včera *se mu* Věra to snažila rozmluvit.
g. Věra *se mu to tu* snaží rozmluvit.

- personal pronouns in prepositional cases

  Again, if *tu* were a constant clitic, (176e) would attest the status of pronominal PPs as clitics. As a result, even non-lexical items would have to be made eligible to clitic-hood.

(176) a. Dana *se mi* s ním včera pochlubila.
b. *Dana *se s ním mi* včera pochlubila.
c. *Dana *s ním se mi* včera pochlubila.
d. Dana *se mi tu s ním* včera pochlubila.
e. Dana *se mi s ním tu* včera pochlubila.

- The short adverbs *tam* 'there', *teď* 'now', and *tak* 'so' do not pass the test, because they do not seem to accept a position within the cluster:

(177) a. Tomáš *se ho* tam/teď nebojí.
b. *Tomáš *se tam/teď ho* nebojí.
c. *Tomáš *tam/teď se ho* nebojí.
d. Tam/Teď *se ho* Tomáš nebojí.
e. Tomáš *se ho* nebojí tam/teď.
f. Z nějakého důvodu *se ho* Tomáš tam/teď naštěstí nebál.

(178) a. Tomáš *se ho* tak nebojí.
b. *Tomáš *se tak ho* nebojí.

c. *Tomáš *tak se ho* nebojí.
d. Tak *se ho* Tomáš nebojí.
e. ??Tomáš *se ho* nebojí tak.
f. ??Včera *se ho* Tomáš tak naštěstí nebál.

- "modification particles" *vlastně* 'in fact, actually' and *přece* 'still'

(179) a. Tomáš *se ho* vlastně/přece nebojí.
b. *Tomáš *se vlastně/přece ho* nebojí.
c. *Tomáš *vlastně/přece se ho* nebojí.
d. Vlastně/Přece *se ho* Tomáš nebojí.
e. *Tomáš *se ho* nebojí vlastně/přece.
f. Včera *se ho* Tomáš vlastně/přece naštěstí nebál.

7.1.4 Summary

In this section, the class of clitics has been defined for the purpose of this work. Clitics, as defined above, will be formally distinguished from other items by specifying their field attribute. An example of an *s-node* for the reflexive particle *se* was already given in (89), and is repeated here as (180).

(180)
$$\begin{bmatrix} \textit{s-node} & \\ \text{PHONOLOGY} & \langle \textit{se} \rangle \\ \text{FIELD} & \begin{bmatrix} \textit{cl-rfl-fld} & \\ \text{REGION} & \begin{bmatrix} \textit{cl-fld} & \\ \text{REGION} & \textit{matrix-fld} \end{bmatrix} \end{bmatrix} \\ \text{I-CENTRE} & \textit{no} \end{bmatrix}$$

The region *cl-fld* consists of a number of fields. These fields are assigned to lexical entries of the appropriate clitics. The assignments and order of fields within *cl-fld* will be specified below.

Inconstant clitics will receive two lexical entries: one which is assigned a field within the region *cl-fld*, and another which is assigned a field in the usual way, i.e., according to its word class.

7.2 What is the second position?

This question amounts to asking what the first position is, i.e., what kinds of items can precede a clitic cluster. The range of options is listed below according to syntactic structure of the initial expressions. However, this partitioning should not imply that only syntactic factors are responsible.

7.2.1 A complete subtree in the initial position

This is the trivial option – the initial item is a maximal projection, in other words a complete subtree or phrase.[12] The subtree may be governed by a dependent of the finite verb governing the clause, of an embedded verb (182b), or of a noun (189) and (190). The initial item – a *wh-* or 'topicalized' item – is often subject to unbounded dependency.

Local dependent The initial item can be a single word (181a), a phrase (181b) or a clause (181c) (all governed by a dependent of the main verb).

(181) a. Eva *se* dívá na televizi. [DAN:604]
Eva RFL watches on TV.
'Eva is watching TV.'
b. Proud teplého vzduchu na okraji balkánské výše *se* dostává do střední Evropy. [DAN:604]
c. Kde Sokrates skutečně zemřel, *se* zřejmě nikdy nedovíme. [DAN:604]

Coordination also counts as a complete subtree:

(182) a. Bolek a Lolek *se* po dlouhé době zase objevili na obrazovce.
b. Kam má jít a co má vyřídit, *jsem mu* zapomněl připomenout.

A complete subtree (§7.2.1) is placed in the first position when it compacts to *pre-cl-fld* of the finite clause. This is achieved most easily when the subtree itself is a constituent of the clause, as in (181).

Whenever there is no item which is assigned *pre-cl-fld* 'obligatorily' by a surface-level constraint or the components of Deep/Surface Order Principle (121) different from Identical Order Principle (117), the latter interacts with Topological Order Principle (124) in assigning *pre-cl-fld* to the least dynamic dependent node in the local tree. If all items in the subtree governed by this node compact, the initial position is occupied by a complete subtree.

Compaction is trivial in (181a). The initial nominal group in (181b) compacts to *noun-fld* due to TOP (see Table 6.4). The wh- clause in (181c) can occur in the initial position due the fact that finite clauses are also a subtype of *dep-fld*s 6.7, which can compact either to *rest-fld*s or to a *pre-cl-fld* (Table 6.3). Compaction of finite clauses is also due to TOP.

It is previewed that coordination should be treated as a compacted subtree with the category of the conjuncts visible to most other constraints.

Conjunctions in the initial position Most coordinating conjunctions (*a, ale, ano, avšak, ba i, kdežto, leč, než, nýbrž*) do not count as the initial

[12]including a trivial case corresponding to a single terminal node

item (183a), while with subordinating conjunctions (complementizers) and relative expressions there is a choice (183b) and (183c).[13]

(183) a. Pomohl jsem jí do vlaku a přes okno *jsem se s ní* rozloučil.
b. Nikoho nenapadlo, že *by sis tam* nakonec mohla zvyknout.
c. Nikoho nenapadlo, že nakonec *by sis tam* mohla zvyknout.

Dependent clauses introduced by a subordinating conjunction (complementizer) are covered by the region *sconj-cls-fld*, which is a subtype of *emb-cls-fld*. If its setup is defined as in Table 7.2, then the field *pre-cl-fld* is optional and both (183b) and (183c) are accounted for.

Region	Field	Order	Occupancy
sconj-cls-fld	*sconj-fld*	1	1
	pre-cl-fld	2	≤1
	cl-fld	3	≤1
	rest-fld	4	any
	fin-fld	5	≤1

Table 7.2: The region of dependent clause with subordinating conjunction

Non-finite verbs The initial item can also be an infinitival clause (184)–(184c) or maximal projection of a passive participle (184d).

(184) a. Rozdávat ženám květiny *se* před volbami musí.
b. Číst dětem pohádku *se mi* dnes opravdu nechce.
c. Pohlídat děti Novákům *si* troufne jen tetička z Plzně.
baby-sit kids Nováks-DAT RFL ventures only auntie from Pilsen
'Only the auntie from Pilsen is brave enough to baby-sit the Nováks' kids.'
d. Obdarovány květinami *si* některé dámy být nepřály.

Past participles are not acceptable as the head of the initial item:

(185) *Rozdával ženám květiny *jsem* skoro denně.

[13]In addition to the initial (first), and post-initial (second) positions, Svoboda (2001) uses three other positions: pre-initial, medial and final. The pre-initial position accommodates items such as the complementizer in (183c), as well as coordinating conjunctions and other unstressed words forming a prosodical unit with the following expression in the initial position (*ať, tak, copak*), and also other items, such as *no přece* in (212). I will not discuss this position here, but simply assume that whether this position is filled or not, the post-initial (2nd) position follows the initial position.

Similarly as in finite clauses, the order of items in *rest-fld* is determined by DSOP. The integration of an infinitival clause into a higher clause is made possible by treating *inf-fld* as a subtype of *dep-fld*.

However, the introduction of *inf-fld* solves only cases where the infinitive depends on an autosemantic finite verb, such as (184c). Examples (184a) and (184b) involve a modal verb, which is treated as a function word and a non-head daughter in the flat derivation tree. At the same time, the modal verb is a sister to the head daughter, the infinitive. It can be seen that there is no single region covering the infinitive with some of its tectogrammatical dependents, which would correspond to *pre-cl-fld*. Unless a constraint of the partial compaction type applies, only regions corresponding to individual daughters can be assigned *pre-cl-fld*.

What is needed is a constraint compacting the infinitive and some of its dependents to *pre-cl-fld*, rather than assigning one of those items to *pre-cl-fld* and the rest to *rest-fld*. It means another case of partial compaction:[14]

(186) Initial Non-Finite Verb Compaction I

$$
\left(\begin{array}{l}
\left[\begin{array}{l} \textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{h} \left[\begin{array}{l} \textit{infinitival} \vee \textit{pass-pple} \\ \textsc{f-word}\ \textit{no} \\ \textsc{s-node}\ \boxed{1} \end{array}\right] \\ \textsc{nonhead-daughters}\ \boxed{2} \\ \textsc{surface}\ \boxed{3} \end{array}\right] \\
\wedge\ \texttt{append}(\boxed{4}, \boxed{5}, \boxed{3}) \\
\wedge\ \texttt{member}(\boxed{1}, \boxed{4}) \\
\wedge\ \texttt{member}\left(\left[\textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{h} \left[\begin{array}{l} \textit{verb} \\ \textsc{f-word}\ \textit{yes} \\ \textsc{s-node}\ \boxed{6} \end{array}\right] \right], \boxed{2}\right) \\
\wedge\ \texttt{member}(\boxed{6}, \boxed{5})
\end{array}\right)
$$
$$
\rightarrow \left(\begin{array}{l} \texttt{region}(\boxed{4}, \textit{pre-cl-fld}) \\ \vee\ \texttt{region}(\boxed{3}, (\textit{matrix-fld} \vee \textit{emb-cls-fld})\ ) \end{array}\right)
$$

In every sign with an infinitive or passive participle (see (184d)) as the autosemantic head and a synsemantic verb as a non-head daughter, with an *s-list* that can be non-deterministically partitioned into two sublists – $\boxed{4}$, including the *s-node* of the infinitive, and $\boxed{5}$, including the *s-node* of the finite verb – the smallest common region of $\boxed{4}$ is *pre-cl-fld*, or the region of the whole sign is a finite clause.

This constraint interacts with DSOP in the expected way: only infinitives which can be positioned initially for communicative reasons can be positioned

[14] S|L|C|H stands for SYNSEM|LOCAL|CATEGORY|HEAD.

there together with some of their dependents.[15]

Now consider (184d). The most obvious difference is in the verbal form in the initial position. This has already been reflected in (186), where the form of the head verb is specified as a disjunction. However, there is a structural difference as well: the fronted participle is a part of the passive infinitive *být obdarovány*, itself a dependent of the finite verb *nepřály*. The constraint (186) applies to the sign for embedded passive infinitival clause before it is combined with the higher finite verb. This results in the passive participle with the dependent *květinami* being assigned *pre-cl-fld*, which can only be positioned in the higher clause.

On the other hand, example (185) with past participle and some of its dependents in the initial position is correctly disallowed. Sentences such as (187) where bare past participle (or other autosemantic verbal form) occurs in the initial position are licensed in another way: DSOP handles the *s-node* for the head non-finite verb form as a *s-node* corresponding to any other *d-node* in the local tree.

(187) Viděl *jsi* někdy něco podobného?

Function words The phenomenon of some function words occurring in the initial position is very common. This applies to modal verbs and *být* as the future and passive auxiliary (188).

(188) Budu *si* to pamatovat.

The most natural account would be based on their communicative function, subjecting them to DSOP in the same way as content words. However, this presupposes a solution to the issue of split nodes (see the discussion in §6.5.7.4 on p. 202). Without such a solution, the initial field type is optionally assigned to the auxiliary, if non-clitic auxiliaries are assigned a field which is a subtype of *dep-fld*. I will leave this issue for further research.

Wh- dependents of a noun So far, the governor of the initial expression was a dependent of a verb. In (189a) the initial item is an interrogative pronoun and in (189b) a relative pronoun, both dependents of a noun.

(189) a. Jaká *se vám* vybaví představa?
b. Vážně nevím, kterou *si* vybral nevěstu.

Interrogative and relative expressions feature in this pattern with relative ease, which seems to be related to their status as the most dynamic part of the sentence. As already suggested in §6.5.4 on p. 192, interrogative and relative expressions expressions are lexically assigned *pre-cl-fld*, which makes

[15] As stated, the constraint allows also for some cases of 'partial fronting' – see (195) and (195) below.

them eligible for partial compaction (166) and 'liberation' into the higher *s-list*. If the preference for the final position of the governors is not due to the interaction with DWO, then a version of partial compaction constraint must be applied to assign not only the initial field to the wh- expression, but also the final field (*fin-fld*) to the noun.

Other dependents of a noun In (190) the initial items are acceptable only under sentential stress and the preferred position of the noun is at the end of the clause.

(190) a. PĚKNÁ *se* nám vybaví představa.
b. BOHATOU *si* vybral nevěstu.
c. TAKOVÝ *se mi* líbí básník.

Unlike Serbo-Croatian, which routinely allows prosodical words in the initial position (except when they are part of an idiom), Czech is not so permissive, even if the initial items bear sentential stress:

(191) a. *TENHLE *mi* člověk slíbil peníze.
b. *VYSOKÝ *mi* slíbil peníze člověk.

I am not ready to provide an explanation for why (190) is plausible and (191) not.

Furthermore, it can be observed in the variants of (192a) that modifiers of a noun are acceptable in the initial position under contrastive stress and that postmodifiers (192c) seem to be more acceptable in that position than premodifiers (192b).[16]

(192) a. Obrázkovou knížku o Praze *jsem mu* dal už minule.
b. ?**Obrázkovou** *jsem mu* dal knížku o Praze.
c. **O Praze** *jsem mu* dal obrázkovou knížku.

The examples in (190) can be accounted for if the fronted adjectives receive the initial field assignment. Then the clause-initial positioning already follows due to the Partial Compaction constraint (166). The natural way of assigning *pre-cl-fld* is again by invoking the communicative function of the item. It can be either the only item in focus in the whole sentence, or a special kind of focus proper, or perhaps contrastive topic. Any of these properties, if represented within *d-node*, may be reflected by the assignment of *pre-cl-fld*. Since in the current setup the special kind of focus proper and contrastive topic are not represented, I will leave this issue aside, together with the subtle contrast in (192).

[16] See (206) below for parallel examples with partial trees in the initial position.

Holding together The preference for a complete subtree in the initial position is so strong that it can exclude readings which would presuppose a partial subtree, as in (193b).

(193) a. Lidé na vesnici *se* stěhují neradi.
people on village RFL move unwillingly
'People in the country don't like to move.'
b. Lidé *se* na vesnici stěhují neradi.
people RFL on village move unwillingly
'People don't like to move to/in the country.'

The excluded reading of (193b) is that of (193a) involving 'people in the country'. Note that the second ('in the country') reading of (193b) does not presuppose a partial tree in the initial position: the PP modifies the verb in both readings. Thus, it is synonymous – modulo FSP – with (194), where the initial PP as topic proper depends on the verb and can again be interpreted as an adverbial of location or direction.

(194) Na vesnici *se* lidé stěhují neradi.
on village RFL people move unwillingly
'In/To the country people don't like to move.'

Even though there is a preference for the initial expression to be a complete subtree, other options are possible, as shown below.

7.2.2 A partial subtree in the initial position

Non-finite verbs Acceptability somewhat decreases if the governor of the initial expression occurs with only some of its dependents. The acceptability of 'partially fronted' embedded infinitival clauses seems to depend on their syntactic function and – to a less prominent degree – on lexical properties of the matrix verb. Contrastive reading of the initial expression improves some of the examples.

(195) a. Rozdávat květiny *se* ženám před volbami musí.
b.??Rozdávat ženám *se* květiny před volbami musí.
c. ?Rozdávat *se* ženám květiny před volbami musí.

(196) a. Číst pohádku *se mi* dnes dětem opravdu nechce.
b. ?Číst dětem *se mi* dnes pohádku opravdu nechce.
c. ?Číst *se mi* dnes dětem pohádku opravdu nechce.

(197) a. Pohlídat děti *si* Novákům troufne jen tetička z Plzně.
b.??Pohlídat Novákům *si* děti troufne jen tetička z Plzně.
c. ?Pohlídat *si* děti Novákům troufne jen tetička z Plzně.

Similarly with passive participles:

(198) ?Obdarovány *si* přály být květinami jen některé dámy.

Examples (195), (196) and (198), where the non-finite verb is autosemantic with a synsemantic verbal daughter, are already covered by the Initial Non-Finite Verb Compaction constraint (186).[17]

Examples in (197) are parallel to (184c), repeated here as (199).

(199) Pohlídat děti Novákům *si* troufne jen tetička z Plzně.
baby-sit kids Nováks-DAT RFL ventures only auntie from Pilsen
'Only the auntie from Pilsen is brave enough to baby-sit the Nováks' kids.'

In (199) the complete subtree governed by the infinitive occupies the initial position due to DSOP, similarly as any other dependent of the main verb in the clause. However, in (197) there are only partial subtrees in the initial position. This suggests that there should be an alternative to compacting infinitive (or passive participle) with all its dependents to *inf-fld*, namely that compacting the infinitive and some of its dependents to *pre-cl-fld*, while letting the items in the rest compact individually to *rest-fld* of the finite clause.

(200) INITIAL NON-FINITE VERB COMPACTION II

$$
\left(\begin{array}{l}
\left[\begin{array}{l}
\textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{h}\ \left[\begin{array}{l} \textit{infinitival} \vee \textit{pass-pple} \\ \textsc{f-word}\ \textit{no} \\ \textsc{s-node}\ \boxed{1} \end{array}\right] \\
\textsc{surface}\ \boxed{2}
\end{array}\right] \\
\wedge\ \texttt{append}(\boxed{3}, \boxed{4}, \boxed{2}) \\
\wedge\ \texttt{member}(\boxed{1}, \boxed{3})
\end{array}\right)
\rightarrow
\left(\begin{array}{l}
\texttt{region\_setup}(\boxed{3}, \left[\begin{array}{l} \textit{inf-fld} \\ \textsc{region}\ \textit{pre-cl-fld} \end{array}\right]) \\
\vee\ \texttt{region}(\boxed{2}, \textit{inf-fld})
\end{array}\right)
$$

In every sign with an infinitive or passive participle (see (184d)) as the autosemantic head, with an *s-list* ($\boxed{2}$) that can be non-deterministically partitioned into two sublists – $\boxed{3}$, including the *s-node* of the infinitive, and $\boxed{4}$ – the smallest common region of $\boxed{3}$ is *inf-fld* within *pre-cl-fld*, or the region of the whole sign is *inf-fld*.

Nouns and pronouns In (201a) the interrogative expression *kolik* is syntactically equivalent to a numeral. As such, it can be analysed as the head

[17] The subtle distinctions in the acceptability of some examples may be due to DWO. The constraint (186) should therefore select only such items for *pre-cl-fld* compaction, which would be positioned leftmost according to DSOP.

of the noun *let* and represents another species of the 'partial fronting' type. As in (190) above, the initial item in the answer (201b) requires sentential stress in order to be acceptable in that position.

(201) a. Kolik *je ti* let?
b. HODNĚ už je mi let.

The pair (202a) and (202b) shows a similar contrast. Interestingly, in (202c) the initial item need not be stressed, although the judgement concerning the contrast between (202b) and (202c) is rather fragile.

(202) a. Co *ses* dozvěděl nového?
b. NIC *jsem se* nedozvěděl nového.
c. Nic *jsem se* nového NEDOZVĚDĚL.

Adapting slightly the constraint on partial compaction of non-finite verbs (200), the initial position of the governing noun is made possible by (203).

(203) INITIAL NOUN COMPACTION

$$\left(\begin{array}{l} \begin{bmatrix} \text{S} \mid \text{L} \mid \text{C} \mid \text{H} \begin{bmatrix} \textit{noun} \\ \text{S-NODE } \boxed{1} \end{bmatrix} \\ \text{SURFACE } \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{append}(\boxed{3}, \boxed{4}, \boxed{2}) \\ \wedge\ \texttt{member}(\boxed{1}, \boxed{3}) \end{array}\right)$$

$$\rightarrow \left(\begin{array}{l} \left(\begin{array}{l} \texttt{region\_setup}\left(\boxed{3}, \begin{bmatrix} \textit{noun-fld} \\ \text{REGION } \textit{pre-cl-fld} \end{bmatrix}\right) \\ \wedge\ \texttt{region\_setup}(\boxed{4}, \textit{rest-fld}) \end{array}\right) \\ \vee\ \texttt{region}(\boxed{2}, \textit{noun-fld}) \end{array}\right)$$

In every sign headed by noun, with an *s-list* ($\boxed{2}$) that can be non-deterministically partitioned into $\boxed{3}$, including the *s-node* of the noun, and $\boxed{4}$, either the region of $\boxed{3}$ is *noun-fld* within *pre-cl-fld* and the region of $\boxed{4}$ is *rest-fld* within *pre-cl-fld*, or the region of the whole sign is *noun-fld*.

Assuming that *noun* as a subtype of *head* covers pronouns and a subset of numerals, the constraint allows not only for (201) and (202), but also for all examples in (205) and (206). However, the examples show that some cases of partial fronting may be much less plausible. Cf. the variants of (204) in (205).

(204) Velmi vlhký oceánský vzduch *se* přesouvá přes Německo k východu.

(205) a. *?Vzduch *se* velmi vlhký oceánský přesouvá přes Německo k východu.

b.??Vzduch *se* přesouvá velmi vlhký oceánský přes Německo k východu.
c. ?Vzduch *se* přesouvá přes Německo k východu velmi vlhký oceánský.

It can be argued that in the marginally acceptable examples in (205) the adjectives are interpreted as modifying the verb rather than the fronted noun. However, in (206) this option should not be possible, yet (206d) has a similar acceptability status. What seems to be more critical is the possibility to contrast the partially fronted expression with some other discourse entities, leaving the non-contrasted parts of the nominal group to the right of the clitics:[18]

(206) a. Obrázkovou knížku o Praze *jsem mu* dal už minule.
b.??**Knížku** *jsem mu* dal obrázkovou o Praze.
c. ?**Obrázkovou knížku** *jsem mu* dal o Praze.
d. ?**Knížku o Praze** *jsem mu* dal obrázkovou.

Here I will rely again on DSOP, which can provide at least a partial solution to the subtle differences in acceptability.

The subtree(s) or its part(s) in the initial position must be continuous: consider (207) and (208) as unacceptable variants of (204), where the string *velmi vlhký* corresponding to a subtree embedded within the initial subtree is split.

(207) a. *Vlhký vzduch *se* velmi přesouvá přes Německo k východu.
b. *Vlhký vzduch *se* přesouvá velmi přes Německo k východu.
c. *Vlhký vzduch *se* přesouvá přes Německo k východu velmi.

(208) a. *Velmi vzduch *se* vlhký přesouvá přes Německo k východu.
b. *Velmi vzduch *se* přesouvá vlhký přes Německo k východu.
c. *Velmi vzduch *se* přesouvá přes Německo k východu vlhký.

Since the modifiers of the fronted noun compact within their local trees, examples in (207) and (208) are correctly excluded.

7.2.3 Multiple subtrees in the initial position

The initial expression can also consist of several adverbials, local or temporal, together providing a 'setting' or 'stage' for the described event or situation (209a), or indicating a path or duration (209b). Each of the multiple subtrees is a major sentential constituent (i.e., is governed by a dependent of the main verb).

[18] The boldface type denotes contrastive stress.

(209) a. Loni v létě na Bahamách *se* *mi* zdálo všecko snadné.
Last summer on Bahamas RFL to me seemed all easy
'Last summer on the Bahamas everything seemed easy to me.'

b. Od Plzně přes Budějovice až do Brna se potutelně usmívala.

Koktová (1999) and Avgustinova and Oliva (1995) provide examples where the initial expression can consist of multiple items if they constitute a contrastive topic. Avgustinova and Oliva (1995) qualify this option with a syntactic restriction: the items may consist only of free adjuncts or of circonstant (free modification) complements.

The explanation of Avgustinova and Oliva (1995) why multiple subtrees can occur in the initial position rests on the notion of *communicative segments*, defined as a contiguous sequence of adjacent syntactic units with the same degree of communicative dynamism.[19] The initial expressions in (209) would constitute a single communicative segment and so would the partial subtrees exemplified in the preceding sections. A single complete subtree represents a trivial example of a communicative segment. The position of clausal clitics can then be specified as following the first substantial communicative segment.[20] Avgustinova and Oliva (1995) conclude that by using this generalization the placement of clausal clitics obeys the same (i.e., communicative) constraints as the constraints on word order in Czech in general.

The latter statement has to be taken with a grain of salt: there are many word order phenomena in Czech – including those involving clitics – for which syntax, phonetics or prosody rather than communicative factors are responsible. Adopting the notion of communicative segment as the sole basis for specifying the position of clausal clitics seems to shift the burden of constraining the initial expression to another level and leads to a paradox: any syntactic and prosodical constraints on what constitutes the initial expression have to be applied to a discourse-based entity.

Therefore, an adequate solution should honour the primary role of discourse in selecting which items occur in the initial position, but – at the same time – the discourse-based factors should not override factors originating elsewhere.

Although the notion of communicative segment is probably the best available explanation for multiple subtrees in the initial position and one that might shed light on other puzzling phenomena related to surface order, there is one obstacle to its adoption here. In order to formalize the notion and map tectogrammatical communicative segments onto surface word order. the present way of tectogrammatical representation of expressions would have to be substantially modified. Most probably, the representation

[19]Thus, two or more semantemes on TR may have the same degree of CD. A cluster of clausal clitics also constitute a communicative segment, but the clitics are "informationally unessential" and do not participate in the CD ordering.

[20]A substantial communicative segment can be assigned a degree of CD.

of dependency relations would have to be decoupled from the representation of deep word order. The latter would then be ready to include a structure expressing communicative segmentation.

I will leave this potential move for future research and content myself with the possibility of underspecifying surface order: the initial field could be allowed to hold more than one dependent subtree by redefining its occupancy property.

7.2.4 A non-tree in the initial position

Here, the initial expression does not include a governor with some of its dependents (as in the 'partial subtree' type), nor does it consist of several subtrees whose governors depend on the main verb. The initial expression consists of two or more sister dependents of a (typically nominal) dependent of the main verb, the whole subtree being split by the clitic cluster and the rest of the clause. In many respects, this type parallels the interrogative or contrastive subtype of 'complete subtree' and 'partial subtree' type, where the initial expression is a dependent of a noun.

Examples in (210) have already been discussed in §6.5.5. As a solution, the PP Compaction constraint was suggested (152).

(210) a. O jakou *se* jedná soutěž?
b. O jak dotovanou *se* jedná soutěž?
c. *O jak *se* jedná dotovanou soutěž?

The examples in (211) show that this pattern seems to restricted to preposition with an interrogative expression in the initial position, yet it can be made more plausible with a special TFA, as in (211c), where *vzduch* has a contrastive interpretation.

(211) a. *Velmi vlhký oceánský *se* vzduch přesouvá přes Německo k východu.
b. *Velmi vlhký oceánský *se* přesouvá vzduch přes Německo k východu.
c. ?Velmi vlhký oceánský *se* přesouvá přes Německo k východu **vzduch**.

Similarly in (212), where *bohatou* is contrasted.

(212) No přece tu **bohatou** *si* vybral nevěstu.

I will ignore the only acceptable example in (211) and use (212) merely to sketch a solution resting on two points: (i) *matrix-fld* must be extended to accommodate items such as discourse connectives *no přece*; and (ii) the demonstrative pronoun *tu* should be treated as a proclitic, compacting with the following adjective into an item, which can be assigned the initial field.

7.2.5 Summary

Multiple factors seem to determine what kinds of expressions can occupy the initial position: syntactic, prosodical and communicative (discourse-based).

In (181)–(190) the initial expression consists of a single continuous subtree (a syntactic constituent), in (195)–(202) of a part of a single continuous subtree including its governor, in (209) of several continuous subtrees, in (210) of several parts of a subtree without its governor. Note that it is not necessary for the subtree to be governed by the main verb (182b), (189a)–(190). Still, there are syntactic restrictions on the initial expression, as shown by the unacceptable examples (207) and (208) involving discontinuity within more deeply embedded structures. Also, the initial tree may not be headed by a past participle (185).

In (210) the preposition or the determiner must form a prosodical unit with the first item of the nominal group. This suggests that prosodical factors may override the preference for a single complete tree in the initial position.

Finally, there is more than enough evidence about the importance of communicative factors. First of all, the choice of items for the initial position is determined by their communicative dynamism: it is either topic proper (the usual case), contrastive topic (as in (190) and (192c)), or – under sentential stress – focus proper ((201b) and (202b)). Again, the cases with partial subtrees in the initial position suggest that communicative factors can override the syntax-based preference for complete subtrees. Similar or equal communicative properties of multiple subtrees seem to be responsible for the possibility of their initial presence in (209) and (209a).

The emerging picture is that of syntactic preference for a single initial subtree, selected by communicative criteria, which can be overridden for communicative reasons by splitting the subtree or by splicing two or more subtrees, and for prosodical reasons by splicing a prosodically dependent item with its host, even though they do not form a subtree of any sort.

7.3 Clitic climbing and haplology

Czech clitics often follow a pattern common in some Romance and some other Slavic languages and allow (or even strongly prefer) the placement of one or more clitics hosted by an embedded infinitive in the second position of the sentence, as in (213a), where the accusative pronoun is a dependent of the embedded infinitive, or in (213b), where the reflexive belongs to the infinitive complement of the predicative adjective.[21]

[21]Note that – unlike in English – the infinitive in (i) is treated in accordance with the Czech syntactic tradition as the subject of the sentence and thus no clitic climbing occurs.

(213) a. Pavel *nám* *ho* pomohl najít.
Paul we-DAT. he/it-ACC. helped to find
'Paul helped us to find him/it.'

b. Nakonec *si* byl ochoten to připustit i Tomáš.
In the end RFL was willing it to admit even Tom
'In the end, even Tom was willing to admit it.'

At least in some cases, if there is a chain of successively embedded non-finite verbs, the clitic(s) may also climb into a domain of an intermediate member of the chain, as in (223) below.

This property of Czech clitics should not be suprising given the general availability of scrambling dependents of an embedded infinitive with the dependents of a higher verb:

(214) Pavel *nám* tu ztracenou knihu nakonec po přemlouvání
Paul we-DAT. that lost book in the end after persuasion
pomohl najít.
helped to find
After some persuasion, Paul eventually helped us to find the lost book.

It is not necessary for the climbing clitic to be dependent of an infinitive. Genitive and other pronominal clitics, including reflexives, can climb from domains governed at least by adjectives, adverbs and numerals (215) and (242).

(215) Marie *si* musí začít být vědoma svých předností.

7.3.1 General Constraints

There are some robust and some looser restrictions to clitic climbing, applying indiscriminately to all forms of clitics.

Robust constraints The robust constraints include the following:

- Climbing is not possible from finite clauses (216) and subtrees headed by gerunds (217), adjectival participles (218), and adverbial participles (transgressives) (219).[22]

  (216) a. Šéf *ho* nařídil zbavit všech výsad.

(i) Nakonec *si* bylo snadné odpustit.
In the end RFL was easy to forgive
'In the end, it was easy to forgive each other.'

[22] I will use the term *deverbatives* for the latter three categories. These categories present a barrier to clitic climbing in general: a clitic cannot climb through their domains.

b. Šéf nařídil, aby *ho* zbavili všech výsad.
c. *Šéf *ho* nařídil, aby zbavili všech výsad.

(217) a. Dědeček nemá rád dětské ušklíbání *se* nad polévkou.
b. *Dědeček *se* nemá rád dětské ušklíbání nad polévkou.

(218) a. Uvítali bychom více takových kajících *se* hříšníků.
b. *Uvítali bychom *se* více takových kajících hříšníků.

(219) a. Ředitel vzhlédl od dopisu, tváře *se* ustaraně.
b. *Ředitel *se* vzhlédl od dopisu, tváře ustaraně.

A consequence of this constraint is that only personal pronouns and reflexives are allowed to climb: the conjunctions *-li* and *však* and all forms of the auxiliary *být* or the copula *být* can occur only in finite clauses, from which climbing is impossible. The other two clitics – adverbs *už* and *prý* – are very rare in non-finite clauses, and if they occur in a matrix clause, a reading involving climbing from an embedded infinitive seems impossible. Thus, I will assume that no specific lexical constraint on clitic climbing is necessary.

A question arises whether the constraints on climbing from deverbatives can be extended also to non-verbal categories (nouns, adjectives, adverbs). The answer is no: it is fairly common for genitive pronominal clitics governed by such categories to climb. However, it seems that a slightly looser constraint can hold: a climbing clitic may not be governed by a finite verb or a deverbative and may not climb through a domain governed by a finite verb, a deverbative, or a non-verbal category. Thus, the only category through whose domain a clitic may climb is infinitive.

- Clitics dependent on a 'lower' verb (or other category) cannot climb over clitics dependent on a 'higher' verb (or other category).

(220) a. Pavel *se* snažil *mu* pomoci *ho* najít.
b. Pavel *se mu* snažil *ho* pomoci najít.
c. Pavel *se mu ho* snažil pomoci najít.
d.*?Pavel *se ho* snažil *mu* pomoci najít.

- Two phonologically identical clitics cannot co-occur in a single clitic cluster as a result of clitic climbing:

(221) a. Kamila mi slíbila to vrátit MNĚ.
b. *Kamila mi mi to slíbila vrátit.
c. Kamila mi to slíbila vrátit.
d. Kamila mi slíbila mi to vrátit.

This phenomenon has been noted, e.g., in Avgustinova and Oliva (1995) and applies to all clitics.[23] Example (221c) is a case of haplology, which occurs when two phonologically identical clitics (reflexives and personal pronouns) would otherwise meet in a single cluster. See (232) and further for a discussion of haplology of reflexives.

Note that two phonologically (and morphologically) identical pronouns can co-occur in a single cluster, if both originate within the same clause, as in (222):

(222) a. (A kdo pomůže Heleně s dračí smyčkou?)
b. Lukáš *ji ji* už naučil.

Preferences At the opposite end, there are constraints whose better name is preferences. Karlík, Nekula, and Rusínová (1995, p. 651) claim that climbing occurs if the main verb is modal. In (223) the chain begins by a future auxiliary and a modal, and the 'climbing-to-top' version (223a) is certainly more acceptable than the other versions where the clitic has not climbed, or has not climbed up all the way to the top.[24]

(223) a. Karel *si* bude ještě tento rok chtít koupit nový počítač. [KAR:651]
b. Karel bude ještě tento rok chtít koupit *si* nový počítač.
c. Karel bude ještě tento rok chtít *si* koupit nový počítač.
d. Karel bude ještě tento rok *si* chtít koupit nový počítač.

On the other hand, if the main verb is not modal, Karlík, Nekula, and Rusínová (1995, p. 651) observe that another factor is involved, namely the number of overt dependents of the embedded infinitive: the clitic is 'attracted' by the other dependents. In (224) the verb *pomoci* has only one overt dependent and the 'climbing-to-top' version (224a) is preferred.

(224) a. Měl jsem ho rád, a tak *jsem se mu* rozhodl pomoci.
b. Měl jsem ho rád, a tak *jsem se* rozhodl *mu* pomoci.
c. Měl jsem ho rád, a tak *jsem se* rozhodl pomoci *mu*.

If, however, the non-finite verb has more overt dependents, the non-climbing version (225a) sounds better:

(225) a. Měl jsem ho rád, a tak *jsem se* rozhodl pomoci *mu* se stavbou chaty.

[23] Although for different reasons: auxiliaries cannot be repeated since two finite verbs are not possible in a single clause.

[24] The 'non-top' versions sound better with a pause before the embedded domain with the clitic.

b. Měl jsem ho rád, a tak *jsem se mu* rozhodl pomoci se stavbou chaty.

Summary The strict constraints to clitic climbing presented so far are of a syntactic nature and – unlike the 'constraints qua preferences' just mentioned – are suitable candidates for a smooth integration into the formal system: (i) climbing is impossible from domains governed by finite clauses, gerunds, adverbial and adjectival participles (216)–(219), (ii) a clitic can climb on the way to the sentential 2nd position over domains headed only by infinitives (iii) clitics dependent on a 'lower' verb cannot climb over clitics dependent on a 'higher' verb (220), and (iv) if two phonologically identical clitics would meet in a single cluster as a result of clitic climbing, haplology occurs.[25]

The extended version of Partial Compaction constraint (226) 'liberates' a *pre-cl-fld* or a *cl-fld*.

(226) Partial Compaction, version 3

$$
\left(\begin{array}{l}
[\mathrm{SURFACE}\ \boxed{1}] \\
\wedge\ \mathtt{shuffle}(\boxed{2}, \boxed{3}, \boxed{1}) \\
\wedge \left(\begin{array}{l}
\mathtt{region}(\boxed{2}, \textit{pre-cl-fld}) \\
\vee \left(\begin{array}{l}
\mathtt{region}(\boxed{2}, [\mathrm{REGION}\ \textit{cl-fld}]) \\
\wedge \neg\, [\,\mathrm{S}\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{A\text{-}MORPH}\ \textit{finite}] \\
\wedge \neg\, [\,\mathrm{S}\,|\,\mathrm{L}\,|\,\mathrm{C}\,|\,\mathrm{HEAD}\ (\textit{v-noun} \vee \textit{adjp} \vee \textit{advp})\,]
\end{array}\right)
\end{array}\right)
\end{array}\right)
\rightarrow \exists_{\boxed{4}} \left(\begin{array}{l}
\mathtt{region\_setup}(\boxed{3}, \boxed{4}) \\
\vee\ \mathtt{region}(\boxed{1}, \boxed{4})
\end{array}\right)
$$

There are two additional conditions to be satisfied for the constraint to apply to a *cl-fld*: the complex morphological form of the governor cannot be finite and its head value cannot be a verbal noun, an adjectival or adverbial participle.[26]

In order to prevent a clitic from climbing through other regions that those governed by an infinitive, the condition concerning deverbatives introduced in (226) should be extended by another condition requiring that the clitic is a daughter in the sign. If this is not the case, the governor must be an infinitive. The version of the Partial Compaction constraint in (227) incorporates this modification.[27]

[25] However, see (233) for a few examples where haplology occurs even if the two clitics are not phonologically different.

[26] I am assuming the hierarchy of head values as in Przepiórkowski (1999a, p. 418).

[27] The attribute R abbreviates REGION and the string S|L|C|H|S-N|R stands for SYNSEM|LOCAL|CATEGORY|HEAD|S-NODE|REGION.

(227) Partial Compaction, version 4

$$
\begin{array}{l}
\left(\begin{array}{l}
[\textsc{surface}\ \boxed{1}] \\
\wedge\ \mathtt{shuffle}(\boxed{2}, \boxed{3}, \boxed{1}) \\
\wedge \left(\begin{array}{l}
\mathtt{region}(\boxed{2}, \textit{pre-cl-fld}) \\
\vee \left(\begin{array}{l}
\mathtt{region}(\boxed{2}, \boxed{5}[\textsc{r}\ \textit{cl-fld}]) \\
\wedge\ \neg\ [\textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{a-morph}\ \textit{finite}] \\
\wedge \left(\begin{array}{l}
[\textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{head}\ \textit{infinitival}] \\
\vee \left(\begin{array}{l}
[\textsc{nonhead-daughters}\ \boxed{6}] \\
\wedge\ \mathtt{member}([\textsc{surface}\ \boxed{5}], \boxed{6}) \\
\wedge\ \neg\ [\ \textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{head}\ (\textit{v-noun} \vee \textit{adjp} \vee \textit{advp})\ ]
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right) \\
\rightarrow \exists_{\boxed{4}} \left(\begin{array}{l}
\mathtt{region\_setup}(\boxed{3}, \boxed{4}) \\
\vee\ \mathtt{region}(\boxed{1}, \boxed{4})
\end{array}\right)
\end{array}
$$

The latter version is still rather naive in its assumption that it is always only items compacting to a single region *pre-cl-fld* or a single item [REGION *cl-fld*], which are 'liberated'. The more realistic version of Partial Compaction is shown below:

(228) Partial Compaction, version 5

$$
\begin{array}{l}
\left(\begin{array}{l}
[\textsc{surface}\ \boxed{1}] \\
\wedge\ \mathtt{shuffle}(\boxed{2}, \boxed{3}, \boxed{1}) \\
\wedge \left(\begin{array}{l}
\mathtt{member}(\boxed{7}, \boxed{2}) \\
\rightarrow \left(\begin{array}{l}
\mathtt{component}(\textit{pre-cl-fld}, \boxed{7}) \\
\vee \left(\begin{array}{l}
\boxed{7}\Big[\textsc{f}\ \boxed{5}[\textsc{r}\ \textit{cl-fld}]\Big] \\
\wedge\ \neg\ [\textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{a-morph}\ \textit{finite}] \\
\wedge \left(\begin{array}{l}
[\textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{head}\ \textit{infinitival}] \\
\vee \left(\begin{array}{l}
[\textsc{nonhead-daughters}\ \boxed{6}] \\
\wedge\ \mathtt{member}([\textsc{surface}\ \boxed{5}], \boxed{6}) \\
\wedge\ \neg\ [\ \textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{head}\ (\textit{v-noun} \vee \textit{adjp} \vee \textit{advp})\ ]
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right) \\
\rightarrow \exists_{\boxed{4}} \left(\begin{array}{l}
\mathtt{region\_setup}(\boxed{3}, \boxed{4}) \\
\vee\ \mathtt{region}(\boxed{1}, \boxed{4})
\end{array}\right)
\end{array}
$$

The constraint applies to every sign satisfying a complex condition: The sign's *s-list* can be construed as a shuffle of $\boxed{2}$ and $\boxed{3}$, where each member of $\boxed{2}$ is assigned either *pre-cl-fld* or *cl-fld*. If the latter is the case the governor must not be a finite verb. Furthermore, the governor must be either infinitive or the clitic is a daughter in the sign and the governor is not a deverbative. Only then there is a region $\boxed{4}$ to which the rest of the sign's *s-list* compacts, or there is a region $\boxed{4}$ to which the whole *s-list* compacts.

Haplology of clitics can only be treated by changing the way two *s-list*s, each including one of the two clitics, are combined. It means that the Surface List Composition Principle (105) must be modified:

(229) SURFACE LIST COMPOSITION PRINCIPLE, version 2

$$\begin{bmatrix} \textit{non-lexical} \\ \textsc{surface}\ \boxed{1} \\ \textsc{head-daughter} \mid \textsc{surface}\ \boxed{2} \\ \textsc{nonhead-daughters}\ \boxed{3} \end{bmatrix} \rightarrow \begin{pmatrix} \texttt{collect\_slists}(\boxed{3}, \boxed{4}) \\ \wedge\ \texttt{append}(\boxed{2}, \boxed{4}, \boxed{5}) \\ \wedge\ \texttt{haplo\_clitics}(\boxed{5}, \boxed{6}) \\ \wedge\ \texttt{multi\_shuffle}(\boxed{6}, \boxed{1}) \end{pmatrix}$$

The relation `haplo_clitics/2` checks whether in the combined *s-lists* ($\boxed{5}$) there is an uncompacted clitic, which is assigned the same field as the other clitic. If this is the case, one of the two clitics (more precisely, its *s-node*) is left out from the *s-list* and the rest of the first argument is identified with the second argument ($\boxed{6}$). If this is not the case, the first argument as a whole is identified with the second argument.

The last outstanding strict constraint on clitic climbing from the four listed on p. 229 is that concerning the impossibility of clitics originating in a more embedded syntactic structure climbing higher than other less embedded clitics. Unfortunately, this problem is difficult to solve if climbing is handled as an exclusively ordering problem, because there is no indication of the clitic's origin within a higher *s-list*, not even an indication of its origin relative to other clitics. This issue will recur on two more occasions further below.

7.3.2 Reflexives

The following examples (230) do not involve clitic climbing (the brackets denote clause boundaries). All of them are 100% acceptable if a prosodical break occurs between the two reflexives: the two adjacent clitics do not form a single cluster.

(230) a. [Přál *jsem si*] [*se* vzchopit]. [SYN]
b. [Snažím *se*] [*si* to představit]. [SYN]
c. [Snažil *se*] [*se* tam dostat]. [AO95(22a)]

These examples are covered by clitic fields being optionally present on multiple levels of embedding.

The following examples from Avgustinova and Oliva (1995) show that clitic climbing of two reflexives into a single cluster is not possible, even if they are phonologically (and morphologically) different:

(231) a. *[Stále *si se* snažím] [získat její přízeň]. [AO95(21c)]

b. *[Stále *se si* snažím] [získat její přízeň]. [AO95(21d)]

Two (possibly even more) reflexive pronouns or particles in a single cluster are subject to haplology – (232a) and (232b).[28]

(232) a. Děvče *se* stydělo převléknout.
b. Netroufla *si* říct o více knedlíků.

Interestingly, the two reflexives need not be phonologically and morphologically identical. In (233a) the matrix verb requires the reflexive particle *se* and the embedded verb the reflexive particle *si*, yet haplology is allowed with *si* as the single resulting form. The pair (233b) and (233c), where – again – the matrix verb is *snažit se* while the embedded verb is *představit si*, shows that the form resulting from the haplology of *se* and *si* cannot be *se*. The same contrast can be seen in (233d) and (233e). Examples (233f) and (233g) show that a matrix *si* cannot haplologize with an embedded *se*.

(233) a. Jan *si* bál vzít kravatu. [KO]
b. Snažím *si* to představit.
c. *Snažím *se* to představit.
d. Styděla *si* sednout do první řady.
e. *Styděla *se* sednout do první řady.
f. *Troufla *si* usadit v první řadě.
g. *Troufla *se* usadit v první řadě.

As (234) shows, *si* can climb above the region where *se* originates.

(234) ?Marie si nemusela stydět sednout do první řady.

I conclude that two reflexive clitics indeed cannot co-occur in one clitic cluster (231), and in addition to the general assumption on haplology (point (iii) on p. 229). The reflexive *si* should also be accepted as the result of haplologizing a lower *si* with a higher *se*.

The former observation is reflected by the setup of *cl-fld*, where the field *cl-rfl-fld* accommodates at most one item. I have no suggestion concerning the latter phenomenon given the absence of information on the clitic's origin. This issue was already mentioned above.

7.3.3 Datives

Examples (235a) and (235b) are parallels of (230a) and (230b), where the two dative pronouns form two separate clusters, with a preferred prosodical break

[28] It does not matter whether the two items are both of the same category or not, what seems to matter is their phonological identity. In both (232a) and (232b) it is a particle which goes with the matrix verb and a pronoun which goes with the infinitive. However, see below for examples where the two items need not be even phonologically identical.

in between. However, unlike reflexive pronouns, the two dative pronouns can form a single cluster, as shown by (235c) and (235d).

(235) a. [Podařilo se *mi*] [*mu* amputovat pravou zadní nohu]. [BOBO]
b. [Podařilo se *mu*] [*mi* udělat velkou radost].
c. Včera se *ti jí* to konečně povedlo vysvětlit.
d. Včera se *jí ti* to konečně povedlo vysvětlit.

Although the order of the two dative clitics in (235c) and (235c) seems to suggest that the linear order of clitics reflects that of their governors, (236) shows that it is not the case.

(236) a. Poslat kurýrem *se mi mu ho* dnes nepodařilo. [AO(20)]
b. Poslat kurýrem *se mu mi ho* dnes nepodařilo.

In (236a) *mi* is a complement of *nepodařilo se* and *mu* is a complement of *poslat*, which means that the linear orders of the clitics and their governors are mutually reversed. Example (236b), where *mu* is a complement of *nepodařilo se* and *mi* is a complement of *poslat*, presents only additional evidence that it is the level of embedding of the governor, rather than its linear position, which predicts the position of the clitic.

What is significant about the above examples is that only some verbs seem to allow this kind of clitic climbing:

(237) a. Poslat *se mi mu ho* nepovedlo.
b. Poslat *se mu mi ho* nepovedlo.

(238) a. Poslat *se mu ho* neuráčila.
b. Poslat *se mi ho* neuráčila.

(239) a.??Poslat *mi mu ho* neslíbil.
b.??Poslat *mu mi ho* neslíbil.

I will assume that if two datives meet in a clitic cluster due to climbing, their order corresponds to the level of embedding of their governors. This is the third and final case where clitic climbing defies a solution based only on the information encoded within *s-list*s. Also, I will refrain from an attempt to identify the class of verbs which allow the climbing of dative pronominal clitics.

7.3.4 Accusatives

Similarly as with dative clitics, when two accusative clitics meet in a single cluster due to clitics climbing, their order corresponds to the level of embedding of their respective governors, with the complement of the matrix verb coming first (240) and (241).

(240) a. Třeba *tě ji* nechají zahrát.

b.??Třeba *ji tě* nechají zahrát.
c. Třeba *tě ji* zahrát už nenechají.
d.??Třeba *ji tě* zahrát už nenechají.

(241) a. Někdy *vás ji* naučím hrát. [SYN]
b. Dodnes *nás ji* baví hrát. [SYN]
c. Nebylo návštěvy, aby *mne je* před ní nenutili přeříkávat. [SYN]

Thus, I will extend the conclusion regarding pronominal clitics in the dative to those in the accusative.

7.3.5 Genitive: climbing from non-clausal dependents

Most examples involving the climbing of a genitive personal pronoun are interesting in that the pronoun's governor is not a verb but rather an adverb (242a), a numeral (242b) (at least according to some syntactic analyses), adverbs/numerals (242c) and (242d), and an adjective (242e).

(242) a. Je *mi tě* líto. [KAR:649]
b. Napadlo *mne jich* nejmíň 18. [SYN]
c. Netoulá *se jich* tu zrovna moc. [SYN]
d. Je *nás* tu čím dál víc.
e. Už *je vás* ale skoro plný dům.

Note that in (242e) the clitic has climbed to the clausal clitic position from an adjective governed by a noun. However, a clitic (*jim*) governed by a pronoun (*všem*) does not seem to be able to cross the barrier of a non-finite clause (243).

(243) a. Viděl *ho* pomáhat *jim* všem.
b. *Viděl *jim ho* pomáhat všem.

A more deeply embedded genitive clitic can form a single cluster with another clitic:

(244) Král mu jich přikázal sníst dvě plné mísy.

I will assume that genitive clitics are not subject to any other restrictions beyond those applying to personal pronouns.

7.3.6 Summary

1. A clitic can climb to a higher 2P unless it is governed by a finite verb (216), a deverbative (gerund) (217), an adjectival participle (218), or an adverbial participle (219).

2. A clitic may only climb through a domain governed by an infinitive.

3. A more deeply embedded clitic cannot climb over a less deeply embedded clitic (220).

4. Two phonologically identical clitics with different governors either do not co-occur in a single clitic cluster or haplologize (221), (231), (232).

5. Two reflexives *si* and *se* can haplologize yielding *si*, if the reflexive *si* originates in a more embedded domain (233).

6. The order of pronominal clitics with the same morphological case co-occurring in a single cluster due to clitic climbing corresponds to the level of embedding of their governors.

Only the first three points have been reflected in the present framework. The last three points may serve as evidence that clitic climbing and haplology are phenomena which require a syntactic rather than *s-list*-based solution. I will leave the issue open for further investigation.

7.4 The order of second position clitics

What follows is a numbered list of positions – 'fields' within a single clitic cluster. Some types of clitics exhibit remarkably rigid ordering properties: such is the case of constant clitics, including most forms of personal pronouns. Such types of clitics can be assigned a unique position. Other types of clitics, either because of inconclusive data or because of demonstrated possibility to occur at multiple or arbitrary positions within a clitic cluster, can be assigned a disjunction of fields or, equivalently, a region generalizing over such a disjunction, up to the region of a clitic cluster.

None of the fields are obligatory. However, it is important to realize that very often two clusters are placed adjacently, and the order of clitics is respected only within each individual cluster, cf. (245).

(245) a. [Nenapadlo *ho*] [*mu to* poslat].
b. Před léty nebylo pokládání věnců u tohoto pomníku tak komorní záležitostí jako dnes a [účastnit *se ho*] [*si* pokládala za čest řada státních, hospodářských a zejména stranických orgánů teplického okresu].

7.4.1 Clitics with unique positions

1. *cl-lis-fld*

   - conjunction *-li* 'whether'
   - *s* – contracted form of the past tense auxiliary *být* or the copula *být*, except *ses* and *sis*

No clitic climbing from embedded clauses is allowed for the two forms. Individual speakers may have preferences regarding one of the other order, but the preferences differ and also seem to be influenced by phonological factors:

(246) a. Řekla-*lis mu to*, tím lépe.
b. Řekla*s-li mu to*, tím lépe.
c. ?Málo-*lis* pronásledován výčitkami?
d.??Málo*s-li* pronásledován výčitkami?

The co-occurence of *-li* and *s* is a marginal phenomen anyway, since *-li* tends to be avoided after the *-l...* endings characteristic for the past participle, while with the *s* contraction of past tense auxiliary past participle is required. However, it is necessary to assume that this position can be filled by 0–2 items.

2. *cl-be-fld*
   - forms of *být* as the past tense auxiliary: *jsem, jsi, je, jsme, jste, jsou*
   - forms of *být* as the present conditional auxiliary: *bych, bys, by, bychom, byste*
   - non-negated forms of *být* as the periphrastic passive auxiliary and as the copula: *jsem, jsi, je, jsme, jste, jsou*

   In a single clause, all the forms are mutually exclusive and none of them is subject to clitic climbing. Therefore, this position can be filled by 0–1 item.

3. *cl-rfl-fld*
   - *se* and *si* – reflexive particles or reflexive pronouns
   - *ses* and *sis* – contractions of the past tense auxiliary *být* or the copula *být* with reflexive particles or reflexive pronouns

   The two forms *se* and *si* are mutually exclusive in a single cluster (247a).

(247) a. *Představil *jsem si se*, jak stojím na jeho místě.
b. Představil *jsem si* sebe, jak stojím na jeho místě.

Also the forms *ses* and *sis* are mutually exclusive and so is the pair *se* and *si* vs. *ses* and *sis*.

This position can be filled by 0–1 item.

4. *cl-ethdat-fld*

   - dative pronouns *ti* and *vám* as 'ethical dative', perhaps marginally also *jim* in the (obsolete) polite forms using third person plural

   Clitic climbing of ethical datives is not possible.[29]

   (248) Včera *ti mi* povídal, že se žení. [KAR:649]

   Ethical dative can follow or precede the reflexive:

   (249) a. Včera *se ti* najednou tak rozpršelo.
   b. Včera *ti se* najednou tak rozpršelo.

   The forms *ses* and *sis* do not co-occur with ethical dative, but this is a constraint stemming from the unacceptability of ethical dative in a clause with a 2nd person subject:

   (250) a. *Včera *ses ti* pěkně zřídil!
   b.??Včera *ti jsi* byl pod obraz boží.

   This position can be filled by 0–1 item.

5. *cl-freedat-fld*

   - personal pronoun as 'free' dative – different from ethical dative, which is only used in 2nd person:

   (251) Ta kočka *ti jim* spí v posteli.

   This position can be filled by 0–*n* items. Clitic climbing of free dative may be possible, however more than one free dative in a single clitic cluster is probably hard to interpret.

6. *cl-dat-fld*

   - dative pronouns *mi, ti, jí, mu, nám, vám, jim*

   See (170a)–(170d) above.

[29]Ethical dative may sound acceptable even at other positions:

(i) ?Včera *ti jsem* potkala Máňu!

If the placement of ethical dative is less restricted, its position should be specified by a disjunction of fields or by a region consisting of more fields.

Since a dative clitic can climb from an embedded non-finite clause, this field can be filled by 0–n items.[30]

7. *cl-acc-fld*

   - accusative pronouns *mne, mě, tě, ji, jej, ho, je, nás, vás*

   There are a few Czech verbs with two accusative complements, such as *naučit*. The two pairs in (252) show that the order of two accusative clitics matters even if both have the same governor.

   (252) (A co s tím uzlem?)
   a. Naučím *tě ho* zítra.
   b. *Naučím *ho tě* zítra.
   c. Zítra *tě ho* naučím.
   d. *Zítra *ho tě* naučím.

   It seems that their order respects systemic ordering, contrary to the standard assumption in FGD about the order of items in the topic, which are supposed to be governed by discourse-related principles.

   This is a position which can be filled by 0–n items.

8. *cl-gen-fld*

   - genitive pronouns *mne, mně, tě, jí, ho, nás, vás, jich*

   See (242) on p. 234. This is a position which can be filled by 0–n items.

9. *cl-ins-fld*

   - instrumental pronouns *mnou, tebou, jí, jím, námi, jimi, sebou*

   (253) Pak *jsem se jím už* nezabýval.

   This is a position which can be filled by 0–n items.

7.4.2 Clitics with non-unique positions

These types may not be homogeneous classes and also the types in this group may differ in ways which have not yet been made clear enough. Since they can occur anywhere in the clitic cluster, they will be assigned a position with its precise order unspecified.

- *cl-nom-fld*

  nominative pronouns *já, ty, ona, on, ono, my, vy, ony, oni*

[30] The order of multiple datives resulting from clitic climbing corresponds to the level of embedding of their respective governors (236). Similarly for the other cases.

(254) a. Tak tohle *jsem mu já* neříkal. [KAR:649]
b. Tak tohle *jsem já mu* neříkal.
c. Tak tohle *já jsem mu* neříkal.

- *cl-uz-fld*, *cl-pry-fld*, *cl-vsak-fld*

  the "short" adverbs *už* 'already', *prý* 'allegedly'; *však* as a conjunction 'however'

7.4.3 Summary

Positions of clitics within a cluster:

- Unordered:

  cl-nom-fld, *cl-uz-fld*, *cl-pry-fld*, *cl-vsak-fld*: nominative pronouns, *už*, *prý*, *však*

- Ordered:

  1. *cl-lis-fld*: *-li*, *s*; 2
  2. *cl-be-fld*; 1
  3. *cl-rfl-fld*: *se, ses*, *si, sis*; 1
     cl-ethdat-fld; 1
  4. *cl-freedat-fld*; *n*
  5. *cl-dat-fld*; *n*
  6. *cl-acc-fld*; *n*
  7. *cl-gen-fld*; *n*
  8. *cl-ins-fld*; *n*

Table 7.3 summarizes the ordering and compaction facts in the established form.

Finally, it is necessary to make sure that the setup of *cl-fld* will be respected in every *s-list* where it occurs. Unlike most other regions, *cl-fld* is never the common region for all items in an *s-list*, so the general constraints cannot apply to *cl-fld*. This is remedied by the following constraint on clitic compaction:

Region	Field	Order	Occupancy
cl-fld	*cl-lis-fld*	1	≤2
	cl-be-fld	2	≤1
	cl-rfl-fld	3∨4	≤1
	cl-ethdat-fld	3∨4	≤1
	cl-freedat-fld	5	any
	cl-dat-fld	6	any
	cl-acc-fld	7	any
	cl-gen-fld	8	any
	cl-ins-fld	9	any
	cl-nom-fld	any	≤1
	cl-uz-fld	any	≤1
	cl-pry-fld	any	≤1
	cl-vsak-fld	any	≤1

Table 7.3: An overview of topological fields for clitics

(168) Clitic Compaction

$$\forall \boxed{5} \left(\begin{array}{l} \left(\begin{array}{l} [\text{SURFACE}\ \boxed{0} \oplus \boxed{1}] \\ \wedge\ \mathtt{region}(\boxed{1}, \boxed{4}) \\ \wedge\ \mathtt{shuffle}(\boxed{2}, \boxed{3}, \boxed{1}) \\ \wedge \left(\begin{array}{l} \mathtt{member}(\boxed{5}, \boxed{2}) \\ \rightarrow \boxed{5} \left[\text{FIELD} \mid \text{REGION} \begin{bmatrix} \textit{cl-fld} \\ \text{REGION}\ \boxed{4} \end{bmatrix} \right] \end{array} \right) \end{array} \right) \\ \rightarrow \exists \boxed{6} \left(\begin{array}{l} \boxed{5}[\text{FIELD} \mid \text{REGION}\ \boxed{6}] \\ \wedge\ \mathtt{region\_setup}(\boxed{2}, \boxed{6}) \end{array} \right) \end{array} \right)$$

In every sign whose *s-list*'s suffix ($\boxed{1}$) with the smallest common region $\boxed{4}$ can be construed as a shuffle of two *s-list*s, $\boxed{2}$ and $\boxed{3}$, where the former *s-list* $\boxed{2}$ consists entirely of *s-node*s ($\boxed{5}$) which compact to the smallest common region of $\boxed{1}$, namely $\boxed{4}$, there is a region $\boxed{6}$ which is the common region to all members of $\boxed{2}$ and whose setup must be obeyed by the members. Indeed, the region $\boxed{6}$ is *cl-fld*.

7.5 Conclusions

Clausal clitics have been selected as a set of items with very complex ordering regularities. Not all phenomena presented in this chapter have received an adequate description.

First of all, the class of clausal clitics has been defined for the purpose of this work, with the distinction between constant and inconstant clitics.

Next, some items which can occupy the clause-initial position preceding a clitic cluster have been identified with a conclusion that they are determined jointly by syntactic, discourse and prosodical factors, although there is a marked preference for a complete syntactic subtree in the initial position. Then the constraints on clitic climbing and haplology have been summarized in six points. So far, only three of them have been described formally. Finally, the order of clausal clitics within the clitic cluster has been defined, using the system of topological fields, and a constraint on clitic compaction presented.

Chapter 8

Conclusion

I do not think that this is the best moment to finish: there are too many issues which would deserve more attention and lots of others which were left unresolved or just ignored. Still, I believe I can list a few achievements before mentioning a failure or two:

1. Arguments have been presented in favour of the specific combination of FGD as a well-founded linguistic theory, and RSRL as an adequate constraint-based formal language. The original arguments were based mostly on the difficulties of a multistratal description employing movement rules for linearization of dependency tree nodes. Later, the approach was justified by being applied to a range of word order phenomena in Czech, including an account of the relation between deep and surface order and its interaction with surface-level constraints.

2. The word order principles of Vilém Mathesius were shown to be compatible with the constraint-based formalism. The principal role of functional sentence perspective and the hierarchy of communicative dynamism, as assumed in FGD, has been embodied in constraints interacting with the other ordering constraints.

3. The issue of how to describe the relation between surface string and tectogrammatical representation in a compositional way was decided by adopting a flat derivation structure with function words standing as sisters to dependents (except for cases where recursive hosting of function words by other function words is appropriate). Tectogrammatical representation was formalized as a recursive list structure, morphemic string as a non-recursive list with adjacency and ordering information encoded within its individual members.

4. The empirical facts and premises of the theory have been formalized in a way which allows for the interaction of factors conditioning surface word order, where deep word order determines surface ordering when

it is underspecified by other constraints. This has been achieved by applying a few general and some construction-specific constraints to recursive topological fields, tectogrammatical representation and – in some cases – also to morphosyntactic properties of the signs involved.

5. Some regularities concerning Czech clausal clitics have been suggested, confirming the interaction of syntactic, discourse and prosodical factors, with syntactic factors playing the main role.

Unfortunately, not all goals have been achieved. Implementation of the description of a fragment of Czech is previewed as the next step, together with the necessary rephrasing of some of the descriptions into a more computationally tractable form. This is necessary in order to verify the descriptions, but also to assess chances of further development. Of course, the theoretically adequate description should ideally be kept separate from the computationally tractable version as its source, if the two cannot be the same.

Not all phenomena presented in the chapter on clitics have received an adequate description. This may lead to modification of some more or less essential formal or theoretical aspects. Before such a move is made, a more extensive application of the approach to other phenomena is previewed, not only to those related to word order and not only those present in Czech.

Appendix A

Formalization

A.1 Signature

top
- *sign* — PHONOLOGY *list*
  SYNSEM *synsem*
  SURFACE *list*
  - *lexical*
    - *basic*
    - *derived* — STEM *lexical*
      - *0-deriv*
        - *adj-deriv*
        - *fun-deriv*
        - ...
      - *pass-deriv*
      - *past-deriv*
      - ...
  - *non-lexical* — HEAD-DAUGHTER *sign*
    NONHEAD-DAUGHTERS *list(sign)*
- *synsem* — LOCAL *local*
  NONLOCAL *nonlocal*
- *local* — CATEGORY *category*
  DEEP *deep*
  CONTENT . . .
- *nonlocal* — . . .
- *category* — HEAD *head*
  VALENCY *list(synsem)*
  DEPENDENTS *list(synsem)*
  F-WORDS *list(synsem)*
  PARTICIPANTS *list(synsem)*
  FREE-MODIFIERS *list(synsem)*
  SUBJECT *list(synsem)*
  A-MORPH *a-morph*
- *deep* — STATUS *status*
  TREE *list*

head				MODIFIED *sign*
				S-NODE *s-node*
				F-WORD *bool*
				...
	noun			
	adjective			
	base			
	pass-pple			
	infinitival			
	v-noun			
	adjp			
	advp			
	...			
status				
	embedded			
	unembedded			SENTMOD *sentmod*
				...
sentmod				
	enunc			
	...			
d-node				FUN *fun*
				CB *bool*
				CORE *d-wcl*
fun				
	participant			
		act		
		pat		
		addr		
		orig		
		eff		
		id		
		mat		
	free-modifier			
		time		
			tsin	
			twen	
			thl	
			...	
		addr		
		...		
	...			
bool				
	yes			
	no			
d-wcl				...
list				
	e-list			
	ne-list			FIRST *top*
				REST *list*

s-node PHONOLOGY *list*
        FIELD *field*
        I-CENTRE *bool*

- *field*
  - *matrix-fld*
  - *embedded-fld* REGION *field*
    - *pre-cl-fld*
    - *cl-fld*
    - *rest-fld*
    - *fin-fld*
    - *dep-fld*
      - *noun-fld*
      - *pp-fld*
      - *adj-fld*
        - *l-adj-fld*
        - *h-adj-fld*
      - *adv-fld*
      - *inf-fld*
      - *aux-fld*
      - *emb-cls-fld*
        - *sconj-cls-fld*
        - *wh-cls-fld*
    - ...

A.2 Principles

A.2.1 The 'backbone' principles

A.2.1.1 Deep list composition principle (DLCP)

(168) = (103)

$$
\textit{non-lexical} \rightarrow \left(\begin{array}{l} \begin{bmatrix} \textsc{synsem} \mid \textsc{local} \mid \textsc{deep} \mid \textsc{tree}\ \boxed{5} \\ \textsc{head-daughter} \mid \textsc{synsem} \mid \textsc{local} \mid \textsc{deep} \mid \textsc{tree}\ \boxed{1} \\ \textsc{nonhead-daughters}\ \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{collect\_dlists}(\boxed{2}, \boxed{3}) \\ \wedge\ \texttt{append}(\boxed{1}, \boxed{3}, \boxed{4}) \\ \wedge\ \texttt{permute}(\boxed{4}, \boxed{5}) \end{array} \right)
$$

A.2.1.2 Surface list composition principle (SLCP)

(169) = (229)

$$
\begin{bmatrix} \textit{non-lexical} \\ \textsc{surface}\ \boxed{1} \\ \textsc{head-daughter} \mid \textsc{surface}\ \boxed{2} \\ \textsc{nonhead-daughters}\ \boxed{3} \end{bmatrix}
\rightarrow \left(\begin{array}{l} \texttt{collect\_slists}(\boxed{3}, \boxed{4}) \\ \wedge\ \texttt{append}(\boxed{2}, \boxed{4}, \boxed{5}) \\ \wedge\ \texttt{haplo\_clitics}(\boxed{5}, \boxed{6}) \\ \wedge\ \texttt{multi\_shuffle}(\boxed{6}, \boxed{1}) \end{array} \right)
$$

A.2.1.3 Valency principle (ValP)

(170) = (107)

$$
\textit{non-lexical} \rightarrow \left(\begin{array}{l} \begin{bmatrix} \textsc{synsem} \mid \textsc{local} \mid \textsc{category} \mid \textsc{valency}\ \langle\rangle \\ \textsc{head-daughter} \mid \textsc{synsem} \mid \textsc{local} \mid \textsc{category} \mid \textsc{valency}\ \boxed{1} \\ \textsc{nonhead-daughters}\ \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{collect\_synsems}(\boxed{2}, \boxed{1}) \end{array} \right)
$$

A.2.1.4 Phonology principle (PhonP)

(171) = (108)

$$
\textit{non-lexical} \rightarrow \left(\begin{array}{l} \begin{bmatrix} \textsc{phonology}\ \boxed{1} \\ \textsc{surface}\ \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{collect\_phonology}(\boxed{2}, \boxed{1}) \end{array} \right)
$$

A.2.2 Principles constraining *d-lists*

A.2.2.1 Tree as lists principle (TLP)

(172) = (109)

$$\begin{bmatrix} \textit{deep} & \\ \text{TREE} & \boxed{1} \end{bmatrix} \rightarrow \texttt{d\_list}(\boxed{1})$$

A.2.2.2 At least one NB node principle (ONBP)

(173) = (110)

$$\begin{bmatrix} \textit{deep} & \\ \text{STATUS} & \textit{unemb} \\ \text{TREE} & \boxed{1} \end{bmatrix} \rightarrow \texttt{nested\_member}([\text{CB } \textit{no}], \boxed{1})$$

A.2.2.3 Governor Position Principle (GPP)

(174) = (111)

$$\begin{bmatrix} \textit{deep} & \\ \text{TREE} & \left\langle \boxed{1} \oplus \langle \textit{d-node} \mid \boxed{2} \rangle \right\rangle \end{bmatrix} \rightarrow$$

$$\begin{array}{l} \forall_{\boxed{3}} \left(\texttt{dependent}(\boxed{3}\textit{d-node}, \boxed{1}) \quad \rightarrow \quad \boxed{3}[\text{CB } \textit{yes}]\right) \\ \land \begin{pmatrix} \forall_{\boxed{4}} \left(\texttt{dependent}(\boxed{4}\textit{d-node}, \boxed{2}) \quad \rightarrow \quad \boxed{4}[\text{CB } \textit{no}]\right) \\ \lor \quad \forall_{\boxed{5}} \exists_{\boxed{6}} \exists_{\boxed{7}} \begin{pmatrix} \left(\texttt{member}(\boxed{5}, \boxed{2}) \land \texttt{member}(\boxed{6}[\text{CB } \textit{yes}], \boxed{5})\right) \\ \rightarrow \quad \texttt{nested\_member}(\boxed{7}[\text{CB } \textit{no}], \boxed{5}) \end{pmatrix} \end{pmatrix} \end{array}$$

A.2.2.4 Non-Bound nodes Last Principle (NBLP)

(175) = (112)

$$\forall_{\boxed{4}} \forall_{\boxed{5}}$$

$$\begin{pmatrix} \begin{pmatrix} \begin{bmatrix} \textit{deep} \\ \text{TREE } \boxed{1} \end{bmatrix} \land \texttt{right\_subordinates}(\boxed{2}, \boxed{1}) \land \texttt{dependents}(\boxed{3}, \boxed{2}) \\ \land\ \texttt{member}(\boxed{4}\ [\text{CB } \textit{yes}], \boxed{3}) \land \texttt{member}(\boxed{5}[\text{CB } \textit{no}], \boxed{3}) \end{pmatrix} \\ \rightarrow \texttt{precedes}(\boxed{4}, \boxed{5}, \boxed{3}) \end{pmatrix}$$

A.2.2.5 Non-Bound Systemic Ordering Principle (NBSOP)

(176) = (113):

$$\forall\boxed{4}\ \forall\boxed{5}\ \forall\boxed{6}\ \forall\boxed{0}$$

$$\left(\begin{array}{l}\left(\begin{array}{l}\begin{bmatrix}\textit{deep}\\ \text{TREE}\ \boxed{1}\end{bmatrix} \wedge \texttt{right\_subordinates}(\boxed{2}, \boxed{1}) \wedge \texttt{dependents}(\boxed{3}, \boxed{2})\\ \wedge\ \ \boxed{3} = \boxed{4} \oplus \langle\ \boxed{5}[\text{CB } \textit{no}], \boxed{6}[\text{CB } \textit{no}]\ \mid\ \boxed{0}\ \rangle\end{array}\right)\\ \to \texttt{so\_precedes}\Big(\boxed{5}, \boxed{6}\Big)\end{array}\right)$$

A.2.3 DWO/SWO principles

A.2.3.1 Deep/Surface Order Principle (DSOP)

(177) (see §6.3):

$$\forall\boxed{5}\ \forall\boxed{6}\ \forall\boxed{7}\ \forall\boxed{8}$$

$$\left(\begin{array}{l}\left(\begin{array}{l}\begin{bmatrix}\textit{non-lexical}\\ \text{SYNSEM}\mid\text{LOCAL}\mid\text{DEEP}\mid\text{TREE}\ \boxed{1}\\ \text{SURFACE}\ \boxed{2}\\ \text{HEAD-DAUGHTER}\ \boxed{3}\\ \text{NONHEAD-DAUGHTERS}\ \boxed{4}\end{bmatrix}\\ \wedge\ \texttt{deep\_surf}(\boxed{5}, \boxed{6}, \boxed{7}, \boxed{8}, \boxed{3}, \boxed{4})\\ \wedge\ \texttt{component}(\begin{bmatrix}(\textit{rest-fld} \vee \textit{pre-cl-fld})\\ \text{FIELD}\ \boxed{9}\end{bmatrix}, \boxed{7})\\ \wedge\ \texttt{component}(\begin{bmatrix}(\textit{rest-fld} \vee \textit{pre-cl-fld})\\ \text{FIELD}\ \boxed{9}\end{bmatrix}, \boxed{8})\end{array}\right)\\ \to \left(\begin{array}{l}\left(\begin{array}{l}\texttt{d\_precedes}(\boxed{5}, \boxed{6}, \boxed{1})\\ \wedge\ \texttt{precedes}(\boxed{7}, \boxed{8}, \boxed{2})\end{array}\right)\\ \vee\ \left(\exists\boxed{9}\exists\boxed{0}\left(\begin{array}{l}\boxed{5}[\text{CB } \textit{yes}]\ \wedge\ \boxed{1}\langle\ \boxed{9}[\textit{d-list}]\mid\boxed{0}\ \rangle\\ \wedge\ \texttt{member}\Big(\boxed{5}, \boxed{9}\Big)\\ \wedge\ \texttt{component}(\textit{pre-cl-fld}, \boxed{7})\end{array}\right)\right)\\ \vee\ \left(\exists\boxed{9}\exists\boxed{0}\left(\begin{array}{l}\left(\begin{array}{l}\texttt{append}(\boxed{0}, \boxed{6}, \boxed{1})\\ \vee\ \left(\begin{array}{l}\texttt{append}(\boxed{0}, \boxed{9}[\textit{d-list}], \boxed{1})\\ \wedge\ \texttt{member}(\boxed{6}, \boxed{9})\end{array}\right)\end{array}\right)\\ \wedge\ \boxed{8}\Big[\text{I-CENTRE } \textit{yes}\Big]\end{array}\right)\right)\end{array}\right)\end{array}\right)$$

A.2.4 General principles constraining *s-list*s

A.2.4.1 Matrix Compaction Principle (MCP)

(178) = (122)

$$\textit{s-node} \ \rightarrow \\ \exists_{\boxed{1}}\ \boxed{1}[\textit{matrix-fld}]$$

$$[\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{STATUS}\ \ \textit{unembedded}\ ] \ \rightarrow \\ \forall_{\boxed{1}}\ \forall_{\boxed{2}} \left(\left(\boxed{1}[\textit{matrix-fld}] \wedge \boxed{2}[\textit{matrix-fld}]\right) \rightarrow \boxed{1} = \boxed{2}\right)$$

A.2.4.2 Planarity Principle (PlanP)

(179) = (123)

$$\forall_{\boxed{1}}\ \forall_{\boxed{2}}\ \forall_{\boxed{3}}\ \forall_{\boxed{4}}\ \forall_{\boxed{5}} \\ \left(\begin{pmatrix} \texttt{s\_list}(\langle \boxed{1}, \boxed{2}, \boxed{3}\,|\,\textit{list}\rangle) \\ \wedge\ \texttt{region}(:, \boxed{4}) \\ \wedge\ \texttt{topo\_field}(\boxed{1}, \boxed{4}, \boxed{5}) \\ \wedge\ \texttt{topo\_field}(\boxed{3}, \boxed{4}, \boxed{5}) \end{pmatrix} \rightarrow \texttt{topo\_field}(\boxed{2}, \boxed{4}, \boxed{5})\right)$$

A.2.4.3 Topological Order Principle (TOP)

(180) = (124)

$$\left[\text{SURFACE}\ \boxed{1}\textit{ne-list}\right] \rightarrow \exists_{\boxed{2}}\ \texttt{region}(\boxed{1}, \boxed{2}) \wedge \texttt{topo\_order}(\boxed{1}, \boxed{2})$$

A.2.4.4 Field Existence Principle (FEP)

(181) = (125)

$$\begin{pmatrix} \left[\text{SURFACE}\ \boxed{1}\textit{ne-list}\right] \\ \wedge\ \texttt{region}(\boxed{1}, \boxed{2}) \\ \wedge\ \texttt{field\_existence}(\boxed{2}, \boxed{3}) \end{pmatrix} \\ \rightarrow \exists_{\boxed{4}} \begin{pmatrix} \texttt{member}(\boxed{4}, \boxed{1}) \\ \wedge\ \texttt{topo\_field}(\boxed{4}, \boxed{2}, \boxed{3}) \end{pmatrix}$$

A.2.4.5 Field Uniqueness Principle (FUP)

(182) = (126)

$$\begin{pmatrix} \left[\text{SURFACE } s\right] \\ \wedge\ \texttt{region}(s, r) \\ \wedge\ \texttt{field\_uniqueness}(r, f_1, f_2) \end{pmatrix} \rightarrow \forall n_1 \forall n_2 \begin{pmatrix} \begin{pmatrix} \texttt{member}(n_1, s) \wedge \texttt{member}(n_2, s) \wedge \neg\, n_1 = n_2 \\ \wedge\ \texttt{topo\_field}(n_1, r, f_1) \\ \wedge\ \texttt{topo\_field}(n_2, r, f_2) \end{pmatrix} \\ \rightarrow\ \ f_1 = f_2 \end{pmatrix}$$

A.2.5 Constraints on *s-lists* specific to Czech

A.2.5.1 Base-of-Comparison Compaction (BCC)

(183) = (134)

$$\begin{pmatrix} \begin{bmatrix} \text{SYNSEM} \mid \text{LOCAL} \mid \text{CATEGORY} \mid \text{HEAD } \textit{noun} \\ \text{NONHEAD-DAUGHTERS } \boxed{1} \\ \text{SURFACE } \boxed{2} \bigcirc \boxed{3} \end{bmatrix} \\ \wedge\ \texttt{member}\left(\boxed{0}\left[\text{SURFACE } \boxed{3}\left\langle\left[\text{FIELD } \textit{sconj-compar-fld}\right]\right\rangle\right], \boxed{1}\right) \end{pmatrix} \rightarrow \exists_{\boxed{4}}\ \texttt{region\_setup}(\boxed{2}, \boxed{4}\textit{noun-fld})$$

A.2.5.2 Adjective and Base-of-Comparison Compaction (ABCC)

(184) = (143)

$$\begin{pmatrix} \left[\text{SURFACE } \boxed{1}\right] \\ \wedge\ \texttt{append}(\boxed{2}, \boxed{3}, \boxed{1}) \\ \wedge\ \texttt{region}(\boxed{2}, \boxed{4}\textit{adj-fld}) \\ \wedge\ \texttt{region}(\boxed{3}, \boxed{5}\textit{compar-base-fld}) \end{pmatrix} \rightarrow \begin{pmatrix} \texttt{region}(\boxed{1}, \textit{h-adj-fld}) \\ \vee\ \exists_{\boxed{6}} \begin{pmatrix} \boxed{4}\begin{bmatrix} \textit{l-adj-fld} \\ \text{REGION } \boxed{6}\textit{noun-fld} \end{bmatrix} \\ \wedge\ \boxed{5}\left[\text{REGION } \boxed{6}\right] \end{pmatrix} \end{pmatrix}$$

A.2.5.3 PP Compaction (PPC)

(185) = (152)

$$\left(\begin{array}{l} \begin{bmatrix} \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CATEGORY}\,|\,\text{HEAD } \textit{noun} \\ \text{NONHEAD-DAUGHTERS } \boxed{1} \\ \text{SURFACE } \boxed{2}\langle[\text{FIELD } \textit{prep-fld}]\rangle \bigcirc \boxed{3} \bigcirc \boxed{4} \end{bmatrix} \\ \land\ \texttt{member}([\text{SURFACE } \boxed{2}], \boxed{1}) \\ \land\ \texttt{member}([\text{SURFACE } \boxed{3}], \boxed{1}) \end{array}\right)$$

$$\rightarrow \exists_{\boxed{5}} \left(\begin{array}{l} \left(\begin{array}{l} \texttt{append}(\boxed{2}, \boxed{3}, \boxed{5}) \\ \land\ \texttt{region}(\boxed{5}, \textit{pre-cl-fld}) \\ \land\ \texttt{region\_setup}(\boxed{4}, \textit{noun-fld}) \end{array}\right) \\ \lor \left(\begin{array}{l} \texttt{shuffle}(\boxed{3}, \boxed{4}, \boxed{5}) \\ \land\ \texttt{region\_setup}(\boxed{5}, \textit{noun-fld}) \end{array}\right) \end{array}\right)$$

A.2.5.4 Clitics not First (CNF)

(186) = (163)

$$\begin{bmatrix} \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{DEEP}\,|\,\text{STATUS} \quad \textit{unembedded} \\ \text{SURFACE } \langle \boxed{1} \,|\, \boxed{0} \rangle \end{bmatrix}$$

$$\rightarrow\ \neg\ \boxed{1}[\text{FIELD}\,|\,\text{REGION } \textit{cl-fld}]$$

A.2.5.5 Initial Non-Finite Verb Compaction I (INFVC1)

(187) = (186)

$$\left(\begin{array}{l} \begin{bmatrix} \text{S}\,|\,\text{L}\,|\,\text{C}\,|\,\text{H} \begin{bmatrix} \textit{infinitival} \lor \textit{pass-pple} \\ \text{F-WORD } \textit{no} \\ \text{S-NODE } \boxed{1} \end{bmatrix} \\ \text{NONHEAD-DAUGHTERS } \boxed{2} \\ \text{SURFACE } \boxed{3} \end{bmatrix} \\ \land\ \texttt{append}(\boxed{4}, \boxed{5}, \boxed{3}) \\ \land\ \texttt{member}(\boxed{1}, \boxed{4}) \\ \land\ \texttt{member}\left(\begin{bmatrix} \text{S}\,|\,\text{L}\,|\,\text{C}\,|\,\text{H} \begin{bmatrix} \textit{verb} \\ \text{F-WORD } \textit{yes} \\ \text{S-NODE } \boxed{6} \end{bmatrix} \end{bmatrix}, \boxed{2}\right) \\ \land\ \texttt{member}(\boxed{6}, \boxed{5}) \end{array}\right)$$

$$\rightarrow \left(\begin{array}{l} \texttt{region}(\boxed{4}, \textit{pre-cl-fld}) \\ \lor\ \texttt{region}(\boxed{3}, (\textit{matrix-fld} \lor \textit{emb-cls-fld})\ ) \end{array}\right)$$

A.2.5.6 Initial Non-Finite Verb Compaction II (INFVC2)

(188) = (200)

$$\left(\begin{bmatrix} \textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{h} \begin{bmatrix} \textit{infinitival} \vee \textit{pass-pple} \\ \textsc{f-word}\ \textit{no} \\ \textsc{s-node}\ \boxed{1} \end{bmatrix} \\ \textsc{surface}\ \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{append}(\boxed{3}, \boxed{4}, \boxed{2}) \\ \wedge\ \texttt{member}(\boxed{1}, \boxed{3}) \right) \\ \rightarrow \left(\texttt{region\_setup}(\boxed{3}, \begin{bmatrix} \textit{inf-fld} \\ \textsc{region}\ \textit{pre-cl-fld} \end{bmatrix}) \\ \vee\ \texttt{region}(\boxed{2}, \textit{inf-fld}) \right)$$

A.2.5.7 Initial Noun Compaction (INC)

(189) = (203)

$$\left(\begin{bmatrix} \textsc{s}\,|\,\textsc{l}\,|\,\textsc{c}\,|\,\textsc{h} \begin{bmatrix} \textit{noun} \\ \textsc{s-node}\ \boxed{1} \end{bmatrix} \\ \textsc{surface}\ \boxed{2} \end{bmatrix} \\ \wedge\ \texttt{append}(\boxed{3}, \boxed{4}, \boxed{2}) \\ \wedge\ \texttt{member}(\boxed{1}, \boxed{3}) \right) \\ \rightarrow \left(\left(\texttt{region\_setup}(\boxed{3}, \begin{bmatrix} \textit{noun-fld} \\ \textsc{region}\ \textit{pre-cl-fld} \end{bmatrix}) \\ \wedge\ \texttt{region\_setup}(\boxed{4}, \textit{rest-fld}) \right) \\ \vee\ \texttt{region}(\boxed{2}, \textit{noun-fld}) \right)$$

A.2.5.8 Partial Compaction (PartComp)

(190) = (228)

$$
\left(\begin{array}{l}
\left[\textsc{surface}\ \boxed{1}\right] \\
\wedge\ \texttt{\textbf{shuffle}}(\boxed{2}, \boxed{3}, \boxed{1}) \\
\wedge \left(\begin{array}{l}
\texttt{\textbf{member}}(\boxed{7}, \boxed{2}) \\
\rightarrow \left(\begin{array}{l}
\texttt{\textbf{component}}(\textit{pre-cl-fld}, \boxed{7}) \\
\vee \left(\begin{array}{l}
\boxed{7}\left[\textsc{f}\ \boxed{5}\left[\textsc{r}\ \textit{cl-fld}\right]\right] \\
\wedge\ \neg\ \left[\textsc{s} \mid \textsc{l} \mid \textsc{c} \mid \textsc{a-morph}\ \textit{finite}\right] \\
\wedge \left(\begin{array}{l}
\left[\textsc{s} \mid \textsc{l} \mid \textsc{c} \mid \textsc{head}\ \textit{infinitival}\right] \\
\vee \left(\begin{array}{l}
\left[\textsc{nonhead-daughters}\ \boxed{6}\right] \\
\wedge\ \texttt{\textbf{member}}(\left[\textsc{surface}\ \boxed{5}\right], \boxed{6}) \\
\wedge\ \neg\ \left[\ \textsc{s} \mid \textsc{l} \mid \textsc{c} \mid \textsc{head}\ (\textit{v-noun} \vee \textit{adjp} \vee \textit{advp})\ \right]
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{array}\right)
$$

$$
\rightarrow \exists_{\boxed{4}} \left(\begin{array}{l}
\texttt{\textbf{region\_setup}}(\boxed{3}, \boxed{4}) \\
\vee\ \texttt{\textbf{region}}(\boxed{1}, \boxed{4})
\end{array}\right)
$$

A.2.5.9 Clitic Compaction (ClitComp)

(191) = (168)

$$
\forall_{\boxed{5}} \left(\begin{array}{l}
\left(\begin{array}{l}
\left[\textsc{surface}\ \boxed{0} \oplus \boxed{1}\right] \\
\wedge\ \texttt{\textbf{region}}(\boxed{1}, \boxed{4}) \\
\wedge\ \texttt{\textbf{shuffle}}(\boxed{2}, \boxed{3}, \boxed{1}) \\
\wedge \left(\begin{array}{l}
\texttt{\textbf{member}}(\boxed{5}, \boxed{2}) \\
\rightarrow \boxed{5}\left[\textsc{field} \mid \textsc{region} \left[\begin{array}{l} \textit{cl-fld} \\ \textsc{region}\ \boxed{4} \end{array}\right]\right]
\end{array}\right)
\end{array}\right) \\
\rightarrow \exists_{\boxed{6}} \left(\begin{array}{l}
\boxed{5}\left[\textsc{field} \mid \textsc{region}\ \boxed{6}\right] \\
\wedge\ \texttt{\textbf{region\_setup}}(\boxed{2}, \boxed{6})
\end{array}\right)
\end{array}\right)
$$

A.3 Relations

For every relation, the types of arguments are specified. The relations are defined either in standard RSRL, or in the RSRL Prolog-like notation.

A.3.1 $\texttt{append}(\textit{list}, \textit{list}, \textit{list})$

This relation is satisfied iff `arg3` consists of `arg1` followed by `arg2`.

(192) $\forall x\, \forall y\, \forall z\, [\, \texttt{append}(x, y, z) \leftrightarrow$
$[\, x \sim \textit{elist} \wedge y \approx z \wedge y \sim \textit{list}\,] \vee$
$\exists h\, \exists t_1\, \exists t_2\, [$
$x\textsc{first} \approx h \wedge x\textsc{rest} \approx t_1 \wedge$
$z\textsc{first} \approx h \wedge z\textsc{rest} \approx t_2 \wedge$
$\texttt{append}(t_1, y, t_2)\,]\,]\,]$

A.3.2 $\texttt{collect\_dlists}(\textit{list}, \textit{list})$

This relation holds between a list of signs (`arg1`) and a list containing their *d-lists* (`arg2`). The order of the corresponding signs and *d-lists* is preserved.

(193) $\texttt{collect\_dlists}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle$
$\texttt{collect\_dlists}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle[\textsc{synsem}|\textsc{local}|\textsc{deep}|\textsc{tree}\ \boxed{1}] \,|\, \boxed{2}\rangle$
$\wedge\ y\langle\boxed{1}|\boxed{3}\rangle \wedge \texttt{collect\_dlists}(\boxed{2}, \boxed{3})$

A.3.3 $\texttt{collect\_phonology}(\textit{list}, \textit{list})$

This relation holds between a list of objects with a top-level list-valued attribute PHONOLOGY (`arg1`) and a list containing the elements of all the phonology lists (`arg2`), with their order preserved.

(194) $\texttt{collect\_phonology}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle$
$\texttt{collect\_phonology}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle[\textsc{phonology}\ \boxed{1}] \,|\, \boxed{2}\rangle \wedge\ \texttt{append}(\boxed{1}, \boxed{3}, y)$
$\wedge\ \texttt{collect\_phonology}(\boxed{2}, \boxed{3})$

A.3.4 $\texttt{collect\_slists}(\textit{list}, \textit{list})$

This relation holds between a list of signs (`arg1`) and a list containing their *s-lists* (`arg2`). The order of the corresponding signs and *s-lists* is preserved.

(195) $\texttt{collect\_slists}(x, y) \overset{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle$
$\texttt{collect\_slists}(x, y) \overset{\forall}{\Longleftarrow}$
$x\langle[\text{SURFACE}\,\boxed{1}]\,|\,\boxed{2}\rangle \wedge y\langle\boxed{1}|\boxed{3}\rangle \wedge \texttt{collect\_slists}(\boxed{2},\boxed{3})$

A.3.5 collect_synsems(*list*, *list*)

This relation holds between a list of signs (`arg1`) and a list of their *synsem* objects (`arg2`). The order of the corresponding signs and synsems is preserved.

(196) $\texttt{collect\_synsems}(x, y) \overset{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle$
$\texttt{collect\_synsems}(x, y) \overset{\forall}{\Longleftarrow}$
$x\langle[\text{SYNSEM}\,\boxed{1}]\,|\,\boxed{2}\rangle \wedge y\langle\boxed{1}|\boxed{3}\rangle \wedge \texttt{collect\_synsems}(\boxed{2},\boxed{3})$

A.3.6 component(*top*, *top*)

This relation replaces `exist/2` of Penn (1999b). It holds iff `arg1` is a component of `arg2`. The relation can be defined only with respect to a specific signature. Richter (2000, p. 358) presents a schema for defining the relation relative to a signature Σ with a finite set of attributes A. The set of clauses that define the relation `component/2` is the smallest set C such that

(197) $\texttt{component}(x, y) \overset{\forall}{\Longleftarrow} x = y \in C$, and
for each $\alpha \in \mathsf{A}$
$\texttt{component}(x, y) \overset{\forall}{\Longleftarrow} {}^{y}[\alpha\ \boxed{1}] \wedge \texttt{component}(x,\boxed{1}) \in C.$

A.3.7 d_list(*list*)

This relation is satisfied iff `arg` consists of exactly one *d-node* and $0 - n$ lists, each satisfying `d_list(l)` again. Its satisfaction guarantees that the list in `arg` is a proper representation of dependency tree.

(198) $\forall x\ [\ \texttt{d\_list}(x) \leftrightarrow$
$[\ \exists y\ \exists z\ [\ x\text{FIRST} \approx \textit{d-node}\ \wedge\ x\text{REST} \approx z\ \wedge\ \texttt{d\_lists\_only}(z)\ ] \vee$
$[\ x\text{FIRST} \approx y\ \wedge\ \texttt{d\_list}(y)\ \wedge\ \texttt{d\_list}(z)\ ]\ ]\ ]$

$\forall x\ [\ \texttt{d\_lists\_only}(x) \leftrightarrow$
$x \sim \textit{e-list} \vee$
$\exists t\, \exists y\, [\ x\text{FIRST} \sim t\ \wedge$
$\texttt{d\_list}(t) \wedge$
$x\text{REST} \approx y \wedge$
$\texttt{d\_lists\_only}(y)\ ]\ ]$

A.3.8 d_precedes(*d-node*, *d-node*, *list*)

Relation d_precedes/3 makes sure that a *d-node* (arg1), precedes another *d-node* (arg2) in a *d-list* (arg3). Each *d-node* is either the governor of the local tectogrammatical tree or its immediate dependent.

(199) $\texttt{d\_precedes}(x, y, z) \overset{\forall}{\Longleftarrow}$
$\texttt{local\_dwo}(z, \boxed{1}) \wedge \texttt{precedes}(x, y, \boxed{1})$

A.3.9 deep_surf(*d-node*, *d-node*, *s-node*, *s-node*, *sign*, *list*)

The relation deep_surf/6 pairs two distinct *d-nodes* (arg1 and arg2) from the local tectogrammatical tree with their *s-node* counterparts (arg3 and arg4) by inspecting the head daughter sign (arg5) and the signs in the list of non-head daughters (arg6).

The definition uses two auxiliary relations. The relation deep_surf1/5, which uses one list for both the head daughter and the non-head daughters, identifies two distinct signs from the local tree and uses them as an argument in two calls of deep_surf1/3. This relation identifies the sign's *d-list* ($\boxed{1}$), the governor's *d-node* (d) and the corresponding *s-node* (s).

(200) $\texttt{deep\_surf}(d_1, d_2, s_1, s_2, h, d) \overset{\forall}{\Longleftarrow}$
$\boxed{1}\langle h|d\rangle \wedge \texttt{deep\_surf1}(d_1, d_2, s_1, s_2, \boxed{1})$

(201) $\texttt{deep\_surf1}(d_1, d_2, s_1, s_2, x) \overset{\forall}{\Longleftarrow}$
$\texttt{member}(\boxed{1}, x) \wedge \texttt{member}(\boxed{2}, x) \wedge \neg\boxed{1} = \boxed{2}$
$\wedge\ \texttt{deep\_surf2}(\boxed{1}, d_1, s_1) \wedge \texttt{deep\_surf2}(\boxed{2}, d_2, s_2)$

(202) $\texttt{deep\_surf2}(x, d, s) \overset{\forall}{\Longleftarrow}$
$${}^{x}\left[\text{SYNSEM}\,|\,\text{LOCAL}\begin{bmatrix}\text{DEEP}\,|\,\text{TREE}\ \boxed{1}\\ \text{CATEGORY}\,|\,\text{HEAD}\,|\,\text{S-NODE}\ s\end{bmatrix}\right]$$
$\wedge\ \texttt{member}({}^{d}[\textit{d-node}], \boxed{1})$

A.3.10 dependent(*top*, *list*)

This relation is satisfied iff arg1 is a member of a member of arg2. With *d-list* as arg2, the relation is satisfied if arg2 is a subtree rooted in r and arg1 is either a dependent node of r or a dependent subtree of such a node. In order to pick only nodes dependent on r, arg1 should be specified as *d-node*.

(203) $\forall x\, \forall y\, [\ \texttt{dependent}(x, y) \leftrightarrow$
$\exists\, h\, [\, h \approx y\text{FIRST} \wedge \texttt{member}(x, h)\ ] \vee$
$\exists\, t\, [\, t \approx y\text{REST} \wedge \texttt{dependent}(x, t)\ ]\ ]$

A.3.11 $\texttt{dependents}(list, list)$

This relation is satisfied iff `arg1` is the list of *d-node* members of the members of `arg2`. With *d-list* as `arg2`, the relation is satisfied if `arg2` is a subtree rooted in r and `arg1` is the list of dependent nodes of r.

(204) $\forall x \forall y\, [\ \texttt{dependents}(x, y) \leftrightarrow$
$[\ x \sim elist\ \wedge\ y \sim elist\ ] \vee$
$\exists h_1\ \exists t_1\ \exists h_2\ \exists t_2\ [$
$[\ x\text{FIRST} \approx h_1\ \wedge\ h_1 \sim d\text{-}node\ \wedge\ x\text{REST} \approx t_1\ \wedge$
$y\text{FIRST} \approx h_2\ \wedge\ y\text{REST} \approx t_2\ \wedge$
$\texttt{member}(h_1, h_2)\ \wedge\ \texttt{dependents}(t_1, t_2)\ ] \vee$
$[\ h_2 \sim d\text{-}node\ \wedge\ \texttt{dependents}(x, t_2)\ ]\ ]\ ]$

A.3.12 $\texttt{field\_existence}(field, field)$

This relation holds if `arg2` is an obligatory field in region `arg1`. This is one of the three relations to which the region setup tables are translated. I will show only a sample definition clause.

(205) $\texttt{field\_existence}(x, y) \stackrel{\forall}{\Longleftarrow}$
$^{x}[matrix\text{-}fld] \wedge {}^{y}[pre\text{-}cl\text{-}fld]$

A.3.13 $\texttt{field\_uniqueness}(field, field, field)$

This relation holds if `arg1` is a region in which the field specified in `arg2` and `arg3` may occur at most once. This is one of the three relations to which the region setup tables are translated. I will show only a sample definition clause.

(206) $\texttt{field\_uniqueness}(x, y, z) \stackrel{\forall}{\Longleftarrow}$
$^{x}[matrix\text{-}fld] \wedge {}^{y}[cl\text{-}fld] \wedge {}^{z}[cl\text{-}fld]$

A.3.14 $\texttt{haplo\_clitics}(list, list)$

This relation holds if the *s-list* in `arg2` is identical to the *s-list* in `arg1`, except for cases when there is an *s-node* in `arg1` whose value of FIELD is identical to another *s-node* in `arg1` and which has the REGION attribute set to *cl-fld*. In this case, only one of the two or more such *s-node*s is on the *s-list* in `arg2`.

(207) $\texttt{haplo\_clitics}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle$
$\texttt{haplo\_clitics}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle \boxed{1}[\text{FIELD}\ \boxed{2}[\text{REGION}\ \textit{cl-fld}]\,] \,|\, \boxed{3} \rangle$
$\wedge\ \texttt{member}([\,\text{FIELD}\ \boxed{2}], \boxed{3})$
$\wedge\ \texttt{haplo\_clitics}(\boxed{3}, y)$
$\texttt{haplo\_clitics}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle \boxed{1}[\text{FIELD}\ \boxed{2}[\text{REGION}\ \textit{cl-fld}]\,] \,|\, \boxed{3} \rangle \wedge y\langle \boxed{1}|\boxed{4} \rangle$
$\wedge\ \neg\ \texttt{member}([\,\text{FIELD}\ \boxed{2}], \boxed{3})$
$\wedge\ \texttt{haplo\_clitics}(\boxed{3}, \boxed{4})$

A.3.15 left_subordinates($list$, $list$)

This relation satisfied iff `arg1` is the prefix of `arg2` immediately preceding a *d-node*, i.e. if `arg2` is a subtree rooted in r and `arg1` is a list of subtrees left-dependent on r.

(208) $\forall x\, \forall y\, [\ \texttt{left\_subordinates}(x, y) \leftrightarrow$
$\exists\, z\, \exists\, h\, [$
$h \approx z\text{FIRST} \wedge$
$h \sim \textit{d-node} \wedge$
$\texttt{append}(x, z, y)\ ]\,]$

A.3.16 local_dwo($list$, $list$)

This relation holds between a *d-list* (`arg1`) and a list of *d-node*s (`arg2`) just in case the *d-nodes* in `arg2` are nodes of the topmost tree of `arg1`, ordered according to DWO.

(209) $\texttt{local\_dwo}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle$
$\texttt{local\_dwo}(x, y) \stackrel{\forall}{\Longleftarrow}$
$x\langle \boxed{1}|\boxed{2} \rangle \wedge y\langle \boxed{3}|\boxed{4} \rangle$
$\wedge\ \texttt{node}(\boxed{3}\textit{d-node}, \boxed{1}) \wedge \texttt{local\_dwo}(\boxed{2}, \boxed{4})$

A.3.17 member(top, $list$)

This relation is satisfied iff `arg1` is a member of `arg2`. With *d-list* as `arg2`, the relation is satisfied if `arg2` is a subtree rooted in r and `arg1` is r or a dependent subtree of r.

(210) $\texttt{member}(x,y) \overset{\forall}{\Longleftarrow}$
$x \approx y\text{FIRST}$
$\texttt{member}(x,y) \overset{\forall}{\Longleftarrow}$
$\exists z\,[\; z \approx y\text{REST} \land \texttt{member}(x,z)\;]$

A.3.18 `multi_shuffle`(*list*, *list*)

This relation holds between a list of lists (`arg1`), where all the lists are shuffled into another list (`arg2`). In the recursive case, the first two lists of `arg1` are shuffled together using `shuffle/3` and `multi_shuffle/2` is invoked again with the result as the head of `arg1`. If only a single list remains in `arg1`, `arg1` is identified with `arg2`.

(211) $\texttt{multi\_shuffle}(x,y) \overset{\forall}{\Longleftarrow}$
$x\langle\rangle \land y\langle\rangle$
$\texttt{multi\_shuffle}(x,y) \overset{\forall}{\Longleftarrow}$
$x\langle\boxed{0}\rangle \land y\langle\boxed{0}\rangle$
$\texttt{multi\_shuffle}(x,y) \overset{\forall}{\Longleftarrow}$
$x\langle\boxed{0}|\langle\boxed{1}|\boxed{2}\rangle\rangle \land \texttt{shuffle}(\boxed{0},\boxed{1},\boxed{3})$
$\land\ \texttt{multi\_shuffle}(\langle\boxed{3}|\boxed{2}\rangle, y)$

A.3.19 `nested_member`(*top*, *list*)

This relation satisfied iff `arg1` is a member of `arg2` or of another list which is recursively embedded in it. With *d-list* as `arg2`, the relation is satisfied if `arg2` is a subtree rooted in r and `arg1` is r or either a subordinate node or a subtree of r.

(212) $\texttt{nested\_member}(x,y) \overset{\forall}{\Longleftarrow}$
$x \approx y\text{FIRST}$
$\texttt{nested\_member}(x,y) \overset{\forall}{\Longleftarrow}$
$\exists\, h\,[\, h \approx y\text{FIRST} \land \texttt{nested\_member}(x,h)\;]$
$\texttt{nested\_member}(x,y) \overset{\forall}{\Longleftarrow}$
$\exists\, t\,[\, t \approx y\text{REST} \land \texttt{nested\_member}(x,t)\;]$

A.3.20 `node`(*top*, *list*)

This relation is very similar to the relation `dependent`, the only difference being that the first argument can also be identified with the root of the local tree. It is satisfied iff `arg1` is either a member of `arg2` or a member of a member of `arg2`. With *d-list* as `arg2`, the relation is satisfied if `arg2` is a subtree rooted in r and `arg1` is one of the following: r, a dependent subtree of r, a dependent node of r, or a dependent subtree of such a node. In order to pick only nodes within a local tree rooted in r, `arg1` should be specified as *d-node*.

(213) $$\forall x \, \forall y \, [\, \texttt{node}(x,y) \leftrightarrow \\ x \approx y\text{FIRST} \lor \\ \exists\, h \,[\, h \approx y\text{FIRST} \land \texttt{member}(x,h)\,] \lor \\ \exists\, t \,[\, t \approx y\text{REST} \land \texttt{node}(x,t)\,]\,]$$

A.3.21 permute($list, list$)

This relation holds if `arg2` is a permutation of `arg2`. The auxiliary relation `pick/3` holds if the list `arg3` is the result of removing an element `arg1` from the list `arg2`.

(214) $$\texttt{permute}(x,y) \overset{\forall}{\Longleftarrow} \\ x\langle\rangle \land y\langle\rangle \\ \texttt{permute}(x,y) \overset{\forall}{\Longleftarrow} \\ x\langle\boxed{1}|\boxed{2}\rangle \land \texttt{permute}(\boxed{2},\boxed{3}) \\ \land\ \texttt{pick}(\boxed{1},y,\boxed{3}) \\ \texttt{pick}(x,y,z) \overset{\forall}{\Longleftarrow} \\ y\langle x|z\rangle \\ \texttt{pick}(x,y,z) \overset{\forall}{\Longleftarrow} \\ y\langle\boxed{1}|\boxed{2}\rangle \land z\langle\boxed{1}|\boxed{3}\rangle \\ \land\ \texttt{pick}(x,\boxed{2},\boxed{3})$$

A.3.22 precedes($top, top, list$)

This relation is satisfied iff the first two arguments occur as members of `arg3` and `arg1` precedes `arg2` in `arg3`. The list should be instantiated to a finite number of elements. The definition involves an auxiliary relation `precedes1`($top, list, list$) whose `arg1` is `arg1` of `precedes1` and whose `arg3` is the rest of the list following its `arg1`.

(215) $$\forall x \, \forall y \, \forall z \ [\ \texttt{precedes}(x,y,z) \leftrightarrow \\ \exists\, z_2\, [\\ \texttt{precedes1}(x,z,z_2) \land \\ \texttt{member}(y,z_2)\ ]\ ] \\ \forall x \, \forall z \, \forall z_2 \ [\ \texttt{precedes1}(x,z,z_2) \leftrightarrow \\ \exists\, t\, [\ t \approx z\text{REST}\ \land\ [\\ [\ x \approx z\text{FIRST}\ \land\ z_2 \approx t\ ] \lor \\ \texttt{precedes1}(x,t,z_2)\ ]\ ]\ ]$$

A.3.23 region($list, field$)

This relation is due to Penn (1999b), here in the rephrased version of Richter (2000, p. 344). However, any errors are mine. It holds between an *s-list* (arg1) and an object of the type *region* just in case the latter is the smallest region that contains all *s-node*s on the *s-list*.

(216)
$$
\begin{array}{l}
\texttt{region}(s, r) \overset{\forall}{\Longleftarrow} \\
\quad \texttt{s\_list}(s\langle {}^{d}[\textsc{field}\ \boxed{1}] \mid \boxed{2} \rangle) \\
\quad \wedge\ \texttt{region\_recurse}(\boxed{2}, \boxed{1}, r) \\
\texttt{region\_recurse}(s, r_1, r) \overset{\forall}{\Longleftarrow} \\
\quad s\langle\rangle \wedge r_1 = r \\
\texttt{region\_recurse}(s, r_1, r) \overset{\forall}{\Longleftarrow} \\
\quad s\langle {}^{d}[\textsc{field}\ \boxed{1}] \mid \boxed{2} \rangle \\
\quad \wedge\ \exists_{\boxed{3}} \left(\begin{array}{l}
\left(\begin{array}{l}
\left(\texttt{component}(\boxed{3}, \boxed{1}) \wedge \boxed{3} = r_1 \right) \\
\vee\ \forall_{\boxed{4}} \left(\begin{array}{l}
\texttt{component}(\boxed{4}, \boxed{1}) \rightarrow \neg\ \boxed{4} = r_1 \\
\wedge\ \exists_{f_1} \exists_{f_2} \left(\begin{array}{l}
\texttt{component}({}^{f_1}[\textsc{region}\ \boxed{3}], r_1) \\
\wedge\ \texttt{component}({}^{f_2}[\textsc{region}\ \boxed{3}], \boxed{1}) \\
\wedge\ \neg f_1 = f_2
\end{array} \right)
\end{array} \right)
\end{array} \right) \\
\wedge\ \texttt{region\_recurse}(\boxed{2}, \boxed{3}, r)
\end{array} \right)
\end{array}
$$

The definition uses an auxiliary relation region_recurse/3: arg1 contains the current rest of the *s-list* arg2 the smallest region as determined from the beginning of the *s-list* up to the current *s-node*, and arg3 the result, i.e., the overall smallest region. The important piece of the second clause in the definition of region_recurse/3 consists of two disjuncts. If the currect smallest region r_1 is a component of $\boxed{1}$, the FIELD value of the current *s-node*, it remains the smallest region even if the current *s-node* is taken into account and region_recurse/3 is called again for the rest of *s-list* ($\boxed{2}$) with r_1 ($= \boxed{3}$) as arg2. If r_1 is not a component of $\boxed{4}$, a larger region must be found which includes the current *s-node*($\boxed{1}$). Such region ($\boxed{3}$) must be a component both of r_1 and $\boxed{1}$, and the next smaller regions in the two structures (f_1 and f_2) must be different. Only then it is guaranteed that $\boxed{3}$ is the *smallest* common region. As in the first case, region_recurse/3 is called again for the rest of *s-list* ($\boxed{2}$) with $\boxed{3}$ ($\neq r_1$) as arg2.

A.3.24 region_setup($list, field$)

This relation holds if the principles TOP, FEP and FUP are satisfied for a *s-list* (arg1) and a region (arg2). It is used when the principles should not apply to the whole value of SURFACE.

$$(217)\quad \texttt{region_setup}(s,r) \stackrel{\forall}{\Longleftarrow}\\
\texttt{region}(s,r)\\
\wedge\ \texttt{topo_order}(s,r)\\
\wedge \begin{pmatrix} \texttt{field_existence}(r,f) \\ \rightarrow \exists n \begin{pmatrix} \texttt{member}(n,s) \\ \wedge\ \texttt{topo_field}(n,r,f) \end{pmatrix} \end{pmatrix}\\
\wedge \begin{pmatrix} \texttt{field_uniqueness}(r,f_1,f_2) \\ \rightarrow \forall n_1 \forall n_2 \begin{pmatrix} \begin{pmatrix} \texttt{member}(n_1,s) \wedge \texttt{member}(n_2,s) \wedge n_1 \neq n_2 \\ \wedge\ \texttt{topo_field}(n_1,r,f_1) \\ \wedge\ \texttt{topo_field}(n_2,r,f_2) \end{pmatrix} \\ \rightarrow\ f_1 = f_2 \end{pmatrix} \end{pmatrix}$$

A.3.25 right_subordinates(*list*, *list*)

This relation satisfied iff arg1 is the suffix of arg2 immediately following a *d-node*, i.e. if arg2 is a subtree rooted in r and arg1 is a list of subtrees right-dependent on r.

$$(218)\quad \forall x\, \forall y\, [\ \texttt{right_subordinates}(x,y) \leftrightarrow\\
\exists\, z\, [\\
z \approx y\text{FIRST} \wedge\\
z \sim \textit{d-node} \wedge\\
x \approx y\text{REST}\ ] \vee\\
\exists\, t\, [\\
t \approx y\text{REST} \wedge\\
\texttt{right_subordinates}(x,t)\ ]\,]$$

A.3.26 s_list(*list*)

This relation holds if arg is a list of *s-node*s.

$$(219)\quad \texttt{s_list}(x) \stackrel{\forall}{\Longleftarrow}\\
x\langle\rangle\\
\texttt{s_list}(x) \stackrel{\forall}{\Longleftarrow}\\
x\langle \textit{s-node} | \boxed{1} \rangle \wedge\ \texttt{s_list}(\boxed{1})$$

A.3.27 shuffle(*list*, *list*, *list*)

This relation is satisfied iff arg3 consists exactly of all elements of arg1 and all elements of arg2, and the relative order of each pair of elements of arg1 and arg2 is preserved in arg3.[1]

[1] I ignore the distinction between lists and chains. Otherwise, shuffle/3 would have to be defined using a notation which would allow a certain description be interpreted as a list or as a chain, which is exactly what Richter (2000, p. 269) does by using *tape descriptions*.

(220) $\mathtt{shuffle}(x, y, z) \stackrel{\forall}{\Longleftarrow}$
$x\langle\rangle \wedge y\langle\rangle \wedge z\langle\rangle$
$\mathtt{shuffle}(x, y, z) \stackrel{\forall}{\Longleftarrow}$
$x\langle\boxed{0}|\boxed{1}\rangle \wedge z\langle\boxed{0}|\boxed{2}\rangle \wedge \mathtt{shuffle}(\boxed{1}, y, \boxed{2})$
$\mathtt{shuffle}(x, y, z) \stackrel{\forall}{\Longleftarrow}$
$y\langle\boxed{0}|\boxed{2}\rangle \wedge z\langle\boxed{0}|\boxed{3}\rangle \wedge \mathtt{shuffle}(x, \boxed{2}, \boxed{3})$

A.3.28 so_precedes(d-$node$, d-$node$)

This relation is satisfied iff arg1's FUN value precedes arg2's FUN value in the list of tectogrammatical function labels, ordered according to SO. The definition given here assumes SO for Czech, which is defined by means of an auxilliary relation so/1. The arg of so/1 is the list of functor types, corresponding to column 2 of the table in Appendix C. For the sake of brevity, so/1 is defined using the angle brackets notation for lists.

(221) $\forall x_1 \forall x_2$ [$\mathtt{so\_precedes}(x_1, x_2) \leftrightarrow$
$\exists y_1 \exists y_2 \exists z$ [
$x_1\text{FUN} \approx y_1 \;\wedge\; x_2\text{FUN} \approx y_2$
$\wedge\; \mathtt{so}(z) \wedge \mathtt{precedes}(y_1, y_2, z)$]]
$\forall z$ [$\mathtt{so}(z) \leftrightarrow$
$\mathtt{so}(\langle att, mat, act, cond, tsin, twen, thl, ttill, tfhl, tpar, caus, reg, resl, cncs, aim, mann, ext, norm, crit, subs, acmp, restr, loc, means, diff, dir1, dir2, addr, orig, pat, ben, dir3, eff, cpr, app, rstr, id, des\rangle)$]

A.3.29 topo_field(s-$node$, $field$, $field$)

This relation is due to Penn (1999b) and Richter (2000, p. 343). It identifies the topological field (arg3) of a s-$node$ (arg1) relative to a region (arg2). If the region r equals the FIELD value $\boxed{1}$ of the s-$node$ d, then the field f is that value. Otherwise, f must be sought deeper inside $\boxed{1}$, where r is the next higher region.

(222) $\mathtt{topo\_field}(n, r, f) \stackrel{\forall}{\Longleftarrow}$
$^{n}\begin{bmatrix} dom\text{-}obj \\ \text{FIELD } \boxed{1} \end{bmatrix} \wedge \boxed{1} = r \wedge f = \boxed{1}$
$\mathtt{topo\_field}(n, r, f) \stackrel{\forall}{\Longleftarrow}$
$^{n}\begin{bmatrix} dom\text{-}obj \\ \text{FIELD } \boxed{1} \end{bmatrix} \wedge \neg\, \boxed{1} = r \wedge \mathtt{component}(f, \boxed{1}) \wedge {}^{f}[\text{REGION } r]$

A.3.30 `topo_order`$(list, field)$

This relation is a modified version of a relation with the same name in Penn (1999b). Here it follows the RSRL notation of Richter (2000). It is one of the three relations to which the region setup tables are translated. The relation hold between a *s-list* (`arg1`) and a region (`arg2`) iff the topological ordering defined for the region is respected in the *s-list*. The definition below is only a sample of a number of clauses, which are necessary in order to translate the region setup tables into RSRL descriptions. It orders the fields *pre-cl-fld, cl-fld, rest-fld* and *fin-fld*, without specifying their region.

(223)
$$
\begin{array}{l}
\mathtt{topo\_order}(s,r) \stackrel{\forall}{\Longleftarrow} \\
\quad s\langle\rangle \\
\mathtt{topo\_order}(s,r) \stackrel{\forall}{\Longleftarrow} \\
\quad s\langle \boxed{1} | \boxed{2} \rangle \wedge \mathtt{topo\_order}(\boxed{2}, r) \wedge \mathtt{topo\_field}(\boxed{1}, r, f) \\
\quad \wedge \left(\begin{array}{l} {}^{f}[\textit{cl-fld}] \rightarrow \\ \forall n_2 \left(\begin{array}{l} \mathtt{member}(n_2, \boxed{2}) \rightarrow \\ \exists f_2\ \mathtt{topo\_field}(n_2, r,\ {}^{f_2}[\neg\ \textit{pre-cl-fld}]) \end{array} \right) \end{array} \right) \\
\quad \wedge \left(\begin{array}{l} {}^{f}[\textit{rest-fld}] \rightarrow \\ \forall n_2 \left(\begin{array}{l} \mathtt{member}(n_2, \boxed{2}) \rightarrow \\ \exists f_2\ \mathtt{topo\_field}(n_2, r,\ {}^{f_2}[\neg \left(\begin{array}{l} \textit{pre-cl-fld} \\ \vee\ \textit{cl-fld} \end{array} \right)]) \end{array} \right) \end{array} \right) \\
\quad \wedge \left(\begin{array}{l} {}^{f}[\textit{fin-fld}] \rightarrow \\ \forall n_2 \left(\begin{array}{l} \mathtt{member}(n_2, \boxed{2}) \rightarrow \\ \exists f_2\ \mathtt{topo\_field}(n_2, r,\ {}^{f_2}[\neg \left(\begin{array}{l} \textit{pre-cl-fld} \\ \vee\ \textit{cl-fld} \\ \vee\ \textit{rest-fld} \end{array} \right)]) \end{array} \right) \end{array} \right)
\end{array}
$$

According to the second clause, the relation holds between a *s-list* (s) and a region r just in case (i) it holds between the tail of s ($\boxed{2}$) and the region and (ii) if the field f of the current head $\boxed{1}$ of s relative to r is one of the non-initial fields in the topological order of r, none of the fields f_2 preceding f in the topological order may be present in any *s-node* n_2 following $\boxed{1}$ in s.

Appendix B

The setup of topological regions

Region	Field	Order	Occupancy
matrix-fld	*pre-cl-fld*	1	1
	cl-fld	2	$\leq$1
	rest-fld	3	any
	fin-fld	4	$\leq$1
pre-cl-fld	*dep-fld*	1	1
rest-fld	*dep-fld*	any	any
noun-fld	*l-adj-fld*	1	any
	noun-fld	2	$\geq$1
	emb-cls-fld	3	$\leq$1
	h-adj-fld	$\geq$4	any
	pp-fld	$\geq$4	any
	compar-base-fld	$\geq$4	$\leq$1
pp-fld	*prep-fld*	1	1
	noun-fld	2	1
l-adj-fld	*adv-fld*	1	any
	adj-fld	2	1
h-adj-fld	*adv-fld*	1	any
	adj-fld	2	1
	pp-fld	$\geq$3	any
	emb-cls-fld	$\geq$3	any
	noun-fld	$\geq$3	any
	compar-base-fld	$\geq$3	$\leq$1
adv-fld	*adv-fld*	1	$\leq$1
	adv-fld	2	1
compar-base-fld	*sconj-compar-fld*	1	1
	noun-fld	2	1

Region	Field	Order	Occupancy
wh-cls-fld	*pre-cl-fld*	1	1
	cl-fld	2	≤1
	rest-fld	3	≥1
	fin-fld	4	≤1
sconj-cls-fld	*sconj-fld*	1	1
	pre-cl-fld	2	≤1
	cl-fld	3	≤1
	rest-fld	4	any
	fin-fld	5	≤1
inf-fld	*rest-fld*	1	any
	cl-fld	2∨5	≤1
	rest-fld	3	any
	inf-fld	4	1
	rest-fld	6	any
cl-fld	*cl-lis-fld*	1	≤2
	cl-be-fld	2	≤1
	cl-rfl-fld	3∨4	≤1
	cl-ethdat-fld	3∨4	≤1
	cl-freedat-fld	5	any
	cl-dat-fld	6	any
	cl-acc-fld	7	any
	cl-gen-fld	8	any
	cl-ins-fld	9	any
	cl-nom-fld	any	≤1
	cl-uz-fld	any	≤1
	cl-pry-fld	any	≤1
	cl-vsak-fld	any	≤1

Appendix C

Systemic ordering in Czech

The table below lists the kinds of dependency relation (tectogrammatical functions) represented as functors. The functors are ordered according to the systemic ordering in Czech, based on Petkevič (in press). The abbreviated functor labels used in Hajičová, Panevová, and Sgall (2000) are adopted as types representing the functors in the signature. In the following, Petkevič (in press) is referred to as Pet and Hajičová, Panevová, and Sgall (2000) as PDT.[1]

Key to the table

\# – the functor's order in the systemic ordering

Type – the type in the signature corresponding to the functor

Function – the standard label of the functor in FGD, boldface in case of inner participant

Gov. – governor: specification of wordclasses (parts of speech) modifiable by an item with this function: V – verb, N – noun, A – adjective/adverb

Several caveats are due:

1. Systemic ordering as a single list of functors which governs the TR order of non-bound dependents irrespective of their governor represents a rather strong hypothesis. The single list requirement has recently been relaxed in favour of a less restrictive notion of a general tendency with exceptions specific to individual lexemes or classes of lexemes.

[1]The list in Petkevič (in press) is itself based on the order of dependency relations as published in Sgall, Hajičová, and Panevová (1986, p. 198f). In the latter work, motivation and arguments for the specific choice and order of functors can also be found.

I am indebted to Vladimír Petkevič for providing me with the list in a machine-readable format. Indeed, it is me who is to blame for any errors, misinterpretations, or false conclusions.

Another possible issue concerns the possibility of partial instead of total order among functors.

2. The functors can be further specified by *c-grammatemes* (Pet), also called *syntactic grammatemes* (PDT). These are omitted, because they are not relevant for SO.

3. Although Pet's choice of functors is adhered to (PDT does not provide information on SO), a note in the table marks cases where PDT treats Pet's c-grammateme as a functor. This concerns the following PDT functors: #4: condition factual (or real) (*cond*) and counterfactual (*cterf*), #16: heredity (*her*), and #32: intent (*intt*).[2]

4. Also, the following PDT functors are omitted: complement, confrontation, part of phraseme, ethical dative, intensification, adverbial of modality, reference to preceding text, rhematizer, temporal: how often, from when, to when, vocative; and all functors for coordination, apposition and parenthesis: conjunction, disjunction, gradation, adversative, consequence, reason, apposition, parenthesis.

[2] According to (Petkevič, in press):

> The main criterion for characterizing a dependency relation *dep* as a c-grammateme within a complementation *c* rather than as an individual complementation is the possibility of coordinating the meaning of *dep* with the basic meaning of *c* and with *c*'s c-grammatemes.

#	Type	Functor	Gov.	Notes and Examples
1	*att*	Attitude	V	He is *probably* back home by now; *with pleasure*
2	*mat*	**Material**	N,A	(Partitive) expressing measured material or measured abstract notion – a bunch *of flowers*; full *of hope*; a glass *of wine*
3	*act*	**Actor**	V,N	*John* made it; *John* slept; he was caught *by the police*; *Rembrandt's* paintings; *a lady* said
4	*cond*	Condition	V	in PDT as two functions – factual (*cond*): I will not leave *if you do not give me money; under good skiing conditions* we shall go out; and counterfactual (*cterf*): *if I had known it*, I would not have gone there
5	*tsin*	Time: since when	V	*since his arrival*; *since the 15th century*
6	*twen*	Time: when	V	he came *yesterday*; she has been working *in the evenings*; *before sleep*
7	*thl*	Time: how long	V	it lasted *two hours*; he remained there *through the whole winter*
8	*ttill*	Time: till when	V	I was there *till Sunday*
9	*tfhl*	Time: for how long	V	she will stay *for two weeks*; *for his whole life*; *for a fortnight*
10	*tpar*	Time: contemporary	V	expresses the parallelism of actions – he was reading *during the journey*; *at the time of vital changes*
11	*caus*	Cause	V	he was smiling *because it was too ridiculous*; *for some reason*; she sacrificed herself *out of pure love*
12	*reg*	Regard	V	positive: *with regard to his health*; *in any case*; *as far as I am concerned* I wouldn't mind; negative: *regardless of what you are doing* I am always on your side
13	*resl*	Result	V	outcome, see 33 – tanned *brown*, they restored the house *into a stately home*
14	*cncs*	Concession	V	he failed *although he tried hard*
15	*aim*	Aim	V	(Purpose) she did it *in order to make a good impression*; *for his mother's sake*
16	*mann*	Manner	V	she was getting on *well*; he is nimble *as a weasel*; PDT distinguishes Heredity (*her*) as a separate function: he does it *after his father*
17	*ext*	Extent	V,A	she works *very hard*; he came *too* late
18	*norm*	Norm	V	in accordance with – kreslili *na pokyn* podle modelu 'they drew *on direction* after a model'

#	Type	Functor	Gov.	Notes and Examples
19	*crit*	Criterion	V	(Standard) he has done everything *according to the rules*
20	*subs*	Substitution	V	she was appointed chairman *instead of Kim*
21	*acmp*	Accompaniment	V	positive: he got on the bus *with a girl*; negative: Bill saw Kate jogging *without her boyfriend*
22	*restr*	Restriction	V	all were rescued *except for the two crew members*
23	*loc*	Locative	V	(Place) He was living *in Paris*; his cottage was built *on the hill*
24	*means*	Means	V	(Instrument) He was writing *with a pen*; he amused them *by telling jokes*
25	*diff*	Difference	V,A	with the comparative degree of comparison – He is *two inches* taller; she drove *twenty metres* from the parking lot
26	*dir1*	Direction: from where	V	he was creeping *out of the tent*
27	*dir2*	Direction: which way	V	through which place – she was running *through the bushes*
28	*addr*	**Addressee**	V	I brought the book *to Paul*; he gave *him* a pen; he taught *him* mathematics
29	*orig*	**Origin**	V,N	he made it *out of wood*; the letter is *from his father*
30	*pat*	**Patient**	V,N	(Objective, Goal) she knitted *a sweater* for her son; she thought *of that*; I saw *it*; he coped *with it*; she is *the boss*
31	*ben*	Benefactive	V	positive: he bought a bunch of flowers *for both Sue and Lucy*; negative: he was *against this idea*
32	*dir3*	Directional: where to	V	she penetrated deep *into the enemy's territory*; he jumped *onto the roof*; PDT distinguishes intent (*intt*) as a separate function: Mum sent Jane *to buy* apples
33	*eff*	**Effect**	V	(Result) she made the log *into a canoe*; they appointed him *vice-president*
34	*cpr*	Comparison	V,A	she is taller *than me*; You can't expect to play tennis *as* well *as me* at your age; she is *like her father*
35	*app*	Appurtenance	N	a leg *of the table*; *Paul's* brother
36	*rstr*	Restrictive	N	(General relationship) *three* glasses; *every* scholar; *black* cupboard; *some* things; the man *who did it*
37	*id*	**Identity**	N	The city *of London*; the notion *of God*
38	*des*	Descriptive property	N	non-restrictive *Golden* Prague; *sweet* France

Appendix D

Sources of examples

If not obvious from context, sources of borrowed examples are marked by abbreviations. A number in paranthesis following the abbreviation is the example number in the source text, a number preceded by colon is the page in the source text.

Abbreviation	Reference
AO95	Avgustinova and Oliva (1995)
BOBO	Ondra "Bobo" Novotný *Komár*, www.bobesh.cz/povidky
DAN	Daneš, Grepl, and Hlavsa (1987)
KAR	Karlík, Nekula, and Rusínová (1995)
KO	Karel Oliva, p.c.
KOCL	Oliva (1998)
MD	Magazín DNES
PS01	Sgall (2001)
SVO	Svoboda (2000)
SYN	Czech National Corpus, SYN2000 (the synchronous part of CNC)

References

Aho, Alfred V. and Jeffrey D. Ullman. 1973. *The Theory of Parsing, Translation and Compiling.* Prentice-Hall, Englewood Cliffs, NJ. I and II.

Alena Böhmová and Eva Hajičová. 1999. How Much of the Underlying Syntactic Structure Can Be Tagged Automatically. *The Prague Bulletin of Mathematical Linguistics*, (71):5–12.

Anna Kupść and Małgorzata Marciniak and Agnieszka Mykowiecka. 2000. Constituent Coordination in Polish: An Attempt at an HPSG Account. In Piotr Banski and Adam Przepiórkowski, editors, *Proceeding of the conference Generative Linguistics in Poland*, Warsaw, Poland. Polish Academy of Sciences.

Avgustinova, Tania and Karel Oliva. 1990. Syntactic Description of Free Word Order Languages. In *Proceedings of COLING 90*, volume 3, pages 311–313, Helsinki.

Avgustinova, Tania and Karel Oliva. 1995. The position of Sentential Clitics in the Czech Clause. In *Wiener Slavisticher Jahrbuch*, volume 41. Verlag der Österreichischen Akademie der Wissenschaften, Wien, pages 21–42. Quoted from the technical report CLAUS No. 68, Department of Computational Linguistics, University of Saarland, Saarbrücken, 1995.

Borsley, Robert D. and Adam Przepiórkowski, editors. 1999. *Slavic in HPSG.* Studies in constraint-based lexicalism. CSLI Publications, Stanford.

Bouma, Gosse, Geert-Jan M. Kruijff, and Richard T. Oehrle, editors. 1998. Universität des Saarlandes und DFKI.

Bouma, Gosse, Rob Malouf, and Ivan A. Sag. 1998. A unified theory of complement, adjunct, and subject extraction. In Gosse Bouma, Geert-Jan M. Kruijff, and Richard T. Oehrle, editors, *Proceedings of the FHCG98. Symposium on Unbounded Dependencies*, pages 83–97, Saarbrücken. Universität des Saarlandes und DFKI.

Bröker, Norbert, 1998. *How to define a context-free backbone for DGs: Implementing a DG in the LFG formalism*, pages 29–38. In Kahane and Polguère (Kahane and Polguère, 1998). Available at `http://www.fas.umontreal.ca/ling/olst/Eng/DGPapers.html`.

Carpenter, Bob. 1992. *The Logic of Typed Feature Structures with Applications to Unification-based Grammars, Logic Programming and Constraint Resolution*, volume 32 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, New York.

Chomsky, Noam. 1957. *Syntactic Structures.* Mouton, The Hague.

Colmerauer, Alain. 1970. Les Systémes Q ou un Formalisme pour Analyser at Synthétiser des Phrases sur Ordinateur. Internal publication 43, Université de Montréal.

Colmerauer, Alain. 1978. Metamorphosis grammars. In Leonard Bolc, editor, *Natural Language Communication with Computers.* Springer Verlag, Berlin. First appeared as "Les Grammaires de Metamorphose", Groupe d'Intelligence Artificielle, Université de Marseilles II, November 1975.

Copestake, Ann. 1999. The (new) LKB system. Technical report, Center for the Study of Language and Information (CSLI), Stanford University, Stanford. Available at http://www-csli.stanford.edu/~aac/lkb.html.

Copestake, Ann, Dan Flickinger, and Ivan A. Sag. 1997. Minimal Recursion Semantics: An introduction.

Crystal, David. 1997. *The Cambridge Encyclopedia of Language.* Cambridge University Press, second edition.

Dalrymple, Mary. 1999. Lexical-functional grammar. In Rob Wilson and Frank Keil, editors, *MIT Encyclopedia of the Cognitive Sciences.* The MIT Press.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell, and Annie Zaenen, editors. 1995. *Formal Issues in Lexical-Functional Grammar.* CSLI Publications, Stanford, CA.

Daneš, František. 1974. Functional sentence perspective and the organisation of the text. In František Daneš, editor, *Papers on functional sentence perspective.* Academia, Prague, pages 106–128.

Daneš, František, Miroslav Grepl, and Zdeněk Hlavsa, editors. 1987. *Mluvnice češtiny*, volume 3. Academia, Praha.

Davis, Anthony. 1997. *Lexical Semantics and Linking and the Hierarchical Lexicon.* Ph.D. thesis, Stanford University.

Davis, Anthony R. and Jean-Pierre Koenig. 1999. Linking as constraints on word classes in a hierarchical lexicon. In *Workshop on Lexical Semantics and Linking in Constraint-Based Theories, The 11th European Summer School in Logic, Language and Information*, Utrecht, August. Available at http://esslli.let.uu.nl/Courses/kordoni.html.

de Saussure, Ferdinand. 1916. *Cours de linguistique générale.* Paris. Publié par Ch. Bally et A. Sechehay avec la collaboration de A Riedlinger.

Dokulil, Miloš, Karel Horálek, Jiřina Hůrková, and Miloslava Knappová, editors. 1986. *Mluvnice češtiny*, volume 1. Academia, Praha.

Dušková, Libuše. 1986. A note on the thematic character of the subject in English in comparison with Czech. *Prague Studies in Mathematical Linguistics*, (9):95–102.

Elisabet Engdahl and Enric Vallduvi. 1994. Information packaging and grammar architecture: a constraint-based approach. Technical report, ILLC, Amsterdam. In DYANA-2 Report R1.3.B.

Esvan, François. 2000. Česká klitika z hlediska typologického. In Hladká and Karlík (Hladká and Karlík, 2000), pages 141–148. Proceedings of the conference at Šlapanice near Brno, 17-19 November 1999.

Firbas, Jan. 1957. Some Thoughts on the Function of Word Order in Old English and Modern English. In *Sborník prací filosofické fakulty brněnské university*, number A 5. University of Brno, pages 72–100.

Firbas, Jan. 1992. *Functional Sentence Perspective in Written and Spoken Communication.* Cambridge University Press, Cambridge, UK.

Flickinger, Daniel. 1987. *Lexical Rules in the Hierarchical Lexicon.* Ph.D. thesis, Stanford University.

Franks, Steven. 1998. Clitics in slavic. In *Position paper for the Workshop "Comparative Slavic Morphosyntax: The State of the Art".* (http://www.indiana.edu/ slavconf/linguistics/).

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized phrase structure grammar.* Harvard University Press, Cambridge, MA, USA.

Götz, Thilo and Walt Detmar Meurers. 1997. The ConTroll system as large grammar development platform. In *Proceedings of the Workshop "Computational Environments for Grammar Development and Linguistic Engineering (ENVGRAM)" held in conjunction with the 35th Annual Meeting of the ACL and 8th Conference of the EACL*, Madrid, Spain.

Grosz, Barbara, Karen Sparck Jones, and Bonnie Lynn Weber, editors. 1986. *Readings in Natural Language Processing.* Morgan Kaufmann, Los Altos, California.

Haegeman, Liliane. 1994. *Introduction to Government and Binding Theory.* Blackwell Textbooks in Linguistics. Blackwell, second edition.

Hajič, Jan. 1998. The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning – Studies in Honour of Jarmila Panevová.* Karolinum, Charles University Press, pages 106–132.

Hajič, Jan, Eva Hajičová, Jarmila Panevová, and Petr Sgall. 1998. Syntax v Českém národním korpusu. *Slovo a slovesnost*, 59(3):168–177.

Hajič, Jan, Eva Hajičová, and Alexandr Rosen. 1992. Machine Translation Research in Czechoslovakia. *META*, 37(4):802–816. A special issue, eds. M. C. Cormier and D. Estival.

Hajičová, Eva. 1998. Movement rules revisited. In Kahane and Polguère (Kahane and Polguère, 1998), pages 49–57. Available at `http://www.fas.umontreal.ca/ling/olst/Eng/DGPapers.html`.

Hajičová, Eva and Jarmila Panevová. 1984. Valency (Case) Frames of Verbs. In Petr Sgall, editor, *Contributions to Functional Syntax, Semantics, and Language Comprehension*. Academia, Praha, pages 147–188.

Hajičová, Eva, Jarmila Panevová, and Petr Sgall. 2000. A manual for tectogrammatical tagging of the prague dependency treebank. ÚFAL/CKL Technical Report TR-2000-09, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Hajičová, Eva, Barbara H. Partee, and Petr Sgall. 1998. *Topic-Focus Articulation, Tripartite Structures, and Semantic Content*, volume 71 of *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Hajičová, Eva and Petr Sgall. 1999. Quantifiers and focus in an underspecified deep structure. *The Prague Bulletin of Mathematical Linguistics*, (72):5–9.

Hajičová, Eva Eva. 1993. *Issues of sentence structure and discourse patterns*, volume 2 of *Theoretical and computational linguistics*. Charles University, Prague.

Hellwig, Peter. 1986. Dependency Unification Grammar. In *Proc. of the 11th COLING*, pages 195–198, Bonn, Germany.

Hellwig, Peter. 1998. Processing of Dependency-based Grammars in the light of Dependency Unification Grammar (DUG). Available at `http://www.gs.uni-heidelberg.de/~hellwig/biblio.html`.

Hinrichs, Erhard W. and Tsuneko Nakazawa. 1994. Linearizing AUXs in German Verbal Complexes. In John Nerbonne, Klaus Netter, and Carl Pollard, editors, *German in Head-Driven Phrase Structure Grammar*, number 46 in CSLI Lecture Notes. CSLI Publications, Stanford, USA, pages 11–38.

Hinrichs, Erhard W. and Tsuneko Nakazawa. 1999. VP Relatives in German. In Kordoni (Kordoni, 1999), pages 1–21.

Hladká, Zdeňka and Petr Karlík, editors. 2000. Number 2, Brno. Masarykova univerzita. Proceedings of the conference at Šlapanice near Brno, 17-19 November 1999.

Hladká, Zdeňka and Petr Karlík, editors. 2001. Number 3, Brno. Masarykova univerzita. Proceedings of the conference at Šlapanice near Brno, 22-24 November 1999.

Holan, Tomáš, Vladislav Kuboň, Karel Oliva, and Martin Plátek, 1998. *Two Useful Measures of Word Order Complexity*, pages 21–28. In Kahane and Polguère (Kahane and Polguère, 1998). Available at `http://www.fas.umontreal.ca/ling/olst/Eng/DGPapers.html`.

Holan, Tomáš, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 2000. On complexity of word order. ÚFAL/CKL Technical Report TR-2000-08, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Hornstein, N. 1999. Movement and control. *Linguistic Inquiry*, 30(1):69–96.

Hudson, Richard. 1990. *English Word Grammar*. Blackwell.

Hudson, Richard. 1998. An Encyclopedia of English Grammar and Word Grammar. Available at `http://www.phon.ucl.ac.uk/home/dick/enc-gen.htm`.

Jacobs, J. 1907. *English Fairy-Tales*. London.

Jakobson, Roman. 1990. *On Language*. Harvard University Press, Cambridge, MA, USA. Edited by Linda R. Waugh and Monique Monville-Burston.

Järvinen, Timo and Pasi Tapanainen. 1998. Towards an implementable dependency grammar. In Kahane and Polguère (Kahane and Polguère, 1998), pages 1–10. Available at `http://www.fas.umontreal.ca/ling/olst/Eng/DGPapers.html`.

Johnson, David and Shalom Lappin. 1997. A Critique of the Minimalist Program. *Linguistics and Philosophy*, 20:272–333.

Johnson, Mark. 1988. *Attribute-Value Logic and the Theory of Grammar*. Number 16 in CSLI Lecture Notes. CSLI Publications, Stanford, California, USA.

Kahane, Sylvaine and Alaine Polguère, editors. 1998. *Processing of Dependency-based Grammars: Proceedings of the Workshop, COLING-ACL'98*. Université de Montréal. Available at `http://www.fas.umontreal.ca/ling/olst/Eng/DGPapers.html`.

Kaplan, Ronald D. and Joan Bresnan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, MA, pages 173–281. Reprinted in Dalrymple et al. (1995, 29-130).

Karlík, Petr, Marek Nekula, and Zdenka Rusínová, editors. 1995. *Příruční mluvnice češtiny*. Nakladatelství Lidové noviny, Praha.

Karlsson, Fred, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*, volume 4 of *Natural Language Processing*. Mouton de Gruyter, Berlin and New York.

Kathol, Andreas. 1995. Linearization-based german syntax. Dissertation, submitted to the Ohio State University.

Kay, Martin. 1983. Unification Grammar. Technical report, Xerox Palo Alto Research Center, Palo Alto, California.

Kay, Paul. 1998. An Informal Sketch of a Formal Architecture for Construction Grammar. In Bouma et al. (Bouma, Kruijff, and Oehrle, 1998), pages 175–184.

King, Paul John. 1989. *A logical formalism for head-driven phrase-structure grammar*. Ph.D. thesis, Manchester University.

King, Paul John. 1999. Towards Truth in HPSG. In Kordoni (Kordoni, 1999), pages 301–352.

Kirschner, Zdeněk. 1982. *A Dependency-Based Analysis of English for the Purpose of Machine Translation*, volume IX of *Explizite Beschreibung der Sprache und automatische Textbearbeitung*. Charles University, Faculty of Mathematics and Physics, Prague.

Kirschner, Zdeněk and Alexandr Rosen. 1989. APAČ – An Experiment in Machine Translation. *Machine Translation*, 4(3):177–193.

Koktová, Eva. 1999. *Word-order Based Grammar*. Mouton de Gruyter, Berlin/New York.

Kordoni, Valia, editor. 1999. *Tübingen Studies in Head-Driven Phrase Structure Grammar*. Arbeitspapiere des SFB 340, Bericht Nr. 132. University of Tübingen, Tübingen, Germany.

Kruijff, Geert-Jan M. 1999. Diamonds for Dependencies. In E. Thijsse and H. Bunt, editors, *Proceedings of the Third International Workshop on Computational Linguistics*, pages 351–366, Tilburg, The Netherlands, January.

Kruijff-Korbayová, Ivana. 1998. *The Dynamic Potential of Topic and Focus: A Praguian Approach to Discourse Representation Theory.* Ph.D. thesis, Department of Mathematics and Physics, Charles University, Praha, Czech Republic.

Kuboň, Vladislav and Martin Plátek. 1993. Robust parsing and grammar checking of free word order languages. In *Proceedings of the 6th Twente Workshop on Language Technology*, pages 157–161, Enschede. Universiteit Twente.

Kupść, Anna. 1999. Haplology of the Polish Reflexive Marker. In Borsley and Przepiórkowski (Borsley and Przepiórkowski, 1999), pages 91–125.

Kupść, Anna. 2000. An hpsg grammar of polish clitics. Dissertation, submitted to the Institute of computer Science at the Polish Acadamy of Sciences, and to the Université Paris 7.

Kuthy, Kordula De and Walt Detmar Meurers. 1999. On Partial Constituent Fronting in German. In Kordoni (Kordoni, 1999), pages 22–73.

Kučerová, Jana. 1995. The extension of the condition of projectivity for structures with coordination, Part I and II. *The Prague Bulletin of Mathematical Linguistics*, (63, 65-66):67–83, 67–91.

L. Bolc and K. Czuba and A. Kupść. 1996. A Survey of Systems for Implementing HPSG Grammars. Technical Report 814, Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland.

Lehmann, S. and S. Oepen. 1996. TSNLP — Test Suites for Natural Language Processing. In *Proceedings of COLING 96*, Copenhagen, Denmark.

Manning, Christopher P. and Ivan A. Sag. 1999. Dissociations between Argument Structure and grammatical relations. In Webelhuth et al. (Webelhuth, Koenig, and Kathol, 1999), pages 63–98.

Manning, Christopher P., Ivan A. Sag, and M. Iida. 1997. The lexical integrity of Japanese causatives. version of March 5, 1997.

Mathesius, Vilém. 1939. O tak zvaném aktuálním členění v ětném. *Slovo a slovesnost*, (5):171–174. Translated as Mathesius (1975).

Mathesius, Vilém. 1942a. *Řeč a sloh [Speech and style]*, volume 1 of *Čtení o jazyce a poezii [Readings about language and poetry]*. Družstevní práce, Prague. reprinted in Mathesius (1982, 92-146).

Mathesius, Vilém. 1942b. Ze srovnávacích studií slovosledných [From comparative word-order studies]. *Časopis pro moderní filologii*, (28):181–190,302–307.

Mathesius, Vilém. 1975. On Information-Bearing Structure of the Sentence. In Susumu Kuno, editor, *Harvard Studies in Syntax and Semantics*. pages 467–480.

Mathesius, Vilém. 1982. *Jazyk, kultura a slovesnost [Language, culture and literature]*. Odeon, Prague.

Mel'čuk, I. A. and A. K. Žolkovskij. 1970. Towards a functioning meaning–text model of language. *Linguistics*, (57):10–47.

Meurers, Detmar. 1995. Towards a semantics for Lexical rules as used in HPSG. In *Proceedings of ESSLLI7 Conference on Formal Grammar*, Barcelona, Spain. Universidad Politecnita de Catalunya.

Moshier, Drew. 1988. Extensions to unification grammars for the description of programming languages. Master's thesis, University of Michigan, Ann Harbour. Doctoral dissertation.

Nerbonne, John, Klaus Netter, and Carl Pollard, editors. 1994. *German in Head-Driven Phrase Structure Grammar*. Number 46 in CSLI Lecture Notes. CSLI Publications, Stanford, USA.

Netter, Klaus. 1994. Towards a Theory of Functional Heads: German Nominal Phrases. In Nerbonne et al. (Nerbonne, Netter, and Pollard, 1994), pages 297–340.

Oliva, Karel. 1989. *A Parser for Czech Implemented in Systems Q*, volume XVI of *Explizite Beschreibung der Sprache und automatische Textbearbeitung*. Charles University, Faculty of Mathematics and Physics, Prague.

Oliva, Karel. 1992. The Proper Treatment of Word Order in HPSG. In *Proceedings of COLING 92*, pages 184–190, Nantes.

Oliva, Karel. 1994. HPSG Lexicon without Lexical Rules. In *Proceedings of COLING 94*, Kyoto.

Oliva, Karel. 1998. A Closer Look on the Czech Data in the "Position Paper: Clitics in Slavic". response to (Franks, 1998).

Oliva, Karel. 2001. Některé aspekty komplexity českého slovního nepořádku. In Hladká and Karlík (Hladká and Karlík, 2001), pages 163–172. Proceedings of the conference at Šlapanice near Brno, 22-24 November 1999.

Oliva, Karel D. and Vladimír Petkevič. 1998. Phenomena-Based Description of Dependency Syntax: A Survey of Ideas and Formalization. In Eva Hajičová, editor, *Issues of Valency and Meaning – Studies in Honour of Jarmila Panevová*. Karolinum, Charles University Press, Praha, pages 133–153.

Panevová, Jarmila. 1979. *Transducing Components of Functional Generative Description 1: From Tectogrammatics to Morphemics*, volume IV of *Explizite Beschreibung der Sprache und automatische Textbearbeitung.* Charles University, Faculty of Mathematics and Physics, Prague.

Panevová, Jarmila. 1980. *Formy a funkce ve stavbě české věty.* Academia, Praha.

Panevová, Jarmila. 1998. Ještě k teorii valence. *Slovo a slovesnost*, 59(1):1–14.

Penn, Gerald. 1993. A utility for typed feature structure-based grammatical theories. Master's thesis, Carnegie Mellon University, Pittsburgh.

Penn, Gerald. 1999a. Linearization and WH-extraction in HPSG: Evidence from Serbo-Croation. In Borsley and Przepiórkowski (Borsley and Przepiórkowski, 1999), pages 149–182.

Penn, Gerald. 1999b. An RSRL Formalization of Serbo-Croation Second Position Clitic Placement. In Kordoni (Kordoni, 1999), pages 177–197.

Peregrin, Jaroslav. 1995. Topic, focus and the logic of language. In *Proceedings of the Göttingen Focus Workshop, 17. DGfS*, Heidelberg. IBM Deutschland. http://www.cuni.cz/p̃eregrin/HTMLTxt/tflog.htm.

Pereira, Fernando and David Warren. 1980. Definite clause grammars for language analysis – a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13:231–278. Reprinted in Grosz, Jones, and Weber (1986).

Petkevič, Vladimír. 1987. A new dependency based specification of underlying representations of sentences. *Theoretical Linguistics*, 14(2/3):143–172.

Petkevič, Vladimír. 1993. Underlying Structures and Unification. *The Prague Bulletin of Mathematical Linguistics*, (59-60):23–44.

Petkevič, Vladimír. 1995a. A new formal specification of underlying structures. *Theoretical Linguistics*, 21:7–61.

Petkevič, Vladimír. 1995b. Formal account of the topic and focus articulation within dependency syntax. In Bohumil Palek, editor, *Proceedings of LP'94*, Prague. Charles University Press.

Petkevič, Vladimír. in press. Underlying structure of sentence based on dependency.

Plátek, Martin, Jiří Sgall, and Petr Sgall. 1984. A Dependency Base for a Linguistic Description. In Petr Sgall, editor, *Contributions to Functional Syntax, Semantics, and Language Comprehension*. Academia, Praha, pages 63–97.

Pollard, Carl. 1996. The Nature of Constraint-Based Grammar. In *Proceedings of Pacific Asia Conference on Language, Information, and Computation*, Seoul, Korea, December. Kyung Hee University.

Pollard, Carl. 1999. Strong Generative Capacity in HPSG. In Webelhuth et al. (Webelhuth, Koenig, and Kathol, 1999), pages 281–298.

Pollard, Carl and Eun Jung Yoo. 1998. A unified theory of scope for quantifiers and *wh*-phrases. *Journal of Linguistics*, (34):415–445.

Pollard, Carl J. and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics*. Number 13 in CSLI Lecture Notes. CSLI, Stanford, California, USA.

Pollard, Carl J. and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Preinhaelterová, Lída. 1999. The Role of Sentence Stress in Topic-Focus Articulation in Czech Compared to the Situation in English and Japanese. *The Prague Bulletin of Mathematical Linguistics*, (71):13–34.

Przepiórkowski, Adam. 1998. 'A Unified Theory of Scope' revisited: Quantifier retrieval without spurious ambiguities. In Bouma et al. (Bouma, Kruijff, and Oehrle, 1998), pages 185–195.

Przepiórkowski, Adam. 1999a. Case assignment and the complement/adjunct dichotomy. Dissertation, submitted to the Neuphilologische Fakultät of the Universität Tübingen.

Przepiórkowski, Adam. 1999b. On Complements and Adjuncts in Polish. In Robert D. Borsley and Adam Przepiórkowski, editors, *Slavic in HPSG*, Studies in constraint-based lexicalism. CSLI Publications, Stanford, pages 183–210.

Przepiórkowski, Adam. 2000. Formal Grammar (HPSG). In *Proceedings of SLING2K Worshop*. To appear. Electronic version available at `http://www.ling.ohio-state.edu/ãdamp/Papers/2000-sling2k/`.

Pullum, Geoffrey. 1989. Formal Linguistics Meets the Boojum. *Natural Language and Linguistic Theory*, (7).

Quirk, Randolph, Sidney Greenbaum, Geoffrey N. Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language.* Longman, London.

Reape, Mike. 1994. Domain Union and Word Order Variation in German. In Nerbonne et al. (Nerbonne, Netter, and Pollard, 1994), pages 151–198.

Richter, Frank. 1999. RSRL for HPSG. In Kordoni (Kordoni, 1999), pages 74–115.

Richter, Frank. 2000. A mathematical formalism for linguistic theories with an application in head-driven phrase structure grammar. Dissertation, submitted to the Neuphilologische Fakultät of the Universität Tübingen, Version of April 28th, 2000.

Richter, Frank, Manfred Sailer, and Gerald Penn. 1999. A Formal Interpretation of Relations and Quantification in HPSG. In Kordoni (Kordoni, 1999), pages 155–176.

Riehemann, Sabine. 1993. Word formation in lexical type hierarchies: A case study of bar-adjectives in german. Master's thesis, Univesität Tübingen.

Rosen, Alexandr. 1994. Grammar Formalisms and the Description of Word Order Variations. A gzipped postscript file (41 pages) available at `http://utkl.ff.cuni.cz/~rosen/public/worep.ps.gz`.

Rosen, Alexandr. 1996. Lexical Issues in Machine Translation. Final report on project 340/93, Open Society Institute, Research Support Scheme, Praha. A gzipped postscript file (58 pages) available at `http://utkl.ff.cuni.cz/~rosen/public/rssfrep.ps.gz`.

Rosen, Alexandr. 1998. Review of Fred Karlsson et al. (eds.) *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*, Mouton de Gruyter 1995. *The Prague Bulletin of Mathematical Linguistics*, (69):57–63.

Rosen, Alexandr, Eva Hajičová, and Jan Hajič. 1992. Derivation of Underlying Valency Frames from a Learner's Dictionary. In *Proceedings of COLING-92 Nantes*, volume II, pages 553–559.

Sgall, Petr. 1967. *Generativní popis jazyka a česká deklinace [Generative Description of Language and the Czech Declension].* Academia, Praha.

Sgall, Petr. 1992. Underlying structure of sentences and its relations to semantics. In Tilmann Reuther, editor, *Festschrift für Viktor Jul'evič Rozenberg.* Wien, pages 273–282. Wiener Slawistischer Almanach, Sonderband 33.

Sgall, Petr. 1993. The Role of Lexical Data in a Dependency-Based Descriprion. In Frank Beckmann und Gerhard Heyer, editor, *Theorie und Praxis des Lexicons*. Walter de Gruyter, Berlin - New York, pages 43–53.

Sgall, Petr. 1995. Formal and Computational Linguistics in Prague. In Eva Hajičová, Miroslav Červenka, Oldřich Leška, and Petr Sgall, editors, *Prague Linguistic Circle Papers*, volume 1. John Benjamins, Amsterdam/Philadelphia, pages 23–35.

Sgall, Petr. 1997. On the usefullness of movement rules. In *Proceedings of the 16th International Congress of Linguists*, Oxford. Pergamon. Paper No. 0503.

Sgall, Petr. 1998. Teorie valence a její formální zpracování [valency theory and its formalization]. *Slovo a slovesnost*, 59(1):15–29.

Sgall, Petr. 2001. Volnost jako univerzální vlastnost jazyka. In Hladká and Karlík (Hladká and Karlík, 2001), pages 49–58. Proceedings of the conference at Šlapanice near Brno, 22-24 November 1999.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel and Academia, Dordrecht and Praha. Editor: Jacob Mey.

Sgall, Petr, O. E. Pfeiffer, W. U. Dressler, and M. Půček. 1995. Experimental research on systemic ordering. *Theoretical Linguistics*, 21(2/3):197–239.

Shieber, Stuart M., Hans Uszkoreit, Fernando Pereira, J. J. Robinson, and M. Tyson. 1983. The Formalism and Implementation of PATR-II. In *Research on Interactive Acquisition and Use of Knowledge*. Artificial Intelligence Center, SRI International, Menlo Park, California, pages 43–53.

Skoumalová, Hana. 2001. *Czech hierarchical lexicon*. Ph.D. thesis, Charles University.

Sleator, Daniel and Davy Temperley. 1993. Parsing English with a Link Grammar. In *Third International Workshop on Parsing Technologies*. Available at `http://bobo.link.cs.cmu.edu/grammar/html/papers/`.

Smolka, Gert. 1988. A feature logic with subsorts. LILOG technical report 33, IBM Deutschland GmbH, Stuttgart, Germany.

Starosta, Stanley. 1988. *The case for lexicase: an outline of lexicase grammatical theory*. Cassell, London.

Starosta, Stanley. 1993. Word order and focus in constrained dependency grammar. In Eva Hajičová, editor, *Proceedings of the conference "Functional description of language"*, pages 237–252, Prague. Quoted from an extended electronic version of the paper available from Stanley Starosta's home page.

Svoboda, Aleš. 2000. Klitika z hlediska funkční větné perspektivy I. In Hladká and Karlík (Hladká and Karlík, 2000), pages 149–159. Proceedings of the conference at Šlapanice near Brno, 17-19 November 1999.

Svoboda, Aleš. 2001. Klitika z hlediska funkční větné perspektivy II. In Hladká and Karlík (Hladká and Karlík, 2001), pages 81–86. Proceedings of the conference at Šlapanice near Brno, 22-24 November 1999.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Editions Klincksieck, Paris.

Toman, Jindřich. 2000. Prosodické spekulace o klitikách v nekanonických pozicích. In Hladká and Karlík (Hladká and Karlík, 2000), pages 161–166. Proceedings of the conference at Šlapanice near Brno, 17-19 November 1999.

Toman, Jindřich. 2001. Ertlova diskuse českých klitik. In Hladká and Karlík (Hladká and Karlík, 2001), pages 73–79. Proceedings of the conference at Šlapanice near Brno, 22-24 November 1999.

Trask, Robert Lawrence. 1993. *A Dictionary of Grammatical Terms in Linguistics*. Routledge, London and New York.

Uhlířová, Ludmila. 1972. On the quantitative analysis of clause and utterance in Czech. *Prague Studies in Mathematical Linguistics*, (4):107–128.

Uhlířová, Ludmila. 1976. Optional constituents in Theme-Rheme structure. *Prague Studies in Mathematical Linguistics*, (5):309–319.

Uhlířová, Ludmila. 2001. Klitika jako kategorie diskusní. In Hladká and Karlík (Hladká and Karlík, 2001), pages 29–35. Proceedings of the conference at Šlapanice near Brno, 22-24 November 1999.

Uszkoreit, Hans. 1986. Categorial Unification Grammar. In *Proceedings of COLING 86*, Bonn. University of Bonn.

Uszkoreit, Hans. 1996. On the essence of HPSG and its role in computational linguistics. `http://www.coli.uni-sb.de/ñansu/hpsg1.html`.

Čermák, František. 1997. *Jazyk a jazykověda [Language and Linguistics]*. Pražská imaginace, Nechvílova 1832, Praha 4, second edition.

Šmilauer, Vladimír. 1966. *Novočeská skladba*. Státní pedagogické nakladatelství, Praha, 3 edition.

Webelhuth, Gert, Jean-Pierre Koenig, and Andreas Kathol, editors. 1999. *Lexical and Constructional Aspects of Linguistic Explanation*. CSLI Publications, Stanford, CA.

Woods, William A. 1970. Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606.

,