Saarland University
Faculty of Humanities II
Department of Computational Linguistics and Phonetics

Mapping the Prague Dependency Treebank Annotation Scheme onto Robust Minimal Recursion Semantics

Master's Thesis

Max Jakob

Master of Science in Language Science and Technology

2009

I hereby declare that this master's thesis is my own work and where it draws on the work of others it is properly cited in the text.
I agree with a public availability of the work.

Saarbrücken, 14th of December, 2009

Max Jakob

Title	Mapping the Prague Dependency Treebank Annotation Scheme onto Robust Minimal Recursion Semantics
Author	Max Jakob
Institution	Department of Computational Linguistics and Phonetics at Saarland University
Study Program	M.Sc. in Language Science and Technology
Supervisor	PD Dr. Valia Kordoni
1. Evaluator	PD Dr. Valia Kordoni
2. Evaluator	Prof. Dr. Hans Uszkoreit

Abstract

This thesis investigates the correspondence between two semantic formalisms, namely the tectogrammatical layer of the Prague Dependency Treebank 2.0 (PDT) and Robust Minimal Recursion Semantics (RMRS). It is a first attempt to relate the dependency based annotation scheme of PDT to a compositional semantics approach like RMRS.

An iterative mapping algorithm that converts PDT trees into RMRS structures is developed that associates RMRSs to each node in the dependency tree. Therefore, composition rules are formulated and the complex relation between dependency in PDT and semantic heads in RMRS is analyzed in detail. It turns out that structure and dependencies, morphological categories and some coreferences can be preserved in the target structures. Furthermore, valency and free modifications are distinguished using the valency dictionary of PDT as an additional resource.

The evaluation result of 81% recall shows that systematically correct underspecified target structures can be obtained by a rule-based mapping approach, which is an indicator that RMRS is capable of representing Czech data. This finding is novel as Czech, with its free word order and rich morphology, is typologically different from language that used RMRS thus far.

Key words: Semantics, Prague Dependency Treebank,
Minimal Recursion Semantics, Language Resources

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Manually annotated linguistic corpora are highly valuable for academic research as well as for applications. They provide reliable resources to evaluate various kinds of approaches and hypotheses in natural language processing and therefore constitute the foundation for empirical corpus linguistics. Moreover, they build the basis for statistical methods as training, development and test data. Unfortunately, these resources are expensive, especially for deep linguistic processing. Annotators need profound insights into the underlying structures of complex linguistic phenomena and hence must not lack an education in linguistics. Furthermore, there is a big variety of annotation schemes with different theoretical backgrounds that put emphasis on various aspects of natural language. For example, Slavic linguistics traditionally uses dependency grammars because free order languages are naturally easier to describe in this manner, while on the other hand, for English, phrase-structure grammars have been developed for an extended period of time. This difference in description becomes a barrier, making cross-fertilization of systems and resources using different formalisms very difficult. To overcome these differences, the relation between various formalisms has been examined in the past. For instance, the relation between constituency trees and dependency trees has been well defined ([Robinson, 1970]) to the extend that the conversion of descriptions on the syntactic level, i.e. phrase-structure and dependency grammars, is feasible, given certain properties. This enables followers of both orientations to potentially benefit from annotation efforts in either description.

In this thesis, two semantic resources are focused on and are being related to each other: the Prague Dependency Treebank 2.0 (PDT) annotation scheme and Robust Minimal Recursion Semantics (RMRS). The latter is a variant of the underspecification formalism Minimal Recursion Semantics (MRS). A dependency based formalism is therefore related to a compositional semantics approach. PDT annotates Czech texts on different layers, with the layer of highest abstraction incorporating meaning as well as some topic-focus information and coreferences. It uses dependency trees in which complex nodes representing lexical units are related to one another and has a sound theoretical background. MRS, on the other hand, is not a semantic theory but rather a practical way of composing a set of predicate logic formulas by allowing for scope relations to be underspecified. Thereby, it increases computational tractability and efficiency without compromising the expressibility of the underlying object language. It has been used as semantic representation in a big variety of systems and grammars for several years, especially for typed feature structure grammars.

The goal of this project is to develop an algorithm that converts PDT trees of the tectogrammatical layer to RMRS structures while trying to keep as much information of the source representation as possible. Although these formalisms adopt different frameworks, on the higher levels of abstraction, there is common ground that makes a conversion possible. For instance, valency plays a core role in the description of relations between meaning bearing units in both formalisms. However, the classical MRS descriptions have to be slightly altered in order to account for the typological difference of Czech in comparison with languages that RMRS was used for so far. Furthermore, the composition rules for constructing complete MRSs from a PDT tree have to be defined. This also involves reformulating the concept of dependency in PDT in terms of the target formalism.

The main benefit of such a mapping algorithm is that it makes the data of the source formalism available to a bigger community of researchers in the field of natural language processing. As a consequence, compositional semantic descriptions could be enriched with information from resources formulated using dependency trees. This bears potential improvements for several areas of deep linguistic processing, such as question answering or machine translation. Moreover, this endeavor is novel in that it explores the capability of the target formalism to represent typologically different languages, in this case, a free word order language with a rich morphology, like Czech. However, at this point it remains an open question how much information can be preserved.

1.1 Related Work

Besides MRS, there exist other underspecification formalisms, like the Constraint Language for Lambda Structures (CLLS, [Egg *et al.*, 2001]) and Hole Semantics ([Bos, 1995]), most of them being inter-convertible or at least convertible to a common structure ([Koller *et al.*, 2003, Fuchss *et al.*, 2004]). Nevertheless, MRS is the most widely used one and making resources available in this format therefore yields the biggest advantage. A broad range of systems has been implemented utilizing it.

The most prominent use of MRS descriptions is in the English Resource Grammar (ERG, [Copestake and Flickinger, 2000, ERG, 2009]). It is a large-scale head-driven phrase structure grammar for English which computes underspecified MRSs for semantic representation of natural language in open-domain applications. The ERG could profit from an exactly defined relation of MRS to dependency schemes in that its outputs could be enriched with dependency information from other resources. Furthermore, [Dridan and Bond, 2006] use a variant of MRS as an abstract representation for sentence comparison of Japanese data. Their approach can be exploited for answer sentence selection in question answering.

In machine translation, MRS was used in many systems since the Verbmobil project ([Bos *et al.*, 1996, Copestake *et al.*, 1995]). This area might profit the most when the Czech language data of PDT can be used by mature translation systems using MRS. The approach in [Žabokrtský *et al.*, 2008] takes a reversed perspective as they try to analyze English sentences to be represented in the dependency scheme of PDT. The capability of this scheme to capture a fixed order language was therefore already investigated. The opposite direction is investigated in this thesis.

Considering the conversion between structures, [Allen *et al.*, 2007] describe a mapping of generic logical forms in frame-like notation onto MRS structures in a deep processing approach for spoken dialogue systems. In [Kruijff, 2001], on the other hand, an approach that relates the theoretical background of PDT to categorial-modal logical descriptions using predicate-valency structures, dependency relations and aspectual categories is developed.

This thesis is organized in the following way. In chapter 2, background information about the involved resources is given, first for PDT and afterwards for MRS and its variant RMRS. In chapter 3, the correspondence of the PDT and the RMRS frameworks is examined. The relation between a linguistic theory and its formalism in the context of this project is clarified and the properties of the produced RMRS structures is outlined. Most importantly, this chapter describes the semantic composition rules for constructing RMRS representations for a complete PDT tree. This involves a special relation between nodes in the tree, which characterizes the main differences between the dependency concepts used in the two formalisms. The concrete algorithm of mapping PDT trees onto RMRS structures is also shown in this chapter. Chapter 4 evaluates the produced representations using certain structural properties. The thesis concludes in chapter 5 with a summarization and some suggestions for future work.

Chapter 2

Background

This chapter introduces the most important background knowledge necessary for understanding this thesis. First, the annotation of the Prague Dependency Treebank 2.0 is outlined in section 2.1. The tectogrammatical layer is discussed in more detail. It is the source representation for the mapping shown in the next chapter. Section 2.2 describes the basic ideas of Minimal Recursion Semantics as semantic representation and also presents a specific variant called Robust Minimal Recursion Semantics. The latter will be the target representations of the mapping. Furthermore, a special graph notation is introduced that will later assist in the evaluation of the structures produced in the mapping.

2.1 Prague Dependency Treebank

The Prague Dependency Treebank 2.0[1] (PDT, [Hajič *et al.*, 2006b]) is an annotated corpus of Czech-language data developed at the Institute of Formal and Applied Linguistics at Charles University in Prague. Its linguistically rich annotation ranges from morphology through syntax to meaning. It is based on the long-standing linguistic tradition of Prague and was adapted for the current computational linguistics research needs ([Hajič *et al.*, 2001], [Hajič, 2006]). The texts were taken from a selection of newspaper and magazine articles of the Czech National Corpus[2].

[1] http://ufal.mff.cuni.cz/pdt2.0/

[2] http://ucnk.ff.cuni.cz/

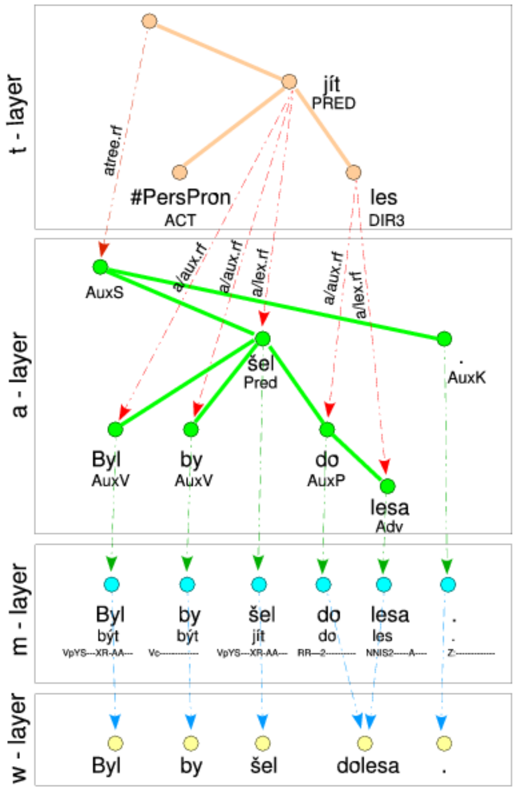


Figure 2.1: Layers of annotation in PDT for the example sentence ”*Byl by šel do lesa.*” (engl. ”*He would have gone into the woods.*”) (taken from [Hajič *et al.*, 2006a]).

2.1.1 Stratificational Annotation

For the annotation of the PDT data, the stratificational approach based on the Functional Generative Description (FGD, [Sgall *et al.*, 1986]) theory was adapted. Three annotation layers are distinguished. Each of them contains enough information to re-generate the original sentence string (or a synonymous one). Furthermore, there are explicit links between the elements of the different layers. They describe the generative relation of the layers from top to bottom. Figure 2.1 shows all linked layers of annotation and the layer with the original sentence string for an example sentence. Note that there are some differences between the theory and the actual corpus annotation due to several reasons. First, concrete implementation compromises had to be made. Second, the annotators have to work in the opposite (analytical) direction than the theory suggests, i.e. from a string of words to its meaning representation. Finally, the annotation efforts are limited by constraints of funding.

The layer of maximal abstraction is the tectogrammatical layer, annotating sentence meaning via dependencies and functions, topic-focus articulation, coreferences and meaning of morphological categories. The information of this layer will be the input for the mapping developed in this thesis[3]. Later chapters will specify which parts of the tectogrammatical information will be mapped and which parts will be left to future research. The two lower layers, the morphological layer and the analytical layer, will not be used in the mapping, as the tectogrammatical layer comprises all necessary information. Nevertheless, they will be briefly outlined here.

Morphological Layer

The morphological layer annotates all tokens in the sentence with a single morphological lemma. It can be viewed as a disambiguated reference to a dictionary entry. Additionally, the tokens are tagged with their part-of-speech tag. For this, a positional tag system is used including 13 different categories (e.g., POS, gender, number, tense, voice, etc.). Furthermore, sentence boundaries are marked.

Analytical Layer

The analytical layer describes the surface syntactic structure of sentences as dependency trees. They are directed, connected, acyclic graphs with a single root node, where each node (with the exception of the technical root node) has exactly one governing node. Also, the nodes are complex in the sense that they have attribute-value matrices associated with them. There is a one-to-one relation of the analytical nodes to the tokens on the morphological layer. This means that the number of nodes in the analytical tree is equal to the number of input tokens, plus one more for the technical root node. The edges of the dependency tree mark surface syntactic relations, so called analytical functions, between the nodes representing the input words. Moreover, the

[3] For details on the data format see [Pajas and Štěpánek, 2005].

order of the words in the sentence is preserved in analytical trees.

The tree edges represent mainly dependency relations, i.e. relations between governing (modified) and dependent (modifying) words. They are constructed by the following general principle in a linear ordering (left to right): the deletion of a dependent node does not harm the grammaticality of the sentence ([Sgall *et al.*, 1986]). This principle is complemented by some conventions, e.g. that prepositions govern nouns and subordinate conjunctions govern auxiliary verbs. All tree edges are marked with analytical functions that describe the type of relation. The analytical functions for dependency relations are *predicate*, *subject*, *object*, *adverbial*, *attribute* and *complement*. There are also other, non-dependency, analytical functions represented as edges, coordination being the most important. The information about the type of analytical function from one node to another is annotated in an attribute-value matrix associated with the dependent node.

2.1.2 Tectogrammatical Layer

"The aim of the tectogrammatical layer is to go beyond the surface shape of the sentence with such notions as *subject* and *object*, and to employ notions like *actor*, *patient*, *addressee* etc., while still being mostly driven by the language structure itself rather than by the general world knowledge" ([Hajič *et al.*, 2001], page 3). Again, the structure is represented as a dependency tree with complex nodes (with associated attribute-value matrices). The nodes of the tectogrammatical layer do not correspond to the previous layer in a one-to-one relation. Only the nodes that carry lexical meaning are represented, i.e. nodes for auxiliary words, like prepositions or modal verbs, disappear on the tectogrammatical layer. Nevertheless, the information of these words is reconstructable from the attributes of the "meaningful" nodes.

Nodes that were deleted on the surface level are restored to the dependency tree. That means that for elliptic constructions, new nodes are generated and added to the representation. All relevant information is then copied to newly generated node. The judgment of when to generate an extra node is driven especially by the concept of valency (see section 2.1.3).

The tectogrammatical layer can be viewed as having four different sublayers of annotation: semantic dependencies and functions, grammatemes for mor-

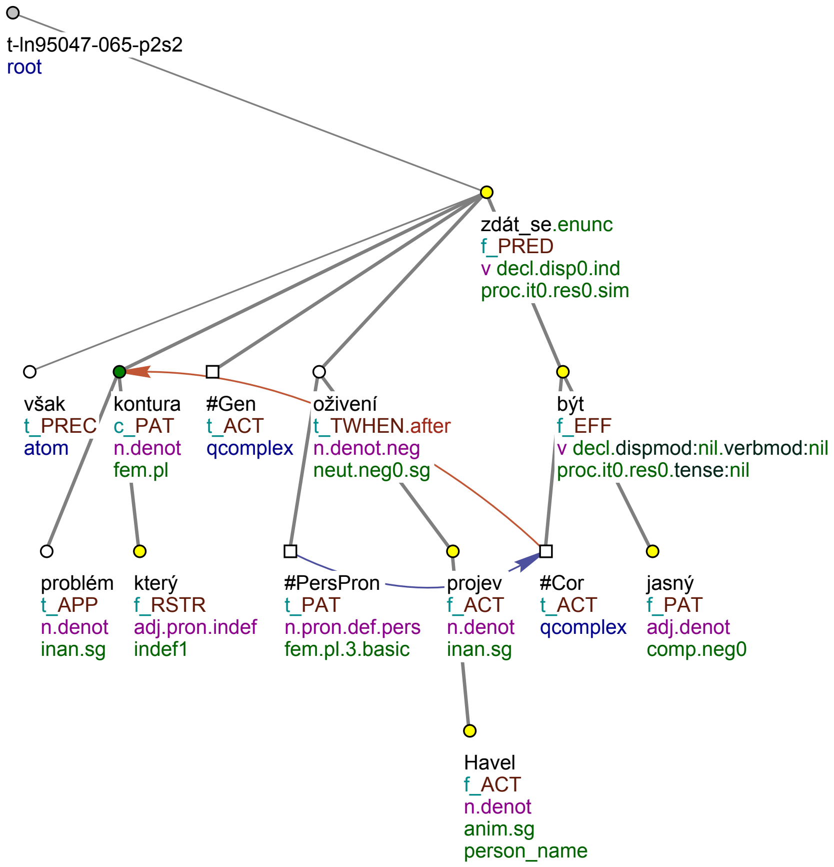


Figure 2.2: Example tree of the tectogrammatical layer for the sentence ”*Některé kontury problému se však po oživením Havlovým projevem zdají být jasnější.*” (engl. ”*Some contours of the problem seem to be clearer after the resurgence by Havel’s speech.*”) (taken from [Hajič *et al.*, 2006a]).

phological categories, grammatical and textual coreference and, finally, topic-focus articulation. Although complete tectogrammatical trees are the input for the mapping that will be presented in the next chapter, only parts of their representation can be mapped onto the target formalism in a straightforward way, as it will be stated later.

Figure 2.2 shows an example dependency tree from the tectogrammatical layer of the PDT. The relations between the nodes are given by the tree structure. Each nodes displays its *tectogrammatical lemma* in the first row. In the second row the *topic-focus articulation attribute* and the *functor* are presented, separated by an underscore. The third and forth row show *grammatemes*: the *semantic part-of-speech* or the *node type* (if no semantic part-of-speech can be assigned) and important morphological categories. Where appropriate, other attributes (like person_name) are displayed in a fifth row. Arrows symbolize *coreference links*. All the just mentioned node attributes will become clearer in their necessity in the following subsections. The final subsection summarizes the most important ones for this project.

Structure and Dependencies

As already mentioned, the nodes on the tectogrammatical layer correspond to *lexical words*, also called *autosemantic* words in the literature, which carry "linguistic meaning". This is a big difference to the previous layer. Analytical nodes representing function words, like prepositions, subordinate conjunctions, etc. correspond to attributes of lexical nodes on the tectogrammatical layer. The lemma of the tectogrammatical nodes is prototypically the same as the morphological lemma, however, there are cases where a substitute for the tectogrammatical lemma is used. Personal pronouns, for example, have a special string ("#PersPron") as their tectogrammatical lemma and store the properties of the pronoun (person, number and gender) in the node attributes, as part of the grammatemes (see below).

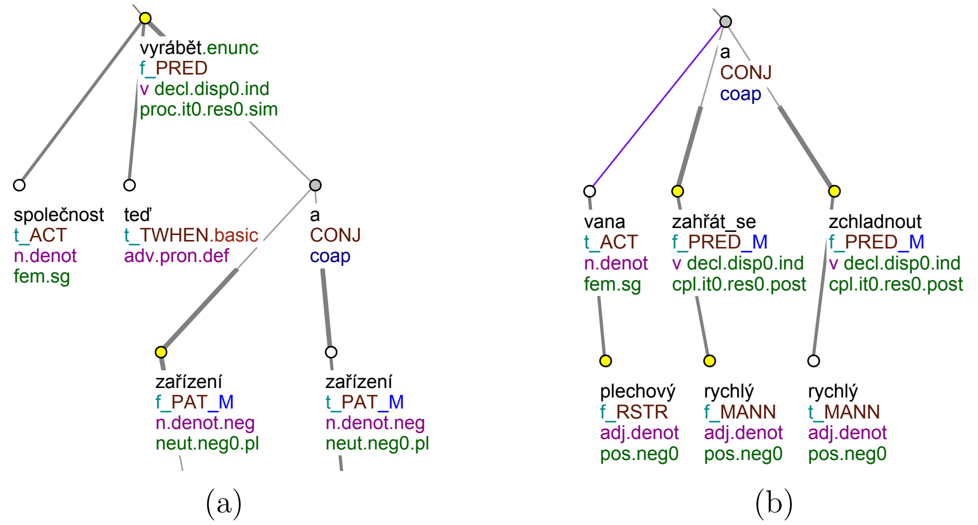
The nodes for lexical words are connected with labeled edges. The labels are called *functors*. They describe the type of the relation between the nodes. For some functors there is a set of possible subfunctors that further refine this characterization[4]. Again, the edge labels are stored in the attribute-value matrix of the dependent node (in figure 2.2: second row, after first underscore, in capital letters; subfunctor appears after a dot where assigned). There are four different major kinds of edges:

1. root (also, distinguish the technical root (topmost node) and the linguistically motivated root (child node of technical root))

2. dependencies (e.g. verbal participants, time, location, manner, etc.)

3. grouping (e.g. coordination, apposition, parenthesis)

[4] For a complete description of all functors and subfunctors that are used in PDT see [Mikulová *et al.*, 2006], chapter 7.



(a) (b)

Figure 2.3: Two tectogrammatical subtrees that illustrate the effective child relation. (a): "*Společnost nyní vyrábí zařízení* [...] *a zařízení* [...]." (engl. "*The company now manufactures* [...] *equipment and* [...] *equipment*") (b): "*Vana plechová se zahřeje rychle a rychle zchladne,* [...]." (engl. "*The tin bath heats up fast and cools off fast,* [...]")

4. other non-dependencies (e.g. negation, conjunction modification, part of an idiom, interjection, loose backward reference, etc.)

There is another important concept of how nodes are related to each other. The *effective child* relation resolves the complex interplay between dependency and coordination edges in tectogrammatical trees. When considering the effective child relation, coordination nodes are ignored for the purpose of getting "linguistic dependencies". On the other hand, in constructions without coordination (and apposition), the effective child relation corresponds to the ordinary child relation for tree structures.

To understand two complex cases of the effective child relation, consider figure 2.3, in which each of the two subtrees contains a conjunction node (a/CONJ/coap). This leads to the following behavior. The topmost node in figure 2.3a has all other nodes except for the conjunction node as its effective children. All direct dependents are considered and the conjunction node is "dived through", yielding the members of the conjunction (marked with _M) as effective children. All other nodes no not have effective children (that are visible in this subtree). In figure 2.3b, the effective children of the node for zahřát_se are the nodes for vana and for rychlý (the same holds for the node for zchladnout but with the other rychlý node). The direct dependents are again considered and additionally, because the zahřát_se node is member of a coordination, the direct dependents of the coordination node that are not members are added. This behavior leads to a more linguistic dependency relation that is free of grouping edges. Note that, when considering effective child relations, the representation is obviously not a tree any more and must be regarded as a graph, which must be taken into account when processing tectogrammatical data. The effective child relation will be revisited in a later chapter.

Grammatemes

Grammatical features are represented on the tectogrammatical layer as well. *Grammateme* is the term for the representation of morphological information that has an impact on the meaning. They are part of the attribute-value matrix associated with the lexical nodes in the tree. Grammatemes also capture some information that is elided on the tectogrammatical layer, such as auxiliary words and types of pronouns. Due to the rich morphology of Czech, there is a big set of grammateme values. Which type of grammatemes are attached to the different nodes is determined by the semantic class of the lexical word. For semantic nouns, for example, number and gender (among others) are specified, while verbs have (among others) tense and a couple of modality features. In figure 2.2, the word kontura is annotated as being feminine in gender and as appearing in plural form in the data. The main verb zdát_se is in "simultaneous" tense, indicative modality, imperfective aspect etc..

Coreference

Grammatical and some textual coreference relations are resolved and marked in the tectogrammatical tree ([Kučová and Hajičová, 2004]). Grammatical coreferences describe, for instance, control structures, i.e. the relationship between participants of verbs of control and participants of dependent verbs. Figure 2.2 contains a sample of a control structure. The actor (ACT) of the verb být has a coreference link, symbolized by an arrow, to kontura (which is the patient argument (PAT) of the governing verb zdát_se). Grammatical coreferences also annotates the antecedent of words like *which*, *whom*, etc., the antecedent of grammateme value inheritance, reflexive pronouns, relative pronouns, as well as some types of reciprocity. Textual coreference, on the other hand, is restricted to the use of demonstrative and anaphoric pronouns.

Topic-Focus Articulation

Information structure of a sentence is annotated using the two attributes for topic-focus articulation and deep word order. The deep word order puts the "newest" information to the right and the "oldest" information to the left in every subtree. The topic-focus attribute marks the division of those nodes into contextually bound and contextually unbound elements. In Figure 2.2, the main verb zdát_se and the node for být including its subtree constitute the focus of the sentence. The four other dependent subtrees under zdát_se are the topic of the complete structure.

Important attributes

This subsection is intended to be a short reference for all node attributes that are important in this project. Some of the given examples can also be found in figure 2.2. The reader is encouraged to come back to this subsection to get a rough idea of an attribute that is used in later chapters.

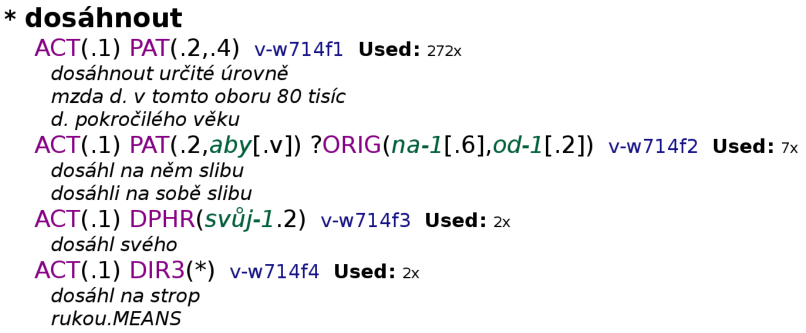
- **node type**: groups tectogrammatical nodes
  Possible values: complex for regular lexical nodes, qcomplex mainly for nodes elided on the surface syntactic level, atom for special types of modifications (like negation) without dependents, coap for coordination and apposition, list for list structures, fphr for foreign language expressions, dphr for idioms

- **tectogrammatical lemma**: represents the lexical content of the node or a substitute
  Possible values: e.g. problém,bt, Praha; #PersPron for personal pronoun, #Neg for negation

- **functor & subfunctor**: functors are semantic values of dependency relations

Possible values: e.g. ACT for actors, LOC.near for a location near something, AIM for purpose

- **grammatemes**: Grammatemes are tectogrammatical correlates of morphological categories
  - **semantic part-of-speech**: complex nodes can be classified as belonging to one of four semantic parts-of-speech (noun, adjective, adverb, verb) with subclassifications (e.g. possessive adjectives)
    Possible values: e.g. n.pron.def.pers for definite personal pronouns, adv.denot.ngrad.neg for denominating, non-gradable, negatable adverbs, adj.quant.grad for quantificational and gradable adjectives, v for verbs
  - others: note that there are 15 other grammateme values (e.g. person, number, aspect) for which the details will not be important
- **sentmod**: the sentmod attribute contains the information regarding the sentence modality
  Possible values: enunc corresponds to declarative clauses, excl to exclamative clauses, desid to optative clauses, imper to imperative clauses and inter to interrogative clauses.
- **member**: dependent nodes of coordination and apposition nodes (with the coap node type) have this attribute if they belong to the grouping. If a direct dependent of a coordination does not have this attribute, the node represents a modification of all members of the group or of the coordination itself.
- **person_name**: annotates if the node represents a name of a person
- **grammatical coreference**: links to another node to annotate control, complex predicates, reciprocity, grammateme inheritance and other grammatical coreferences



* **dosáhnout**

ACT(.1) PAT(.2,.4) v-w714f1 **Used:** 272x

dosáhnout určité úrovně
mzda d. v tomto oboru 80 tisíc
d. pokročilého věku

ACT(.1) PAT(.2,*aby*[.v]) ?ORIG(*na-1*[.6],*od-1*[.2]) v-w714f2 **Used:** 7x

dosáhl na něm slibu
dosáhli na sobě slibu

ACT(.1) DPHR(*svůj-1*.2) v-w714f3 **Used:** 2x

dosáhl svého

ACT(.1) DIR3(*) v-w714f4 **Used:** 2x

dosáhl na strop
rukou.MEANS

Figure 2.4: PDT-Vallex sample entry for the word *dosáhnout* (engl. *to reach*). It has the following frames: (1) to reach (a certain level), (2) to make sbd. promise sth., (3) to achieve one's goal, (4) to reach (up to sth.) (taken from [Hajič *et al.*, 2006a]).

2.1.3 Valency Dictionary

The valency dictionary of the PDT (PDT-Vallex, [Hajič *et al.*, 2003]) is a data source separate from the actual PDT annotation. The concept of valency for lexical words adopted in PDT is summarized in [Panevová, 1994]. The PDT-Vallex stores possible valency frames for individual words in the form of lists, capturing their valency complementations. It therefore can later be used to determine the arity of predicates in the target formalism.

The lexical entries in the dictionary contain one or more *valency frames*. These frames consist of a set of *valency slots*. Each slot is described by a single functor. Subfunctors are not described in the valency lexicon. Each functor has a flag marking obligatoriness for the valency frame. Obligatory valency modifications of the respective frame can be used to determine when to restore nodes in elliptic constructions on the tectogrammatical layer. Also, a list of possible surface expressions is stored for each slot as some slots require, for example, certain morphological cases or the use of a specific preposition. Passivization and other transformations are not explicitly represented in PDT-Vallex.

There are two main types of modifications distinguished: inner participants and free modifications. They correspond roughly to arguments and adjuncts. The difference is that free modifications can modify a verb multiple times and can (in principle) modify any verb. Inner participants may only appear once as a complementation of a particular word and can modify a more or less closed class of words. For verbs, these inner participants are Actor (ACT), Patient (PAT), Addressee (ADDR), Origin (ORIG) and Effect (EFF). Nouns additionally have the adnominal partitive argument (MAT) among the inner participants.

PDT-Vallex comprises all obligatory modifications (inner participants and free modifications) and all optional inner participants. Figure 2.4 shows an example entry of PDT-Vallex. If a node with a specific valency frame occurs in the data, a link to this frame is annotated at the respective node. For all verbs occurring in the PDT, the valency lexicon is complete. Some valency frames for nouns, adjective and adverbs are still missing. This means that this resource is incomplete regarding the whole set of lexical word types contained in the corpus. The arity of predicates for certain words can hence not be reliably predicted using the PDT-Vallex.

2.2 Minimal Recursion Semantics

Minimal Recursion Semantics (MRS, [Copestake *et al.*, 2005]) is a formalism for capturing semantic information that was especially designed for the needs in computational linguistics. This section summarizes the main ideas and concepts of MRS, introduces different notations and discusses a modified version, called Robust Minimal Recursion Semantics (RMRS, [Copestake, 20042006]) that was designed to be more dynamic and less demanding regarding lexical information. RMRS is the target representation for the mapping developed in the next chapter.

2.2.1 Motivation

MRS is a flat semantic representation that uses first-order predicate logic as an object-language. It is not a semantic theory, but rather a means to effectively deal with logical formulas. It is able to underspecify scope information, which results in a significant decrease in computational complexity for building the structures, while keeping the same expressive adequacy as the object-language.

It is, furthermore, intended to be compatible for use in a range of open domain and broad-coverage applications. The most prominent one is the English Resource Grammar (ERG, [Copestake and Flickinger, 2000]), a large, broad-coverage HPSG grammar that uses MRS as its semantic representation. Other applications of the formalism can be found in machine translation, statistical parsing, question answering, information extraction, ontology induction, sentence comparison and other fields in which semantic structures have to be related in an easy way. All these fields could profit from mapping multilingual language resources onto MRS structures, making more data available for deep as well as for shallow processing.

2.2.2 Description

An MRS representation consists of a triple, as shown in (2.1). This section explains all three elements and their purposes. There are, furthermore, two important notations of how to present MRSs, the standard way and as MRS graphs, which are both introduced below.

(2.1) < hook , EP bag , handle constraints >

The first element is the *hook* of the structure. It is important during the semantic of composition of complete MRSs. The second element is the *EP bag*. It is a set of predicates that describes the lexical and some relational semantic information contained in a sentence. The last element is a set of *handle constraints* that specify certain scopal relations of the elements in the EP bag.

At the heart of an MRS representation is a set of *elementary predications* (EP) called the *EP bag*. EPs are basic relations, similar to predicates in first-order logic. They normally correspond to a single lexeme, often referred to by its lemma. Every EP is marked by a label, has a relation name and a certain number of arguments, depending on the arity of the predicate. (2.2) shows the general notation of an EP. (2.3) presents an EP bag for the example sentence *Every white cat probably ate a mouse.*

(2.2) label: relation(argument$_0$, ..., argument$_n$)

(2.3) EP bag:

{ $l1$: _every_q($x1, h1, h2$),
$l2$: _white_adj($x1$),
$l2$: _cat_n($x1$),
$l3$: _probably_adv($e1, h3$),
$l4$: _eat_v($e2_{[tense:past]}, x1, x2$),
$l5$: _a_q($x2, h4, h5$),
$l6$: _mouse_n_1($x2$) }

(2.4) `_lemma_part-of-speech_sense-distinction`

There are certain conventions on how to name the *relations*, shown in (2.4). Relations that describe lexical words start with an underscore, followed by the lemma of the word, followed by another underscore and the part-of-speech information. Optionally, a last underscore can separate the part-of-speech from a number that constitutes an additional sense distinction among words with the same lemma and part-of-speech (e.g. a computer mouse vs. the animal in example (2.3)).

The logical conjunction operator $\wedge$ is given a special status in the MRS formalism ([Copestake *et al.*, 2005], page 288). In natural language it is generally used for composing semantic expressions, while the other logical connectives (disjunction $\vee$, etc.) only contribute to the semantics when they are lexically licensed. Also, they appear in more restricted contexts. As a consequence, *EP conjunctions* are made implicit by using identical labels for all members of the conjunction. The phrase *white cat* in (2.3) is constructed using identical labels, but note that implicit conjunctions are versatile in their potential usage. Prepositional phrases, for example, are constructed in the same way, labeling the preposition EP with the same label as the EP it is attached to.

There are different types of variables that are used in MRS. Table 2.1 lists all of them. Variables can have features attached to them that can carry morphological information. For example, nominal variables can have values for person, number and gender, while event variables carry tense and mood.

Every EP has *characteristic arguments* that get introduced depending on the part-of-speech. For nouns and adjective, the first argument is always a nominal variable (also referred to as *referential index* or *ref-ind*) that stands

Variable	Usage
a	*anchors* uniquely identify an EP (only in RMRS)
l	*labels* ”tag” one or more EPs
h	*holes* are arguments slots for embedding other EPs
x	*nominal* variables are introduced by nouns and adjectives
e	*event* variables are introduced by verbal and adverbial EPs
u	used to mark unspecified obligatory arguments
i	used to mark unspecified optional arguments

Table 2.1: Different types of variables used in the context of MRS. Anchors only appear in RMRS structures (see section 2.2.3).

for the nominal object. Verbs introduce ”neo-Davidsonian” event variables ([Copestake, 20042006], page 3) as their first argument. The same is true for adverbs, but they additionally introduce a hole variable. In general, all EPs that introduce hole variables are called *scopal EPs*. Quantifiers are also scopal EPs, as will be explained below.

Holes can be seen as empty slots for other EPs. By equating the holes with EP labels, a predicate logic formula with embedded predicates can be created. Such linkings are referred to as *configurations* or *scope-resolved* MRSs that represent the individual linguistic readings for a sentence described by an MRS. Possible configurations for the predicates in (2.3) are shown in figure 2.5. The MRS itself, however, is a flat representation and avoids embedding. Moreover, it is *underspecified* concerning the scope relations and stands for the set of all possible configurations that can be constructed by equating holes and labels.

Nevertheless, the possible linking of holes to labels must be restricted. For instance in (2.3), the scopal EP _probably_adv must always embed the EP _eat_v. The other way around would be incorrect, since in MRS, adverbs always embed the modified verb. The constraints on scope relations are formulated using the *qeq* relation (equality modulo quantifiers, $=_q$). A qeq relation always relates a hole to a label and states that the EP referred to by the label either instantiates the hole argument directly, or that one or more scopal EPs intervene, i.e. the referred to EP is embedded in other EPs. In consequence,

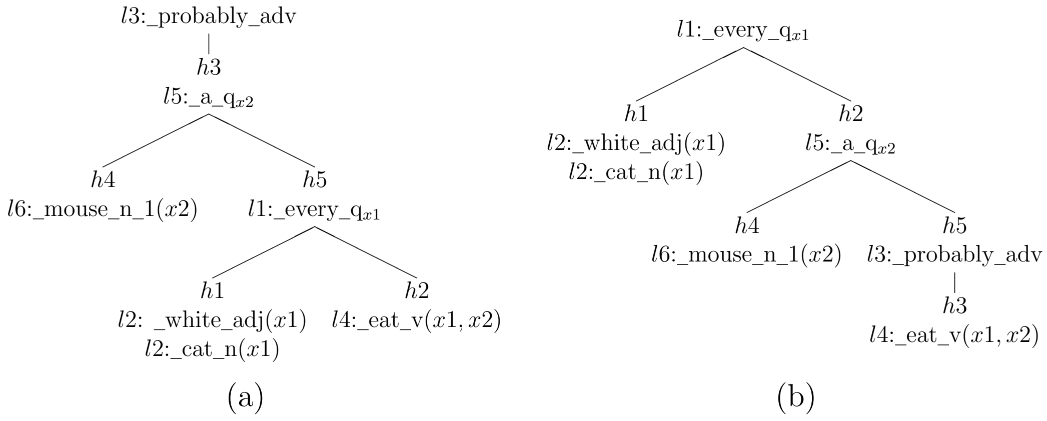


Figure 2.5: Two configurations for the EPs in (2.3)

for the case of adverbs, it remains underspecified whether the adverb modifies the whole verbal phrase (*probably* in figure 2.5a), parts of the verbal phrase or the verb alone (*probably* in figure 2.5b). All these possibilities are among the set of configurations that a single MRS describes.

The set of all qeq relations is called the *handle constraints*. It is the last element of the MRS triple. (2.5) shows the previous example augmented with its handle constraints. Notice that the configurations in figure 2.5 adhere to all the constraints.

(2.5) EP bag:
{ $l1$: _every_q$(x1, h1, h2)$,
$l2$: _white_adj$(x1)$,
$l2$: _cat_n$(x1)$,
$l3$: _probably_adv$(e1, h3)$,
$l4$: _eat_v$(e2_{[tense:past]}, x1, x2)$,
$l5$: _a_q$(x2, h4, h5)$,
$l6$: _mouse_n_1$(x2)$ }

Handle constraints:
{ $h1 =_q l2$, $h3 =_q l4$, $h4 =_q l6$ }

In logical formulas, all nominal variables must be bound by a quantifier. MRS uses generalized quantifiers, meaning that quantifiers are also EPs. Their

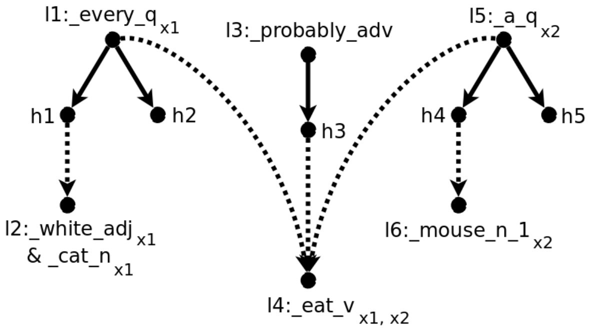


Figure 2.6: MRS graph for the MRS in (2.5)

characteristic arguments are the variable they bind, a hole argument for the restriction and a hole argument for the body. They additionally introduce a handle constraint. The hole of the restriction is qeq to the label of the EP that introduces the bound variable. This ensures that the quantifier embeds the correct EP. The body argument is left unconstrained.

Note that event variables are generally not explicitly bound by quantifiers. [Copestake *et al.*, 2005] assume an implicit wide-scoped quantifier, but admit that this might cause problems in specific cases. However, as it has been shown in [Goss-Grubbs, 2005], omitting the binding for event variables is justified, since it can be made explicit by a simple rule.

A visual representation of MRS structures can be given through MRS graphs. They describe how the different EPs can can be linked together to a configuration. Figure 2.6 shows the MRS graph for the example in (2.5). Subgraphs that are connected by solid edges are called the *fragments* of the graph. They represent the scopal EPs along with their hole arguments. The dashed arrows are called *dominance edges.* They either stand for a qeq relation (e.g. from the restriction of _a_q to _mouse_n_1) or an implicit outscoping requirement between a variable and its binding quantifier (e.g. the use of $x1$ and $x2$ in the EP _eat_v). MRS graphs visualize the different ways of how EPs can relate to each other (with examples from figure 2.6):

- implicit conjunction: EPs are joined into the same node (e.g. _white_adj & _cat_n)
- usage of the same nominal variable: there are dominance edges from the variables' quantifier node to all EPs that use the variable (e.g. $l1 \rightarrow l4$)
- handle constraints: qeq relations are represented by dominance edges outgoing from nodes representing hole arguments to nodes representing labels (e.g. $h1 \rightarrow l2$)

In the next chapter, these three types of relations are going to connect the partial MRSs constructed from different subtrees of the PDT representations. Furthermore, certain properties of MRS graphs are going to assist in defining valid MRS structures in the evaluation chapter.

The first element of the MRS triple is the *hook* ([Flickinger *et al.*, 2003], page 9). The hook is important for semantic composition of phrases and sentences, because both the EP bag and the handle constraints are sets. Using labels, the contained information can be referred to directly. The hook consists of the *top label* and the *index variable*, as presented in (2.6). They represent information that might be accessed externally. The top label is the topmost label considering all handle constraints and excluding quantifiers, and is important when constructing scopal relations. The index variable is used to fill argument positions in EPs with variables of their dependent complementations. Both of these features are accessed by the semantic head when constructing phrases, which will become more apparent in the next chapter.

(2.6) Hook:
[*top label*, *index variable*]

Given all necessary information, it is now possible to display the complete MRS triple for the discussed example sentence in (2.7).

(2.7) $< [l3, e2]$,
{ $l1$: _every_q$(x1, h1, h2)$,
$l2$: _white_adj$(x1)$,
$l2$: _cat_n$(x1)$,

$l3$: _probably_adv($e1, h3$),
$l4$: _eat_v($e2_{[tense:past]}, x1, x2$),
$l5$: _a_q($x2, h4, h5$),
$l6$: _mouse_n_1($x2$) },

$\{ \; h1 =_q l2, \; h3 =_q l4, \; h4 =_q l6 \; \} >$

Note that as a last step of the construction of a complete MRS, after all composition steps have been executed, a specific condition has to be fulfilled. The top must be set to a unique label that does not appear in the EP bag. This top label must not be outscoped by any other label in the EP bag for MRSs representing complete sentences. This is important in order for the MRS to represent all possible configurations, including those in which the top labeled EP is embedded in other EPs. Hence, for a formally correct MRS, the top in (2.7) must finally be changed to a unique label, for example $l0$.

2.2.3 Robust Minimal Recursion Semantics

Robust Minimal Recursion Semantics (RMRS, [Copestake, 20042006]) is a variant of the MRS formalism. It attempts to formalize a semantic description that can be used by both deep and shallow processing techniques. A hybrid combination of deep and shallow techniques can have several advantages and applications. In general, it is more robust to select a set of candidates from the raw data using shallow processing and, in a second step, deep process the individual candidates to extract the required information (e.g. for all fields of applications mentioned in 2.2.1).

RMRS factors out the arguments of the EPs. Therefore the arity of the predicates does not have to be known in advance and the approach can cope without a lexicon. In theory, the arguments of an EP can even be left *underspecified*, but more importantly, it is possible to add elements to the argument list during parsing time. This property makes RMRS more flexible and more robust than MRS.

As a consequence, there is the necessity for an additional way to identify EPs. After two EPs have been joined in an implicit conjunction through label identity, the label refers to the group of EPs and not to the individual EPs any more. But the outfactored arguments must be unambiguously attached to one individual EP. Copestake therefore extents the labeling of EPs by another element, in order to be able to uniquely identify EPs, even after implicit conjunctions. For this purpose, labels are accompanied by *anchors* when marking an EP ([Copestake, 2007a, Copestake, 2007b])[5]. Anchors uniquely identify an EP and never change, while labels can be changed to be equal to other labels during processing to form an implicit conjunction.

(2.8) presents the general form of the RMRS quadruple and (2.9) shows the discussed MRS as an RMRS. It has an additional set, that contains the out-factored arguments of the EPs as first-class predications. The hook is now a triple, additionally specifying the top anchor to which arguments can be attached.

(2.8) < [*top label*, *top anchor*, *index variable*],
EP bag,
arguments set,
handle constraints >

(2.9) $< [l3, a5, e2]$,
{ $l1$:$a1$: _every_q($x1$),
$l2$:$a2$: _white_adj($x1$),
$l2$:$a3$: _cat_n($x1$),
$l3$:$a4$: _probably_adv($e1$),
$l4$:$a5$: _eat_v($e2_{[tense:past]}$),
$l5$:$a6$: _a_q($x2$),
$l6$:$a7$: _mouse_n_1($x2$) },
{ $a1$: RESTRICTION($h1$), $a1$: BODY($h2$),
$a4$: ARG1($h3$),
$a5$: ACT($x1$), $a5$: PAT($x2$),
$a6$: RESTRICTION($h4$), $a6$: BODY($h5$) },
{ $h1 =_q l2$, $h3 =_q l4$, $h4 =_q l6$ } >

All out-factored arguments are named, e.g. actor (ACT) and patient (PAT) of _eat_v. They share the anchor with their corresponding EP. They are not dependent on the label, and remain EP specific even after implicit conjunctions.

[5] The approach using anchors is taken to be favorable over the approach with the in-g relation in [Copestake, 20042006].

If there is no name for an argument, as it is the case for the hole argument of the adverb, it defaults to ARGn with n being the number of existing nameless arguments for the EP. The first EP argument (named ARG0), however, remains part of the EP and is not outfactored. Also note that the hook has been extended by an anchor element as well to be able to add arguments to the main EP of the RMRS.

Two details of typical RMRS descriptions are omitted here. First, it is normally assumed that there are only unique variables (labels, nominal and event variables) in the representation, and that *variable equalities* are described in a separate set. For simplicity, variable equalities are resolved in all structures presented in this thesis. And second, character positions of the words in the original sentence string are usually also explicitly represented along with the EPs to facilitate anaphora resolution and to allow default quantifier scope readings. Copestake, however, admits that they are ”clearly not part of the 'real' semantics” ([Copestake, 2007a], page 4). They are therefore not used in the project at hand, but note that they could become important in future work when word order of the source representation is integrated into the mapping.

It is important to realize that MRS and RMRS are inter-convertible. Under the precondition that optional arguments of EPs are sufficiently instantiated ([Copestake, 20042006], page 8), i.e. explicitly represented using u and i variables (see table 2.1), it is possible to convert MRS structures into RMRS structures and vice versa. All argument EPs therefore have to be merged with their EPs identified by the anchors. Furthermore, variable identities must be resolved.

Chapter 3

Correspondence

This chapter describes basic ideas of how to represent the Prague Dependency Treebank 2.0 (PDT) data using the Minimal Recursion Semantics formalism (MRS). Section 3.1 clarifies the relation of the theoretical background and the formal description of the data. Section 3.2 discusses the type of MRS representation that will be formed. Section 3.3, most importantly, outlines the correspondence between the two formalisms. In section 3.4, the preserved information and the limitations of the mapping are listed. Finally, section 3.5 explains certain implementation design decisions of the mapping algorithm.

3.1 Theory vs. Formalism

The theory behind the source representation is the Functional Generative Description (FGD, [Sgall *et al.*, 1986]) for Czech. The PDT annotation scheme has been developed in accordance with the principles of this stratificational based theoretical background. MRS, however, is not a semantic theory. Due to the capacity of the formalism to allow for underspecifying scope relations, it is an efficient way to describe a set of object language expressions, i.e. predicate logic formulas. This means, in turn, that the mapping developed in this chapter cannot change the theoretical background of the source data. It will rather reformulate the annotation scheme from dependency trees into underspecified MRSs, while the theory behind it remains unaffected, as far as it is possible. Therefore, some modifications to the classical MRS descriptions have to be made, with the obvious ambition to keep as much information as possible from the PDT trees. The future goal, however, must be to map the complete annotation, which would make it possible to re-generate the original sentence strings from MRS representations.

3.2 Properties of the produced MRSs

To capture the necessary semantic information of the PDT annotation in MRS structures, the presented approach relies exclusively on the tectogrammatical layer and the valency dictionary. Each tree on the tectogrammatical layer, together with explicit valency information, will be represented by one MRS.

As described in section 2.1.2, the tectogrammatical layer can be viewed as having four sublayers: structure and dependencies, grammatemes, coreference and topic-focus articulation. The structural information along with the dependency relations are the core of the PDT semantics and they will be mapped utilizing the three relations among EPs listed on page 22. Grammateme values can be mapped in a straightforward way to variable features. For coreference links, this project concentrates on certain phenomena involving grammatical coreference and represents them through variable equalities. The topic-focus articulation will be ignored completely because describing information structure is not part of the classical MRS approach. However, [Wilcock, 2005] developed an extension of MRS to incorporate this information and his proposal could be useful for future work.

Normally, MRS representations are constructed from an input sentence string, using a pipeline going from part-of-speech tagging and syntactic parsing to the syntax-semantics interface. In a setup like this, it is easiest to introduce one EP for each lemmatized input word. The introductory literature to MRS is using examples of this nature. However, in the English Resource Grammar (ERG), a large grammar using MRS, there are constructions for which this is not the case. For example, expletive subjects, infinitival auxiliaries and some closed class words do not introduce EPs, because any semantics for them is undesirable. Moreover, lemmatization introduces the need to capture morphological information in variable features. Hence, there is a certain level of abstraction, away from the input string and the number of tokens, that is generally assumed.

The tectogrammatical annotation, as semantic part of PDT, takes this abstraction further. The tectogrammatical nodes represent exclusively lexical words. The information of functional words is captured in the attributes of nodes for the respective lexical word, e.g. in the form of functors or grammatemes. Although it would be possible to construct MRSs that contain EPs for all tokens from the input string using the lower annotation layers, but valuable high-level information would be discarded. This project will adapt the abstraction level of the tectogrammatical layer, which means any other words than lexical words according to FGD are not represented as EPs. EPs are introduced for most tectogrammatical nodes and functors. All grammateme values are mapped to variable features. Coreference links are utilized to form constructions with shared elements.

Additionally to the tectogrammatical trees, the information of the PDT valency lexicon will be incorporated in the target representation. However, as mentioned in section 2.1.3, the valency dictionary is incomplete for nouns, adjectives and adverbs. This means that there are occurrences of words along with inner participants and other valency modifications in the data, but there is no corresponding valency frame in the lexicon. This is problematic since, according to the FGD theory, inner participants are always part of the valency frame. Therefore, an EP lexicon with all predicate arguments cannot be compiled prior to parsing a PDT tree. It must be possible to add arguments to an EP dynamically. Hence, formally, Robust Minimal Recursion Semantics (RMRS) is the adequate choice over MRS to be used here. Valency modifications can then be represented using anchor equalities. Free modifications are expressed through label equalities, i.e. implicit EP conjunction.

The general approach adopted for the mapping is rule-based. Rules for constructing EPs from nodes and functors as well as rules for implementing the relations between the EPs build up a single RMRS structure for each tectogrammatical tree. These rules characterize the correspondence between PDT and RMRS representations.

3.2.1 Skipped Phenomena

This thesis describes the first attempt to investigate the relation between the PDT annotation and (R)MRS. Its goal is to lay a foundation for a complete mapping that is useful for applications and future research. Complex linguistic constructions that exceed the range of this groundwork will be ignored, i.e. PDT trees with these constructions will be skipped. In the remainder of this chapter, the triggers that cause a skip will be pointed out where appropriate and section 3.4.2 will recap all of them.

3.3 Correspondence between the Formalisms

This main section of the thesis outlines the correspondence between the PDT annotation scheme and RMRS by describing a method for mapping the one representation onto the other. First, in section 3.3.1, *node-RMRSs* are introduced. They are partial RMRSs that represent a subtree rooted at a certain node. Second, section 3.3.2 introduces the concept of *functional roles* for nodes that will be a helpful indicator for the types of variables and constraints that have to be used for certain EPs. In section 3.3.3, the initialization of the node-RMRSs is outlined. Section 3.3.4 explains how to relate node-RMRSs to each other in terms of valency and free modification, and for coordinations. The final section 3.3.5 presents which nodes are relevant for this step of combining node-RMRSs.

3.3.1 node-RMRS

For the task of this mapping, each tectogrammatical node has an RMRS associated with it that is called *node-RMRS*. It represents the tectogrammatical subtree rooted at the respective node. For most nodes, the EP bag of this node-RMRS contains at least one EP, called the *lexical EP*, representing the lexical node information. For leaf nodes this EP is the only element of the EP bag of the node-RMRS (plus potentially a quantifier EP). For non-leafs, the EP bag additionally contains the lexical EPs of all "MRS-dependent" nodes of the descendants. The concept of "MRS-dependents" is explained in the last subsection. The node-RMRS of the root node[1] of a tree is ultimately the complete RMRS representation for this PDT tree. Figure 3.2 shows the node-RMRSs at each node for the example tree in figure 3.1.

[1] In this chapter, the term *root node* never refers to the technical root, but always to the linguistically motivated root which is the child node of the technical root.

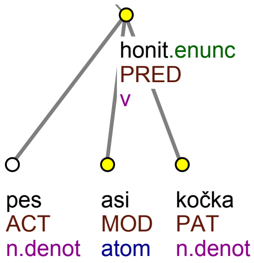


Figure 3.1: Fictive example tectogrammatical tree (omitting the technical root node) for the sentence "*Pes asi honí kočku.*" (engl. "*The dog probably chases a cat.*")

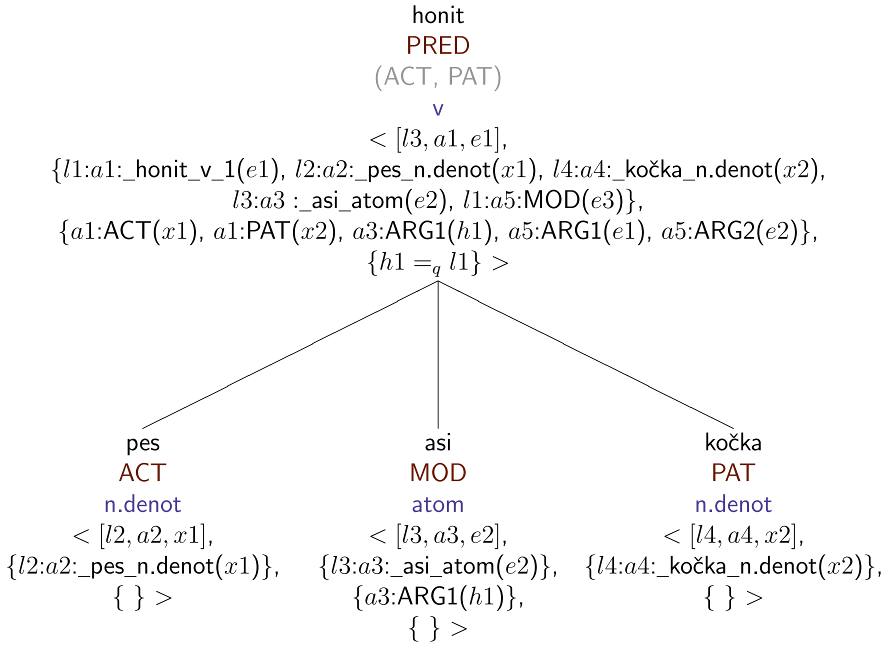

honit
PRED
(ACT, PAT)
v
$< [l3, a1, e1]$,
$\{l1{:}a1{:}$_honit_v_1$(e1)$, $l2{:}a2{:}$_pes_n.denot$(x1)$, $l4{:}a4{:}$_kočka_n.denot$(x2)$,
$l3{:}a3$:_asi_atom$(e2)$, $l1{:}a5{:}$MOD$(e3)\}$,
$\{a1{:}$ACT$(x1)$, $a1{:}$PAT$(x2)$, $a3{:}$ARG1$(h1)$, $a5{:}$ARG1$(e1)$, $a5{:}$ARG2$(e2)\}$,
$\{h1 =_q l1\} >$
pes
ACT
n.denot
$< [l2, a2, x1]$,
$\{l2{:}a2{:}$_pes_n.denot$(x1)\}$,
$\{\ \} >$
asi
MOD
atom
$< [l3, a3, e2]$,
$\{l3{:}a3{:}$_asi_atom$(e2)\}$,
$\{a3{:}$ARG1$(h1)\}$,
$\{\ \} >$
kočka
PAT
n.denot
$< [l4, a4, x2]$,
$\{l4{:}a4{:}$_kočka_n.denot$(x2)\}$,
$\{\ \} >$

Figure 3.2: Tree of figure 3.1 enriched by valency frame functors (in round brackets) and node-RMRSs. Quantifiers and variable features are omitted.

Two questions immediately arise: how are the node-RMRSs, including the lexical EPs, constructed? And how are they related to each other? A central idea in dealing with these questions is that of the *functional role*.

3.3.2 Functional Roles

The concept of *functional roles* is introduced in this section. It is based on a typology of PDT tectogrammatical node attributes and is important in the RMRS construction for two main reasons. First, a functional role defines which information is used to construct lexical EPs. That includes the number and type of variables that are used. The construction of lexical EPs is outlined in section 3.3.3. And second, it indicates how the combination of node-RMRSs works, regarding the introduction of additional EPs, the usage of variables and the adding of handle constraints. This issue will be addressed in section 3.3.4.

The functional role for a node depends on the `nodetype` attribute and for some nodes on additional properties. Table 3.1 lists all functional roles along with node type of tectogrammatical nodes that carry them. For complex nodes, the functional role is simply the value of the semantic part-of-speech attribute truncated at the first dot. The four major functional roles are therefore ***n***, ***adj***, ***v*** and ***adv***. Some quasi-complex nodes (i.e. nodes expressing elided nodes) also get one of those functional roles. Nodes for elided nouns and verbs work in the same way as their explicitly expressed counterparts. The general argument node (tectogrammatical lemma #Gen) gets the *n* functional role as a default behavior because it typically stands for an elided nominal argument. Atomic nodes basically behave like adverbs[2], in that they scope over their governing node. The last two statements should be confirmed by future experiments. Foreign language expressions, nouns that start with a capital letter and nodes with the `person_name` node attribute are treated like named entities. They behave like *n* when being combined with other structures and differ only slightly from them considering the lexical EP construction.

There are five more functional roles while three of them are specifics for coordinations. Nodes with the node type `coap` represent coordination and

[2] Nodes with the PREC functor are the exception to this rule. These nodes refer to the preceding sentence context. A coordination with an empty left element would be more accurate.

Node type	Additional properties	Functional Role
complex	semantic part-of-speech starts with n.	*n*
	semantic part-of-speech starts with adj.	*adj*
	semantic part-of-speech is v	*v*
	semantic part-of-speech starts with adv.	*adv*
qcomplex	tectogr. lemma is #Gen	*n*
	tectogr. lemma is #EmpNoun	*n*
	tectogr. lemma is #EmpVerb	*v*
	tectogr. lemma is #Unsp or #Oblfm	*empty*
	tectogr. lemma is #Cor, #QCor or #Rcp	*resolve*
coap	f. roles of direct members are *n* or *adj*	*coord_nom*
	f. roles of direct members are *v*	*coord_v*
	f. roles of direct members are *adv*	(skip)
atom		*adv*
list		*coord_nom*
fphr		*n*
dphr		(skip)

Table 3.1: General overview of all implemented functional role assignments

apposition. Both phenomena are treated equally in PDT and in this project. Furthermore, list nodes are basically conjunctions of nominal objects and therefore will be treated as coordination. The coordination functional roles are split into ***coord_nom*** (for both coordinated nouns and coordinated adjectives), ***coord_v*** and ***coord_adv***. Those three types behave differently when combining node-RMRSs. The classification in one of these types for a coordination is done by getting the consistent functional role of all member nodes of a coordination. Inconsistent functional roles cannot be handled in the current approach. Furthermore, *coord_adv* is ignored in this project as it is the most complicated functional role. Note that idioms (node type dphr) are also skipped because the author is not a Czech speaker and therefore lacks insight into potentially complex meanings and constructions.

The last two functional roles concern only quasi-complex nodes. The ***resolve*** functional role is assigned for nodes with the tectogrammatical lemma #Cor, #QCor or #Rcp. Control constructions, complex predicates and reciprocal relations all have the characteristic that two nodes share the same modification in the tree representation. This is annotated using coreference links. For RMRS structures, all these phenomena use identical nominal variables in two different EPs. In order to provide this identity, the hook of nodes with the *resolve* functional role is inherited from the grammatical reference antecedent node. Details on how the *resolve* functional role influences the mapping are shown at a later point.

Quasi-complex nodes with the lemmas #Unsp or #Oblfm have the ***empty*** functional role. Both nodes express obligatory elements that are elided on the surface level. In the target RMRS, the valency position or obligatory adjunct will be left unspecified using variables starting with u. However, this only works if the nodes occur as leafs, since this specific behavior is too simple to incorporate a whole subtree.

All tectogrammatical lemmas of quasi-complex nodes that were not mentioned in this section (such as #AsMuch, #Equal, #Some, #Comma, #Dash, etc.) will cause the tree that contains the respective node to be skipped in the experiments of the current project. Later works on the PDT to RMRS mapping should add their behavior to the procedure.

3.3.3 node-RMRS Initialization

This section describes the initialization of the node-RMRSs for different nodes based on node attribute information alone without considering related nodes. The most important part is how the lexical EPs are constructed. Furthermore, quantifiers have to be added for nominal objects. And finally, the hook of the node-RMRS has to be established. The combination of the indiviual node-RMRSs, that are initialized in the way described here, is outlined in section 3.3.4.

Lexical EP construction

A lexical EP is the main predicate of the node-RMRS for each node, carrying the lemma information, along with the morphological categories.

Relation Name Prototypically, the relation name for a lexical EP is built in the way displayed in (3.1):

(3.1) `_t-lemma_semantic-part-of-speech_valframe-index`

Three parts are separated by underscores: the tectogrammatical lemma of the node, its semantic part-of-speech and the index of the valency frame for the word. The last one distinguishes different meanings of the same lemma. If there is no valency entry in the dictionary this, last element is omitted and the relation name ends with the semantic part-of-speech. Nodes that do not have grammatemes, like coap, list and atom, do not have a semantic part-of-speech attribute. For coordination structures (coap and list), their functor is put into the position instead. This is important, as their functor information cannot be preserved in a different way (as will be shown below). Nodes with the atom node type simply have `atom` instead of the semantic part-of-speech in the relation name.

Examples in (3.2) show relation names for the Czech words *honit* (engl. *to chase*) with the first valency frame, and for the word *kočka* (engl. *cat*) that is a denominating noun without a valency frame.

(3.2) `_honit_v_1`
`_kočka_n.denot`

Named entities and cardinal numbers are constructed in a special way. They have `named`, `foreign` or `number` as relation name. This is typically done to generalize over named entities. The semantic part-of-speech of these nodes, however, is then not preserved through the relation name. The tectogrammatical lemma is mapped to a constant argument attribute (CARG) in the EP, as part of the characteristic arguments of these kinds of predicates (see upcoming paragraph and later (3.5)).

Characteristic Arguments Some argument positions are assigned to the lexical EP independently of the valency frame. This involves typically ARG0, CARG and in some cases ARG1. They serve the purpose of specifying the *characteristic arguments* for the lexical EPs, e.g. nominal variables for nouns and hole arguments for scopal EPs. The type of variables filling these characteristic arguments depends on the functional role of a node. Note that the nodes with the *resolve* and *empty* functional roles do not introduce any EPs, and therefore also no characteristic arguments. The other functional roles and

Functional Role	Attribute	Content
n	ARG0	nominal variable
n + is named entity	ARG0 CARG	nominal variable tectogr. lemma
adj	ARG0	nominal variable
v	ARG0	event variable
adv	ARG0 ARG1	event variable hole variable
coord_nom	ARG0	nominal variable
coord_v	ARG0	event variable

Table 3.2: Characteristic arguments added to lexical EPs independently of the valency frame

their effects on the lexical EP are listed in table 3.2. Valency modifications and coordination can add further arguments to an EP during the mapping, additionally to the listed ones.

Variable Features RMRS structures for English have the following features on variables: nominal variables can describe person, number and gender; event variables can have tense, mood and aspect features. Since Czech has a much richer morphology than English, it is necessary to extend the number of features on the variables. In fact, the whole set of grammateme values will be mapped to the variable features, incorporating the morphological information of the tectogrammatical layer. If a grammateme has the value `inher`, the value of the grammatical antecedent is inherited. The semantic part-of-speech is the only grammateme that is not mapped to variables features but into the relation name, as described above. The `sentmod` attribute, describing sentence modality of the main predicate, is the only non-grammateme that is also attached as an event variable feature of the main verb[3]. For nouns and adjectives, the values will be attached to the nominal variables and for verbs and adjectives, they get associated with the event variables.

[3] An alternative would be to introduce an extra EP, not outscoped by any other, that describes the mood of the sentence or clause.

Quantifiers

When introducing new nominal variables, a quantifier for each of them has to be created as well, in order to bind the variable. This is relevant for nodes with the *n* and with the *adj* functional role. Therefore, a special generalized quantifier EP will be produced. It has the nominal variable in ARG0 and it has two hole arguments. The hole of the restriction argument is qeq to the lexical EP that introduces the nominal variable. The hole in the body argument is left unconstrained. This quantifier EP is added to the EP bag of the node-RMRS along with the lexical EP. The qeq constraint is added to the set of handle constraints. As mentioned before, event variables do not need a binding quantifier.

It is important to note that specific quantifiers, triggered by certain words, are not implemented in this project. For example, the Czech equivalents for *every* or *some* are missing, i.e. they are treated as normal non-scoping lexical words. All nominal variables are bound by a general quantifier. Future work by a Czech linguist should develop a complete list of nodes that trigger a quantifier EP and implement their initialization.

Hook

The hook element of an RMRS structure represents information that can be accessed by external RMRSs. The top label is set to the label of the lexical EP when initializing. However, it can change during the course of the RMRS construction when other predicates outscope the lexical EP. The top label is used in scopal combinations of multiple RMRSs (as explained the next section). The top anchor is set to the anchor of the lexical EP, in order to be able to add arguments to it at a later stage. The index variable is set to the ARG0 argument of the lexical EP. It can later fill argument positions of other EPs. The latter two hook elements only change when dealing with coordinations.

The node-RMRS for an example occurrence of the Czech word *honit*, when it is initialized, is presented in (3.3). The first valency frame is annotated, which will later result in the adding of actor and patient to the arguments set. The EP bag only contains the lexical EP. Some variable features (subscript of event variable $e1$) are omitted. The top label, top anchor and index correspond to the label, anchor and ARG0 of the lexical EP. There are no arguments to the EP and there are no handle constraints.

(3.3) $< [l1, a1, e1]$,
{ $l1$:$a1$:_honit_v_1($e1_{[\ldots,resultative:res0,tense:ant,verbmod:ind]}$) },
{ },
{ } >

In (3.4), the initialization of an example occurrence of *kočka* is presented. The EP bag only contains the lexical EP and a quantifier. The hook is constructed as before. The quantifier is binding the introduced nominal variable and has two characteristic arguments. The hole in the restriction is qeq to the lexical EP.

(3.4) $< [l2, a2, x1]$,
{ $l2$:$a2$:_kočka_n.denot($x1_{[number:sg,gender:fem]}$), $l3$:$a3$:udef_q($x1$) },
{ $a3$:RESTRICTION($h1$), $a3$:BODY($h2$) },
{ $h1 =_q l2$ } >

The initialization for an occurrence of the named entity *Havel* (figure 2.2, page 9, lowest level node) is shown in (3.5). The lexical EP has the relation name `named` and the tectogrammatical lemma is a constant argument of the lexical EP.

(3.5) $< [l4, a4, x2]$,
{ $l4$:$a4$:named($x2_{[number:sg,gender:anim]}$), $l5$:$a5$:udef_q($x2$) },
{ $a4$:CARG(`"Havel"`), $a5$:RESTRICTION($h3$), $a5$:BODY($h4$) },
{ $h3 =_q l4$ } >

3.3.4 node-RMRS Combination

In the process of building the node-RMRS for all inner nodes and, most importantly, for the topmost node, i.e. the RMRS representation for the complete sentence, the node-RMRSs of the different subtrees are combined. When a governing and an MRS-dependent (see next subsection) node-RMRS are combined, the union of the two EP bags, the union of the two argument sets and the union of the two handle constraints of both structures are built. Additionally, the lexical EPs of both node-RMRSs must be related in a meaningful way that reflects the structure of a PDT tree in flat RMRS terms. In general, there are three different prototypical ways how two linguistically related EPs can be connected with each other:

1. an EP is a valency modification of its governing EP
2. an EP is a free modification of its governing EP
3. an EP is a member of a coordination or apposition

Valency Modification

The valency dictionary PDT-Vallex lists all obligatory and non-obligatory valency slots (labeled with functors) for a given word. Valency modifications of a particular word can thereby, in most cases, be identified accurately by their functors. In RMRS, the arguments of an EP are first class predicates in the argument set. They share their anchor with the EP that they are an argument for. The anchor is accessed through the top anchor feature of the hook by the governing node-RMRS. The arity of predicates can be specified and manipulated in this way. If a valency modification, i.e. a functor that is in the valency frame of the governing node, is encountered in the data, an argument is created and added to the argument set. The argument is named after the functor and subfunctor information of the dependency. This way, functor and subfunctor information is preserved during the mapping in valency modifications.

The type of variable filling the argument, i.e. the variable in ARG0 of the argument predicate, depends on the functional role of the dependent node. *n* and *adj* assign the nominal variable stored in the index feature of the hook of the dependent node-RMRS to the slot. For *v*, a hole variable is assigned to ARG0 and a constraint is added that relates this hole qeq to the top label of the dependent node-RMRS. Nodes that have one of the coordination functional roles behave according to the functional role of their members when filling valency slots. Furthermore, they inherit the functor from the members. The functional role and the functor information must, hence, be consistent among all the members.

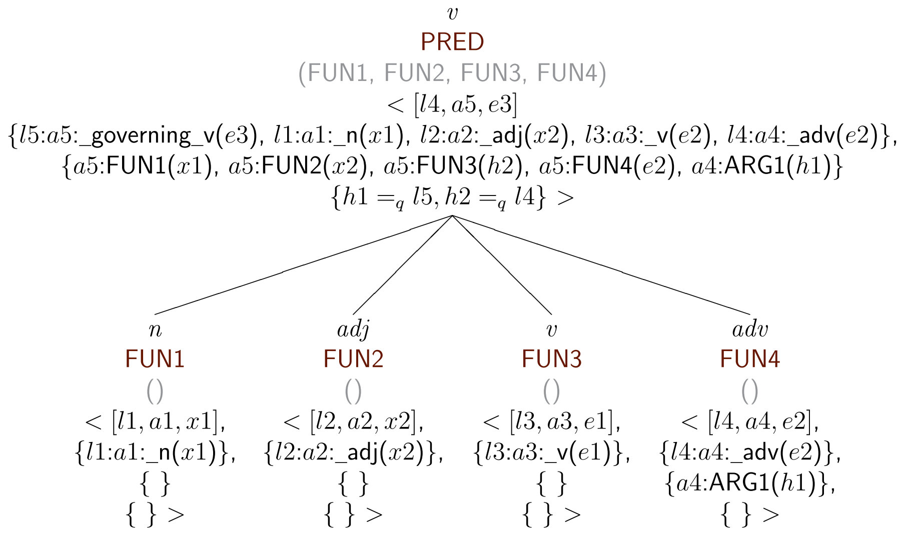


Figure 3.3: Fictive tree fragment to illustrate valency modifications. Valency functors are shown in round brackets. The relation names are all shortened, since there are no complete tectogrammatical lemmas and semantic part-of-speech information in this example. Quantifiers are omitted.

The *adv* functional role invokes the most complex behavior. First, the event variable in the index is assigned to ARG0 of the argument predicate. But since adverbs are scopal predicates, the lexical EP of the *adv* node-RMRS additionally has to outscope the lexical EP it is modifying. Therefore, a qeq-constraint that relates the hole argument of the adverb to the governing lexical EP label is added to the governing node-RMRS. After that, the top label of the governing node-RMRS has to be updated to the top label of the dependent node-RMRS. Figure 3.3 shows schematic examples for all types of valency modifications.

Note that nodes with the *adv* functional role further show a special characteristic. Dependency structure formalisms, and therefore the PDT annotation, are known to not specify whether an adverb is modifying only the verb or the whole subtree rooted at the verb. This underspecification is kept in the RMRS representation, because the qeq-relation only states that the adverb has to outscope the verb. The configurations for an underspecified structure include all possible versions of modification scope (”only-verb” modification, ”whole-subtree” modification and all possible intermediate modifications).

Note that the PDT annotation includes a dependency functor named RSTR for adnominal adjuncts modifying nouns. This name conflicts with the typical naming of one of the characteristic arguments of quantifiers. To ensure a unique semantics for each EP and for each argument, one of the relation names has to be altered. Therefore, the name of the argument slot for the restriction of quantifiers is renamed to `RESTRICTION`.

The valency dictionary, however, cannot be the sole source of determining valency. As described in section 2.1.3, there are six specific functors, i.e. inner participants, that are always valency arguments and never free modifications. Unfortunately, there are occurrences of these functors in the data for which their governing nodes do not have them in their valency frame. This is due to an incomplete annotation in the form of a missing dictionary entry, a missing functor in the dictionary entry or a missing link in the node attributes (mostly adjectives and nouns are affected). In the project at hand, the functors ACT, PAT, ADDR, ORIG, EFF and MAT are always treated like valency arguments to ensure a sound behavior with respect to the FGD theory. That means that it is not predictable how many argument positions a lexical EP will have by looking at the valency frame. The RMRS formalism, however, is flexible enough to deal with this problem, as it is possible to dynamically add arguments to an EP during processing.

The opposite case, however, in which not all valency positions are filled, does also occur in the data. To provide the correct arity of the predicates, argument EPs for the unfilled valency positions have to be added as well. Non-obligatory unfilled arguments will be marked with variables starting with i, to signal that no filler for the argument slot was found. For obligatory unfilled valency arguments, marked in the PDT tree with a node having the *empty* functional role, variables starting with u are used, as shown in figures 3.4 and 3.5.

The *resolve* functional role is used for control structures, complex predicates and some cases of reciprocity. All these structures have grammatical coreference links going out from extra generated nodes that fill a certain valency slot (usually the ACT position). In RMRS, the listed phenomena result in identical variables in multiple EP arguments. If a node has the *resolve* functional role, the hook of the grammatical antecedent node-RMRS is copied to the current

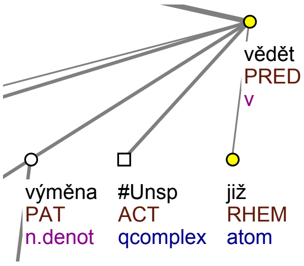


Figure 3.4: Example tectogrammatical subtree for the substring "[...] *o výměně již věděli.*" (engl. "[...] *([they] already knew about the exchange.*") to illustrate the *empty* functional role

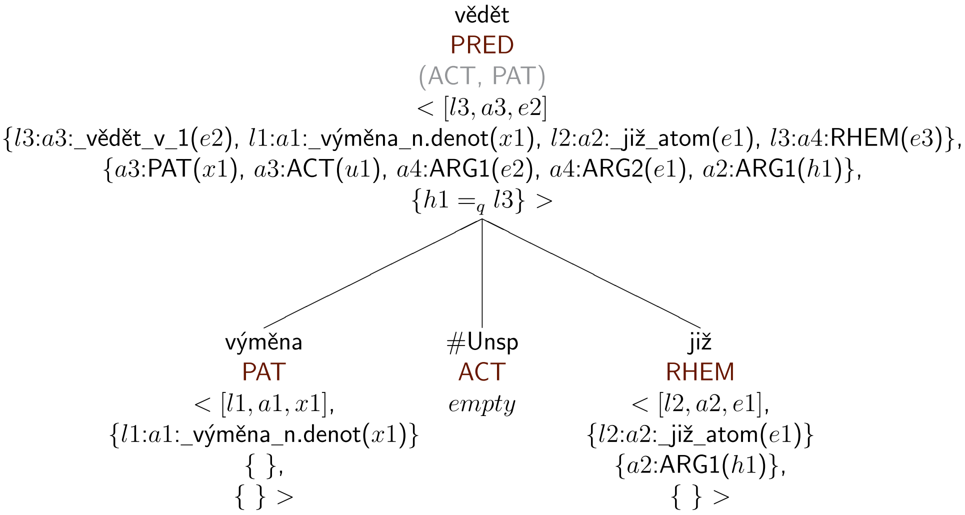

vědět
PRED
(ACT, PAT)
$< [l3, a3, e2]$
$\{l3$:$a3$:_vědět_v_1($e2$), $l1$:$a1$:_výměna_n.denot($x1$), $l2$:$a2$:_již_atom($e1$), $l3$:$a4$:RHEM($e3$)$\}$,
$\{a3$:PAT($x1$), $a3$:ACT($u1$), $a4$:ARG1($e2$), $a4$:ARG2($e1$), $a2$:ARG1($h1$)$\}$,
$\{h1 =_q l3\} >$
výměna
PAT
$< [l1, a1, x1]$,
$\{l1$:$a1$:_výměna_n.denot($x1$)$\}$
$\{ \}$,
$\{ \} >$
#Unsp
ACT
empty
již
RHEM
$< [l2, a2, e1]$,
$\{l2$:$a2$:_již_atom($e1$)$\}$
$\{a2$:ARG1($h1$)$\}$,
$\{ \} >$

Figure 3.5: Subtree of figure 3.4 including node-RMRSs

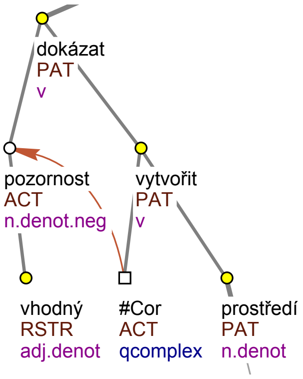


Figure 3.6: Example tectogrammatical subtree for the substring ”[...] *že vhodná pozornost dokáže vytvořit prostředí* [...]” (engl. ”[...] *that appropriate attention can create an environment of* [...]” to illustrate the *resolve* functional role.

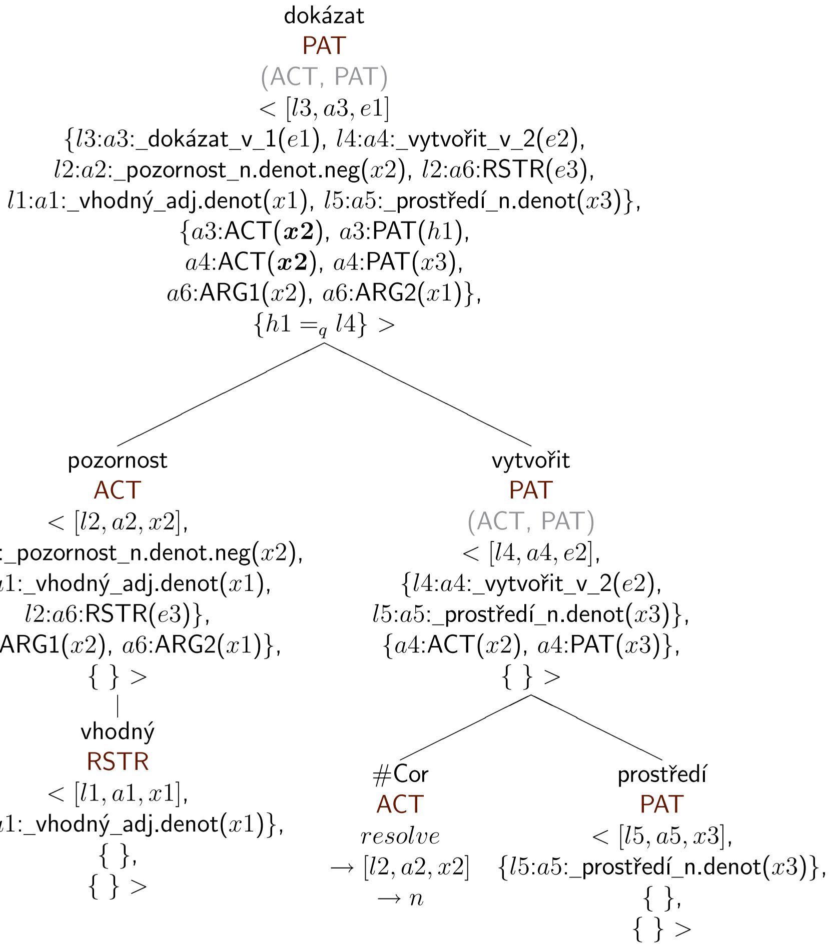

dokázat
PAT
(ACT, PAT)
$< [l3, a3, e1]$
$\{l3{:}a3{:}$_dokázat_v_1$(e1)$, $l4{:}a4{:}$_vytvořit_v_2$(e2)$,
$l2{:}a2{:}$_pozornost_n.denot.neg$(x2)$, $l2{:}a6{:}$RSTR$(e3)$,
$l1{:}a1{:}$_vhodný_adj.denot$(x1)$, $l5{:}a5{:}$_prostředí_n.denot$(x3)\}$,
$\{a3{:}$ACT$(\boldsymbol{x2})$, $a3{:}$PAT$(h1)$,
$a4{:}$ACT$(\boldsymbol{x2})$, $a4{:}$PAT$(x3)$,
$a6{:}$ARG1$(x2)$, $a6{:}$ARG2$(x1)\}$,
$\{h1 =_q l4\} >$
pozornost
ACT
$< [l2, a2, x2]$,
$\{l2{:}a2{:}$_pozornost_n.denot.neg$(x2)$,
$l1{:}a1{:}$_vhodný_adj.denot$(x1)$,
$l2{:}a6{:}$RSTR$(e3)\}$,
$\{a6{:}$ARG1$(x2)$, $a6{:}$ARG2$(x1)\}$,
$\{\ \} >$
vhodný
RSTR
$< [l1, a1, x1]$,
$\{l1{:}a1{:}$_vhodný_adj.denot$(x1)\}$,
$\{\ \}$,
$\{\ \} >$
vytvořit
PAT
(ACT, PAT)
$< [l4, a4, e2]$,
$\{l4{:}a4{:}$_vytvořit_v_2$(e2)$,
$l5{:}a5{:}$_prostředí_n.denot$(x3)\}$,
$\{a4{:}$ACT$(x2)$, $a4{:}$PAT$(x3)\}$,
$\{\ \} >$
#Cor
ACT
resolve
$\rightarrow [l2, a2, x2]$
$\rightarrow n$
prostředí
PAT
$< [l5, a5, x3]$,
$\{l5{:}a5{:}$_prostředí_n.denot$(x3)\}$,
$\{\ \}$,
$\{\ \} >$

Figure 3.7: Subtree of figure 3.6 including node-RMRSs. Processing of the RSTR node is described in the subsection dealing with free modification.

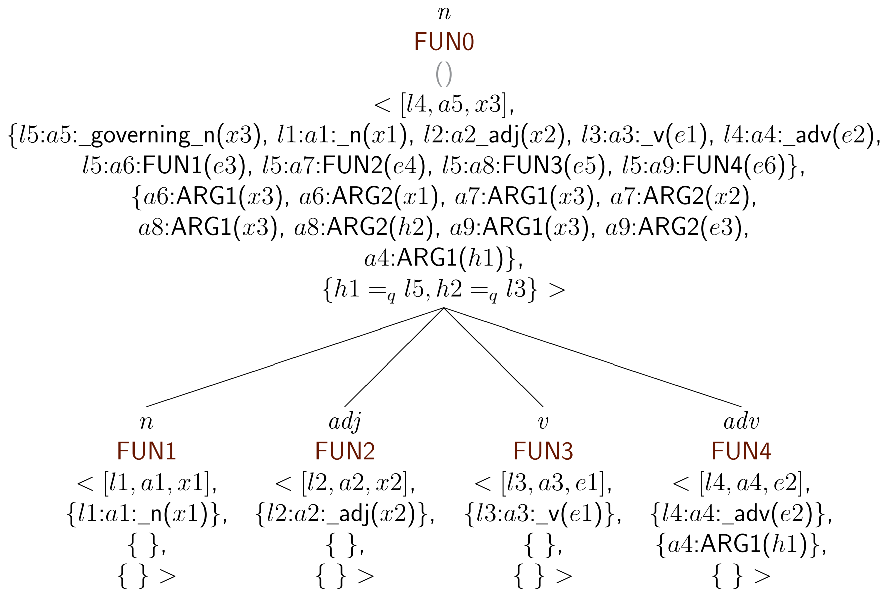


Figure 3.8: Fictive tree fragment for free modifications. None of the functors of the dependent nodes (FUN1-4) fill any valency position of the governing noun.

node. Furthermore, the functional role of this antecedent is inherited. These two pieces of information are enough to construct the correct structure with identical variables. That means that *resolve* nodes do not introduce a predicate. Due to the explicit reference links in the PDT tree, it is, moreover, not necessary to make use of the XARG feature in the hook[4]. Figures 3.6 and 3.7 show an example of how the *resolve* functional role works. Quantifiers are not displayed, but note that this method introduces free variables in node-RMRSs in the subtree containing the #Cor node. The quantifier binding $x2$ gets added at the top node from the left subtree.

Free Modifications

Non-obligatory free modifications are not part of the arguments of an EP. Another EP has to be added to the EP bag to establish the relation between the governing lexical EP and the modifier. This EP is called *connecting EP*

[4] The XARG feature is used by syntax-semantic interfaces to deal with unsaturated subjects in control structures and raising ([Flickinger *et al.*, 2003]).

here and carries the functor and subfunctor in the relation name (FUN1-4 in figure 3.8.). It hence inherits its semantics from the meaning assigned to the functors and subfunctors by FGD. The connecting EP is, furthermore, in an implicit conjunction with the lexical EP of the governing node-RMRS. That means, their labels are identical, but they have distinct anchors. The connecting EP has three characteristic arguments. ARG0 is initialized with an event variable, which is motivated by the similar treatment of prepositional phrases in the ERG. The variable in ARG1 is identical to the index variable of the governing node-RMRS. For ARG2, the same principles as for valency modifications are applied, i.e. the variable depends on the functional role of the dependent node. Figure 3.8 presents a fragment of a tree in which all the possible types of free modifications of a noun are illustrated. The behavior described in this paragraph was largely influenced by [ERG, 2009].

Coordination

The third main type of linking predicates is through coordination. It groups two or more elements of the same type together. In this mapping, coap and list nodes introduce *coordination EPs*. Because coap nodes annotate both coordination and apposition, both phenomena are handled identically. This project distinguishes two different types of coordinations: coordination of nominal objects (functional role *coord_nom*) for nouns and adjectives and coordination of verbs (functional role *v*). Coordination of adverbs is skipped.

Once a coordination EP is set up and is connecting the right elements, it can become an argument to governing structures. The coordination node will then inherit the functor and functional role from its members. Therefore, consistency considering the functional role as well as the functor of the coordination members is important. Inconsistency among the members will cause a tree to be skipped in the upcoming evaluation. A consequence of the functor inheritance is that the functor information of the coordination itself is not preserved through the argument relation of valency modifications or the connecting EP in free modifications. Because this information is important for the PDT tree structure and should be kept, the relation name for coordination EPs contains the functor of the coordination node where other EPs have the semantic part-of-speech (coordination nodes do not have a semantic part-of-

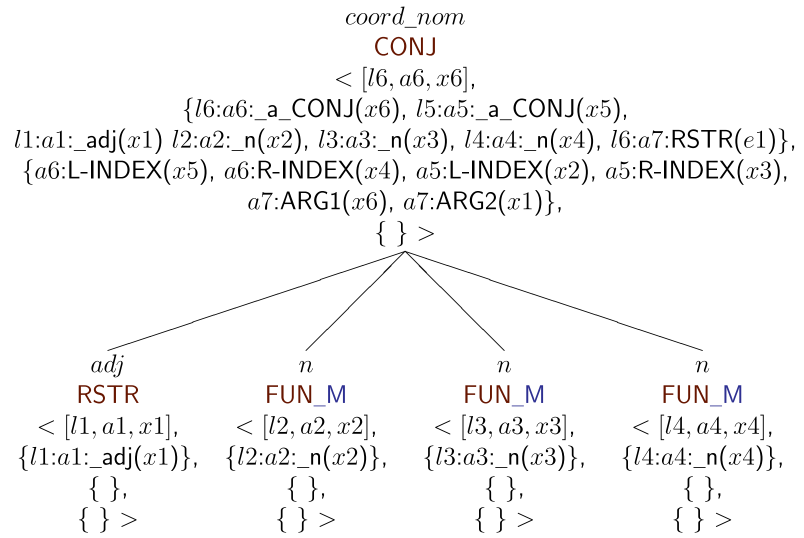


Figure 3.9: Fictive tree fragment for a coordination of nouns. The functor (FUN) and functional role (n) of all members is consistent.

speech attribute anyway). This way, the coordination functor information is preserved.

To form a coordination, all its members have to be grouped. This is done via EPs that binary link the elements together. For coordinations with more than two members a chain of binary relations is constructed. The topmost EP, or its nominal variable, can then represent the whole set of elements and serve as argument for other EPs. During processing, the top label and anchor as well as the index in the hook of the coordination node-RMRS have to be updated to the values of the topmost coordination EP. Note that this design is taken from the classical MRS approach ([Copestake *et al.*, 2005], page 322). However, for RMRS, it is also possible to add an argument to the coordination EP for each member, constructing n-ary coordination relations.

There are two different ways of grouping the elements, depending on the functional role of the member nodes. In general, there has to be a left and a right argument to establish the binary relation. The L-INDEX and R-INDEX positions are part of every coordinating EP. The index variables in the hook of the dependent node-RMRSs are taken into these positions. That is all that is done for *coord_nom*. For *coord_v*, two more attributes are introduced, namely L-HANDLE and R-HANDLE. In coordinations with the CONJ or with the DINJ functor, the top labels of the conjuncted node-RMRSs fill these attributes directly[5]. In all other coordinations, both HANDLE attributes have hole arguments and each hole is qeq to the top label of a coordinated node-RMRSs.

For non-member nodes that are dependents of a coordination node (that modify the whole coordination or each of the members), there are two cases to be distinguished. If they are in the valency frame of the members, they should to be connected to each member individually according to the principles of valency modification. Otherwise they are treated as free modification of the whole coordination (see the next subsection for more details about this behavior). Figure 3.9 shows an example that links three nouns together. The whole coordination is modified by an adjective. For governing structures, the functor (FUN) and the functional role (n) are inherited from the coordination members.

3.3.5 MRS-Dependents

MRS was developed for (head-driven) phrase-structure grammars. The way in which semantic expressions are formed was designed to integrate well with this syntax paradigm, as long as the marking of heads is guaranteed. Initialized structures at the leafs are combined with the structures of the same constituent on all tree levels by employing the semantic composition rules stated in [Copestake *et al.*, 2005], section 4.3.2.

The open issue examined in this section is which pairs of nodes of the tectogrammatical tree are relevant for combining their node-RMRSs using the three methods described in the last section (that are modified versions of the original composition rules). The nodes whose node-RMRSs should be combined with the node-RMRS of a given node, are henceforth called *MRS-dependents*. Even though tectogrammatical dependency trees specify the head of each modification directly through the tree structure, due to non-dependency edges, representing especially coordination and apposition nodes, finding the MRS-dependents for a node is more complicated than taking the

[5] *coord_adv*'s probably works similar to *coord_v*. More complex problems arise from the constraints on the hole arguments of the conjuncted adverbs.

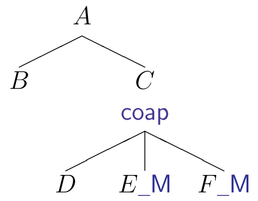


Figure 3.10: Schematic dependency tree to illustrate the effective child relation (possible sentence: "*Peter loves his mother and father*")

direct dependent nodes. Note, however, that MRS-dependents perfectly correspond to dependent nodes in PDT for simple dependency edges, which represent relations between modified and modifying nodes. In stating the rules to acquire the MRS-dependents for a node, the effective child relation (also mentioned in section 2.1.2) is utilized.

Effective child relation

The *effective child* relation in PDT tectogrammatical dependency trees resolves the complex interplay between dependency and coordination edges. Coordination and apposition nodes (both have the node type coap) play a special role due to their mere grouping function. They introduce non-dependency edges. All edges directing from and to coordination nodes do not represent linguistic dependencies but rather "grouping edges". That is why these edges are skipped or "dived through" in a certain way when considering the effective child relation. In consequence, only lexical nodes are related to one another. However, in structures without coordination or apposition, the effective child relation corresponds to the regular child relation in the tree.

To understand the principles of the effective child relation, consider the schematic tree in figure 3.10. Node A has a coordination or apposition node C among its direct dependent nodes. This node C, in turn, has three direct dependent nodes (D, E_M and F_M) with two of them being members of the coordination (marked by _M) and one direct dependent that is not a member. The effective children of A are nodes B, E_M and F_M. The regular direct dependent is included as well as the members of a direct dependent coordination node (this "diving through" coordination nodes is done recursively if

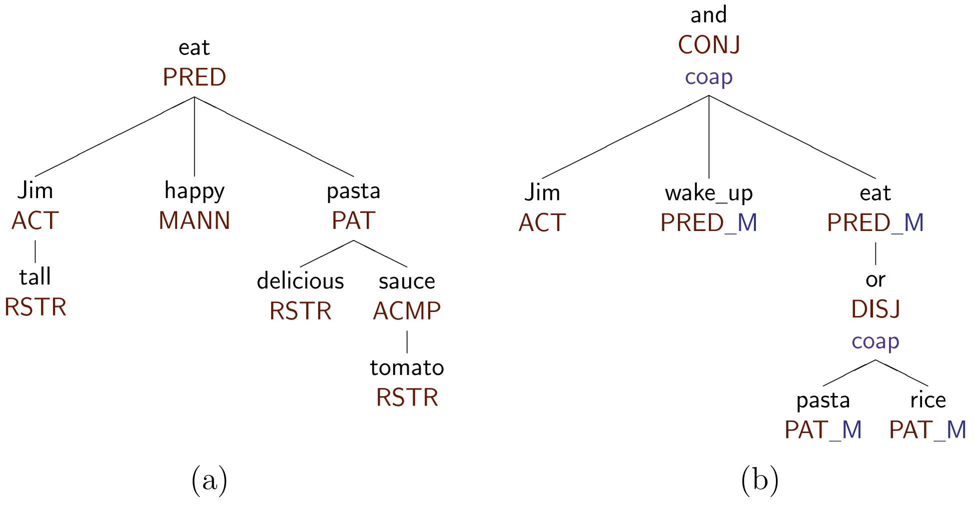


Figure 3.11: Fictive English example PDT trees. (a): *Tall Jim happily ate delicious pasta with tomato sauce.* (b): *Jim woke up and ate pasta or rice.*

there are embedded coordinations at the levels below). The effective child of both E_M and F_M is D which is representing modification shared by E_M and F_M. The two nodes are both members of a coordination and in this case, the effective children include the non-member nodes of this coordination node. B does not have effective children because it has no direct dependents and is not a coordination member. C does not have effective children as well because coap nodes do not have any in general.

Figure 3.11 shows this on two small examples. In figure 3.11a, the effective child relation corresponds to the child relation specified by the tree structure because there are no coordination or apposition nodes present. Each inner node has its direct dependents as its effective children and the leaf nodes do not have any effective children. In figure 3.11b, the two coordination nodes invoke a more complex effective child relation. Consider the nodes for *wake_up* and *eat*. They are both the members of a conjunction. Therefore, the non-member node under their governing coordination node, the node for *Jim*, belongs to their effective children. Since the *wake_up* node is a leaf, the node for *Jim* is its only effective child. The *eat* node, on the other hand, has a direct dependent node that is a disjunction. Therefore, the members of this disjunction, the nodes for *pasta* and for *rice* belong to the effective children. The node for *eat*, hence, has the nodes for *Jim*, *pasta* and *rice* as effective children. All other nodes do not have effective children. This example shows that the effective child relation captures a dependency that ignores coordination and apposition and establishes more "pure linguistic" dependency relations.

It is important to note that the rules for determining the MRS-dependents for a node rely on the effective child relation, but for structures containing coordination or apposition, they do not correspond to it directly. The rules stated next cover all cases of MRS-dependents and reflect how the PDT dependency format can be converted to represent the same structure in MRS terms.

MRS-Dependents for Complex & Quasi-complex Nodes

For complex and quasi-complex nodes, the set of MRS-dependents is acquired using three different rules. One of the rules also mentions a special case that influences the way of adding handle constraints.

Consider all effective children of the current node to be candidate nodes for becoming MRS-dependents.

1. **If there is more than one candidate with the same functor (for valency modifications even excluding the subfunctors), the lowest common ancestor node of all these nodes replaces these candidates and inherits their functor.**

   If there are effective children nodes with equal functors, they must occur under a coordination. For the target RMRS structure, we just want a single valency or free modification for all these nodes. That is why the coordination node grouping all of them will be the MRS-dependent representing all of them.
   In figure 3.12, the effective children of the v node are the nodes n_1, n_2 and n_3. These three nodes all have the same functor. The sole MRS-dependent of the verb node is the node $coord\_nom_1$, because it groups together all nodes with the ACT functor. For valency modifications, if there is no coordination grouping those functors, the PDT tree has to be skipped. A sound interpretation of valency frames in relation to EP

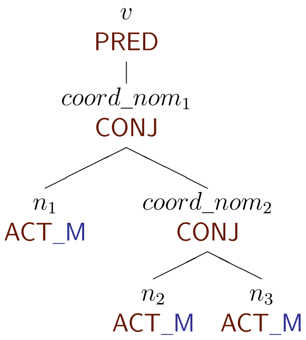


Figure 3.12: Fictive example PDT tree illustrating rule 1. Nodes are labeled with the functional role, an index, a functor, marking of coordination membership if applicable and valency frame functors.

arguments is not given in this case. The same is true for more than one occurring alternate functor of the same valency slot.

2. **If a candidate is not a descendant of the current node and it is not in the valency frame of the current node, it is deleted from the candidate list.**

   Effective children that are not a descendant of the current node are non-member nodes under a coordination or apposition node. That means that they modify multiple nodes, namely the member nodes of the coordination.

   Modification of multiple nodes in RMRS is different for valency modifications and for free modifications. For valency modifications, we add argument positions to each modified EP individually and fill these positions with variables that relate this EP to the modifier EP. Free modifications, on the other hand, introduce EPs themselves that will connect the coordination representing all modifiers with the modified EP. This is captured by another rule.

   Consider any of the two verb nodes in figure 3.13. The only MRS-dependents for both these nodes is the node n. The functor MANN is not in their valency frame and therefore *adj* is not processed by this rule even though it is an effective child. The *adj* node will be covered by rule 4.

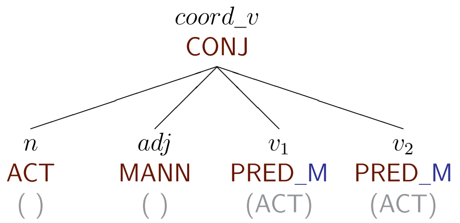


Figure 3.13: Fictive example PDT tree illustrating rule 2. Nodes are labeled with the functional role, an index, a functor, marking of coordination membership if applicable and valency frame functors.

(a) **If, however, a non-descendant *is* a valency modification *and* has the functional role *adv*, the handle constraint is formed in a special way, stated in (3.6).**

(3.6) $adv\_hole =_q top\_of\_shared\_parent\_node\text{-}RMRS$

The hole argument of the *adv* structure is qeq to the top label of the node-RMRS of the shared parent of the two involved nodes. This rule applies for the *adv* node in figure 3.14. $h1$ is qeq to the top label of the shared parent node-RMRS, which is $l4$. The reason to the exceptional way of forming the handle constraint is related to the coordination node that is governing the two verbs. If the handle constraint would be formed strictly according to the last section, the EP for the *adv* node would outscope both verb EPs ($h1 =_q l2$ and $h1 =_q l3$). The crucial point is that the top labels of both node-RMRSs of the verb nodes would be updated to the label of the *adv* node-RMRS. After this update, both node-RMRSs of the verb nodes would have the same top label, i.e. $l2$ and $l3$ would be changed to $l1$. That is a problem for the coordination of these two structures. When forming a coordination of verbs, the coordination EP has two holes that are qeq to the two top labels of the respective node-RMRSs. If these labels are equal (to $l1$), the coordination would not link two distinct structures. This is invalid and hence the adverb has to outscope the coordination node itself, in valency and in free modifications.

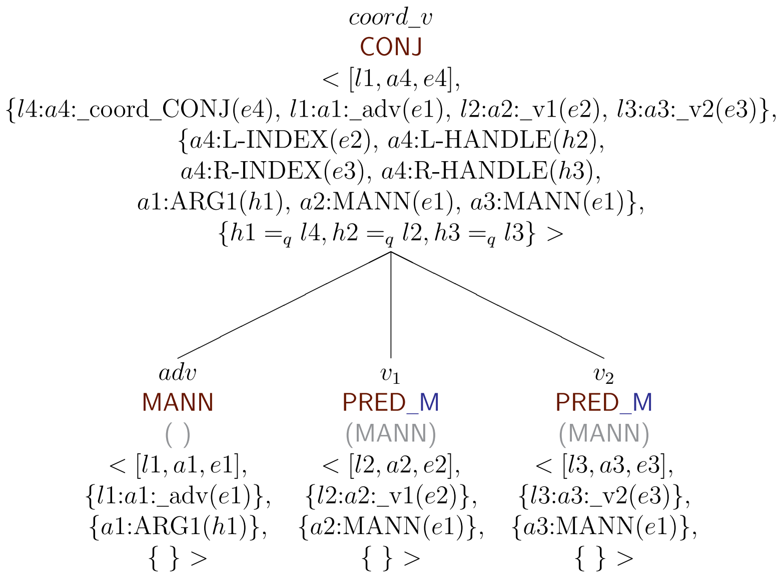

coord_v
CONJ
$< [l1, a4, e4],$
$\{l4{:}a4{:}\_coord\_CONJ(e4),\ l1{:}a1{:}\_adv(e1),\ l2{:}a2{:}\_v1(e2),\ l3{:}a3{:}\_v2(e3)\},$
$\{a4{:}L\text{-}INDEX(e2),\ a4{:}L\text{-}HANDLE(h2),$
$a4{:}R\text{-}INDEX(e3),\ a4{:}R\text{-}HANDLE(h3),$
$a1{:}ARG1(h1),\ a2{:}MANN(e1),\ a3{:}MANN(e1)\},$
$\{h1 =_q l4, h2 =_q l2, h3 =_q l3\} >$
adv
MANN
( )
$< [l1, a1, e1],$
$\{l1{:}a1{:}\_adv(e1)\},$
$\{a1{:}ARG1(h1)\},$
$\{ \} >$
v_1
PRED_M
(MANN)
$< [l2, a2, e2],$
$\{l2{:}a2{:}\_v1(e2)\},$
$\{a2{:}MANN(e1)\},$
$\{ \} >$
v_2
PRED_M
(MANN)
$< [l3, a3, e3],$
$\{l3{:}a3{:}\_v2(e3)\},$
$\{a3{:}MANN(e1)\},$
$\{ \} >$

Figure 3.14: Fictive example PDT tree illustrating rule 2a. Nodes are labeled with the functional role, an index, a functor, marking of coordination membership if applicable and valency frame functors.

3. **After applying rule 1 and/or 2, all candidate nodes become MRS-dependents.**

   Note that this rule also captures direct dependent nodes.

MRS-Dependents of Coordination Nodes

Coordination nodes do not have effective children. For them, a different set of MRS-dependents has to be considered.

4. **coap nodes: all direct dependents of the coordination or apposition are MRS-dependents, except for non-members that are in the valency frame of the members.**

   The members of the coordination get grouped together. Non-members whose functor is in the valency frame of the members are subject to valency modification and will fill the valency position with their index variable (for other coap nodes as direct dependents, the inherited functor of the members is meant here). This is captured by rule 2. They are

therefore connected to the complete structure and need not to be dealt with in another way. The non-members that are not in the valency frame of the members are treated as free modification of either the members or the coordination itself (which is achieved by the same means).
In figure 3.13, the MRS-dependents of the topmost node are v_1, v_2 (both members of the coordination) and *adj* (non-member whose functor is not in the valency frame of the members). The node for n is not included because it is a valency modification of the member nodes.

5. **list nodes: all direct dependents are MRS-dependents.**

   List nodes are treated as having only member nodes, so that all direct dependents are linked together by the rules of coordination. This is a default behavior and is inaccurate for some cases in PDT. Occurrences in which this is not true, either the functors or the functional roles are inconsistent among the direct dependents. This inconsistency will cause a skip in the upcoming experiments (just like inconsistency among members under coap nodes).

No MRS-Dependents of Other Nodes

Other nodes than the just mentioned do not have any MRS-dependents. Nodes representing foreign language expressions and nodes with the atom node type generally only appear as leafs in the data. If they appear as inner nodes, the tree is skipped. Nodes representing idioms (node type dphr) are skipped in general.

Example

Using figure 3.15, examples for several nodes will be outlined next. All leaf nodes in this example do not have any MRS-dependents. The MRS-dependents for all non-leaf nodes are listed below. The node is underlined and its the MRS-dependents appear after a colon. The rules stated above are referred to in the explaining paragraphs.

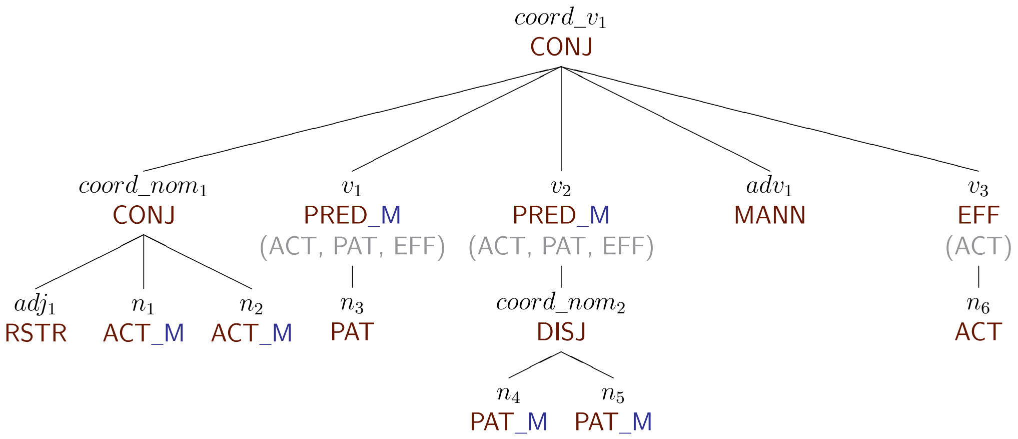


Figure 3.15: Fictive example PDT tree for illustrating MRS-dependents of a node (possible sentence: *These boys and girls saw Jane and heard Peter or Paul yesterday as they came.* (Slavic accusative construction)). The nodes are labeled with their functional role, an index, their functor and _M if they are members of a coordination.

- $\underline{coord\_v_1}$: v_1, v_2, adv_1

  The direct dependents of $coord\_v_1$ are $coord\_nom_1$, v_1, v_2, adv_1 and v_3. The non-member node v_3 is part of the valency frame of the coordinated verbs and therefore not included in the MRS-dependents. Non-member node $coord\_nom_1$ inherits the ACT functor from its members and is therefore also part of that valency frame and not an MRS-dependent. The other direct dependents are MRS-dependents of $coord\_v_1$ (rule 4).

- $\underline{coord\_nom_1}$: adj_1, n_1, n_2

  All direct dependents are MRS-dependents. n_1 and n_2 are members of the conjunction and adj_1 is not in their valency frame (rule 4).

- $\underline{v_1}$: $coord\_nom_1$, n_3, v_3

  The effective children for this node are n_1, n_2, n_3, adv_1 and v_3. n_1 and n_2 appear with the same functor (ACT). Their shared parent coordination node, $coord\_nom_1$, replaces them in the candidate set and inherits the functor (rule 1). $coord\_nom_1$ is not a descendant of v_1, but it is in its valency frame due to the inherited ACT functor. Hence, it remains a candidate (rule 2). v_3 is also not in the same subtree as v_1, but it is in valency position (functor EFF) as well (rule 2). adv_1 is also not a descendant v_1 and it is not in the valency frame. Therefore it is deleted from the candidate list (rule 2). If it *would* be in the valency frame, the special rule 2a for adverbs would apply (handle constraint scopes over the shared parent of the v_2 and adv_1 nodes, which is $coord\_v_1$). n_3 is an effective child in the "direct child" position. It becomes MRS-dependent together with $coord\_nom_1$ and v_3 (rule 3).

- $\underline{v_2}$: $coord\_nom_1$, v_3, $coord\_nom_2$

  See the explanations for $coord\_nom_1$ and v_3 in previous the paragraph. Additionally, n_4 and n_5 are among the candidates for v_2. They both appear with the PAT functor and are joined under a coordination. Hence, $coord\_nom_2$ becomes a candidate (rule 1) and in the next step an MRS-dependent (rule 3).

- $\underline{v_3}$: n_6

  The only effective child of v_3 is n_6. It is in the same subtree as v_3 and is therefore an MRS-dependent (rule 3).

- $\underline{coord\_nom_2}$: n_4, n_5

  Both direct dependent member nodes are MRS-dependents (rule 4).

This example showed which nodes will be processed using the three methods shown in section 3.3.4. The reader can easily verify that all nodes, except for the topmost one, are MRS-dependents to at least one other node. If they are MRS-dependents of multiple nodes, they must be a valency modification. All substructures will therefore be connected in the final target representation.

All necessary information for constructing a mapping algorithm has therefore been established and section 3.5 presents a specific procedure. But at first, the next section lists all features of a tectogrammatical tree that are represented in the target formalism, along with skipped phenomena and lost features.

3.4 Summary of Preserved and Lost Information

Having concluded the description of the correspondence, this section summarizes all information of tectogrammatical trees that is preserved, skipped and lost when representing it as an RMRS.

3.4.1 Preserved Information

This sections lists all properties of a PDT tectogrammatical layer tree that are preserved in the target RMRS representation.

1. structure and dependencies: the structure of the tree is expressed appropriately in RMRS terms through usage of the correct variables in EPs and through qeq-constraints. Some dependency relations are represented in reverse, e.g. EPs for *adv* nodes become the top labels of verbal constructions, instead of being dependent on the verb.

2. `functor`: all functor information is mapped. For valency modifications, the named argument carries the functor. For free modifications, the connecting EP preserves this information. For coordinations, the functor is mapped to the relation name.

3. `subfunctor`: all subfunctor information is preserved through the methods described in item 2.

4. `coref_gram.rf`: for qcomplex nodes with the tectogrammatical lemmas #Cor, #QCor or #Rcp and for grammatemes with inher values the grammatical coreference links are utilized. The annotated information is therefore contained in the target representation.

5. `gram/*`: `gram/sempos` is mapped onto the relation name; all other grammateme values are mapped to variable features

6. `sentmod`: preserved as variable feature

7. `is_member`: only members are linked in coordinations. The annotated information is therefore contained in the target representation.

8. `is_name_of_person`: is a criterion for making a named entity EP. The annotated information is therefore contained in the target representation.
9. `val_frame.rf`: the valency frame information is reflected in the argument positions of the lexical EPs and the number of the frame is mapped to the relation name
10. `t_lemma`: the tectogrammatical lemma is preserved through the relation name of lexical EPs
11. underspecification of adverbial modification scope is preserved

3.4.2 Skipped Phenomena

This sections lists all constructions that cause a PDT tree to be skipped in this project.

1. no nodes in the tree: some trees just have a technical root, e.g. the tree for "...". The semantics of this cannot be captured because there are no tectogrammatical nodes and therefore no EPs.
2. quasi-complex nodes for which the functional role is undefined, i.e. qcomplex nodes with the one of the following tectogrammatical lemmas: #Amp, #Ast, #AsMuch, #Equal, #Some, #Total, #Bracket, #Comma, #Colon, #Dash, #Period, #Period3, #Slash
3. multiple functors for the same valency slot, without a coordination grouping them together (often caused by distinct subfunctors)
4. multiple alternate functors of the same alternation valency slot (often caused by distinct subfunctors)
5. inconsistent functional roles in a coordination
6. inconsistent functors in a coordination
7. *coord_adv*: coordination of nodes with the *adv* functional role
8. *empty* functional role (see section 3.3.2) in non-valency position

9. *empty* functional role (see section 3.3.2) at a non-leaf node
10. *resolve* functional role without annotated `coref_gram.rf`
11. idiom, i.e. node with node type `dphr`

3.4.3 Lost Information

The following node attributes are not represented in the target RMRS structures.

1. correct treatment of quantifier words is not provided; currently treated as non-scoping EPs; quantifier information is therefore not contained in the target structures, although strictly speaking it is is not lost
2. `a`: links to the analytical layer (irrelevant)
3. `compl.rf`: second dependency with predicative complements
4. `coref_special.rf`: special types of textual coreference
5. `coref_text.rf`: textual coreference
6. `coref_gram.rf`: grammatical coreference information that serves other purposes than the functional role *resolve* or inheriting grammateme values is lost
7. *resolve* functional role and more than one item in `coref_gram.rf` → default: first entry is taken as antecedent; additional grammatical coreferences are lost
8. `deepord`: deep word ordering attribute (related to `tfa` attribute)
9. `gram/sempos`: the semantic part-of-speech is not mapped to the relation name for named entities, foreign language expressions and cardinal numbers
10. `id`: node identifier (irrelevant)
11. `is_dsp_root`: marks root of direct speech subtrees; it is unmarekd in the target structures

12. `is_generated`: marks newly established nodes
13. `is_parenthesis`: marks parts of a parenthesis
14. `is_state`: marks modifications with the meaning of state
15. `quot`: marks text segments in quotation marks
16. `tfa`: topic-focus articulation

It is very possible that some of the constructions in sections 3.4.2 and 3.4.3 can be mapped using similar methods as for the successfully mapped phenomena. However, this should be studied and tested by a Czech linguist.

3.5 Implementation

This section outlines the flow of the implemented algorithm that will provide the basis for the evaluation in the next chapter. Straightforward tree processing is not possible, due to the effective child relation that assists in finding the MRS-dependents of nodes. This relation goes beyond the tree property of the PDT representation. Nevertheless, the algorithm is going to construct RMRS structures for all subtrees of a tectogrammatical tree from the leafs to the root node.

3.5.1 Resources

The method that was described in detail in section 3.3 was implemented using the scriptable version of the tree editor TrEd[6] under Perl version 5.10. The data of the Prague Dependency Treebank 2.0 and the valency dictionary PDT-Vallex are necessary as input data. RMRS quadruples in a simple XML format are the output.

3.5.2 Algorithm

[6] http://ufal.mff.cuni.cz/∼pajas/tred/

Algorithm 1 *pdt2rmrs* mapping

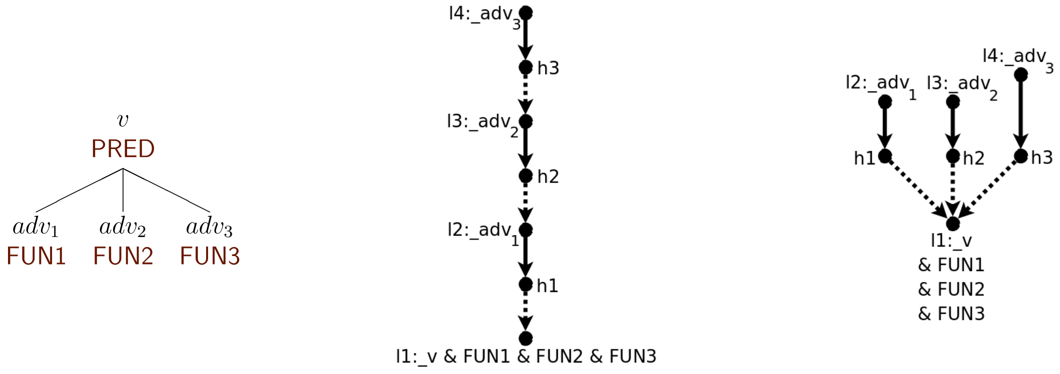
Input: one PDT tectogrammatical tree

Output: one RMRS structure

```
parse PDT-Vallex valency dictionary
sort tectogrammatical nodes bottom-up; equal levels: non-members first
for all tectogrammatical nodes in sorted order do
  current_functional_role ← get functional role of current node
  current_node-RMRS ← initialize node-RMRS for current node based on
  current_functional_role
  sort MRS-dependents: members first; most dominant adv last
  for all MRS-dependents in sorted order do
    dependent_functional_role ← get functional role of dependent
    while dependent_functional_role == resolve do
      dependent ← get grammatical coreference node
      dependent_functional_role ← get functional role of dependent
    end while
    dependent_node-RMRS ← get the node-RMRS of the dependent
    if dependent functor is in valency frame of the current node then
      create valency modification for current_node-RMRSs top anchor
      mark valency functor as processed
    else if current_functional_role starts with coord_*
    and dependent node is a member of the coordination then
      insert the lexical EP of the dependent node into the coordination
    else
      create a connecting EP for the free modification
    end if
    copy EP bag: dependent_node-RMRS ↦ current_node-RMRS
    copy arguments set: dependent_node-RMRS ↦ current_node-RMRS
    copy handle constr.: dependent_node-RMRS ↦ current_node-RMRS
  end for
  if dependent_functional_role == adv then
    current_node-RMRS top label ← dependent_node-RMRS top label
  end if
  for all unmarked valency functors do
    add argument EP filled with u (obligatory) or i (optional) variable
  end for
end for
set the top of the root node-RMRS to a new label
return root node-RMRS
```

The developed algorithm (shown as Algorithm 1) takes a tectogrammatical tree as input and outputs an RMRS. In the first step, the valency dictionary has to be parsed in order to distinguish valency modifications from free modifications. Then all tectogrammatical nodes are sorted in a bottom-up way. This requirement ensures that the hooks are set to the right values when a node-RMRS is processed as an MRS-dependent. Furthermore it guarantees that steps 22 to 24 propagate complete sets, to which no more elements are added afterwards, up the tree. For elements on the same tree level, nodes that are members of a coordination are processed last. The reason is that non-member nodes can be MRS-dependents of the member nodes, even if they are in another subtree. Their complete node-RMRS, therefore, has to be established before it is processed as an MRS-dependent. Otherwise, the governing node-RMRS would not contain the EPs from one level below. In the right tree in figure 2.3 (page 11) the node for vana has to be processed before the two verb nodes that are members of the coordination. As a valency argument of the members, the vana node will not be processed as MRS-dependent of the coordination node. That means that its EP bag will be copied to the verb node-RMRSs when they combine. Therefore, the EP bag of the vana node must be fully built at that point. Otherwise, the RSTR of vana is not copied up to the higher levels. Note that multiple occurrences of the same EP are no problem on higher levels, because the EP bag is a set.

The outer loop (steps 3 to 32), iterating over all nodes in the described order, initializes the node-RMRSs. The inner loop (steps 7 to 25) iterates over all MRS-dependents of the current node. The members are taken first because the hook values have to be set to the topmost coordination EP for modifications to work correctly. Furthermore, the most dominant scoping adverb is taken last. The reason is that the adverbial modification step does not update the top label in each step. This would result in a dominance chain of the adverbs (figure 3.16b), which is not described by the PDT tree. Rather, all adverb holes are qeq to the modified verbs label and one adverb is taken to be the top of the structure in step 27. The current implementation chooses the adverb with the highest deep ordering attribute ($l4$ in figure 3.16c). This should be understood as a technical solution, rather than a means of incorporating information structure knowledge into the target representation.



(a) (b) (c)

Figure 3.16: (a): Fictive PDT tree; (b): undesired MRS graph; (c): desired MRS graph

By the way, figures 3.16a and 3.16c are a good illustration of how the adverbial dependencies are represented in reverse in RMRS.

Steps 9 to 12 follow coreference links if the *resolve* functional role is encountered. That is why in step 13, it is possible that the antecedent node-RMRS has to be initialized, if the coreference is pointing to a higher level. One of the three methods of combining node-RMRSs is then applied to the node-RMRSs of the governing and dependent node in steps 14 through 21. The details were presented in section 3.3.4. Thereafter, the EP bag and handle constraints of the dependent node-RMRS are copied to the governing node-RMRS. Finally, in the outer loop, all valency functors that were not processed, are added as unspecified arguments. The last step of the algorithm is the root condition for complete (R)MRSs. It establishes a top label that is not outscoped by any other label in the resulting RMRS representation.

Chapter 4

Evaluation

The quality of the method for mapping the PDT annotation scheme to the RMRS formalism presented in chapter 3 is evaluated in this chapter. First, the experimental setting is introduced including the definition of a structurally correct RMRS representation. Then, the results are presented and finally an error analysis and result discussion are given.

4.1 Experimental Setup

Manual checking the produced RMRS representations is not possible due to the size of the corpus and the lack of Czech language skills of the author. A different measure of the quality of the produced structures is defined next. It is related to the structural properties of MRSs. For this reason, the RMRS structures have to be converted to MRS representations first. In section 2.2.3, it was established that this process is possible under one condition: unspecified and omitted valency arguments have to be represented explicitly in the RMRS in order for the conversion to MRS to be feasible. The algorithm presented in section 3.5.2 adds EP arguments for unspecified valency arguments. Therefore, the conversion is viable and henceforth, MRS structures will be discussed instead of their equivalent RMRS representations.

4.1.1 Valid MRS Structures

There are two structural conditions that an MRS must fulfill in this experiment in order to be valid.

1. **A valid MRS must be a net.**
2. **A valid MRS must have at least one configuration.**

Net criterion

[Fuchss *et al.*, 2004] argue that the only linguistically relevant MRS structures, in practice, are so called *nets*. This is known as the *net hypothesis* for MRS graphs and was empirically shown to be true. [Flickinger *et al.*, 2005] successfully utilize the net hypothesis for identifying bad grammar rules in the ERG. They further strengthen the hypothesis but also identify a rare class of legitimate MRS graphs that are not nets.

In (4.1), an example MRS is presented and its corresponding MRS graph is shown in figure 4.1. It constitutes a net according to the two conditions mentioned below.

(4.1) *Every cat can eat a mouse.*

$< [l0, e1],$

{ $l1$: _every_q($x1, h1, h2$),
$l2$: _cat_n($x1$),
$l3$: _can_v($e1, h3$),
$l4$: _eat_v($e2, x1, x2$),
$l5$: _a_q($x2, h4, h5$),
$l6$: _mouse_n_1($x2$) },

$\{ h1 =_q l2,\ h3 =_q l4,\ h4 =_q l6 \} >$

The subgraphs of an MRS graph that are connected by solid edges are called the fragments of the graph. An MRS forms a net if each fragment of the MRS graph satisfies the following two conditions ([Flickinger *et al.*, 2005], page 5):

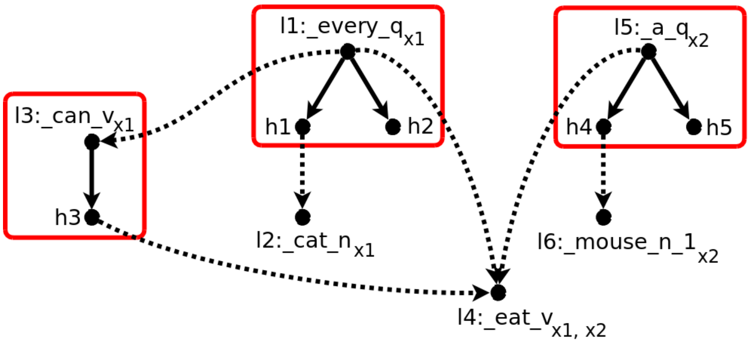


Figure 4.1: MRS graph for the MRS in (4.1). Fragments are bordered.

1. In each fragment, there is exactly one node without outgoing dominance edges (dotted). All other nodes in the fragment have at least one outgoing dominance edge.

2. If a node X has two (or more) outgoing dominance edges, say, to Y and Z, then Y and Z are connected by a *hypernormal path* that does not visit the node X itself. A hypernormal path is an undirected path that does not use two dominance edges that start from the same node ([Flickinger *et al.*, 2005], page 5).

If only nets are linguistically relevant structures, all MRS structures that are produced in this project must be nets. Non-nets are systematically incomplete. They violate one or both described conditions. Figures 4.2 and 4.3 show two classes of non-nets. The "open hole" structure is lacking outgoing dominance edges from exactly one node, either from the root $l1$ or from the rightmost hole $h2$. The "ill-formed island" structure has a node with two outgoing dominance edges but is lacking the hypernormal path that is connecting the two subgraphs under $l3$ and $l6$.

Configuration criterion

If the MRS is a net, it additionally must have at least one configuration. Configurations (or scope-resolved MRSs) correspond to linguistic readings or interpretations. Natural language sentences must have at least one reading, otherwise there is no semantic interpretation, which would make the sentence

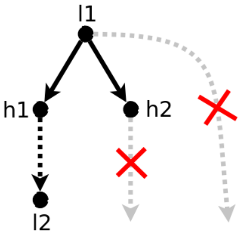


Figure 4.2: Open hole fragment

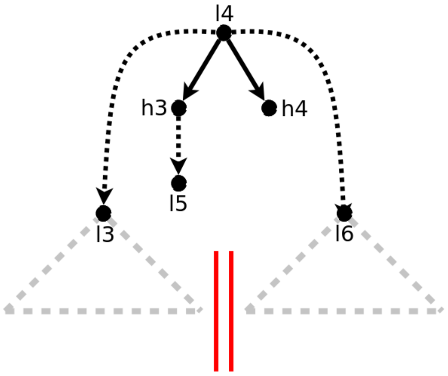


Figure 4.3: Ill-formed island fragment

meaningless. The PDT data describes natural language. Each sentence must therefore have at least one linguistic interpretation. Hence, each MRS must have at least one configuration. While the net hypothesis is more controversial regarding its correctness, the configuration criterion filters out incorrect structures with more certainty. Nevertheless, both criteria must be true in a valid MRS structure.

4.1.2 Procedure

Utool ([Koller *et al.*, 2006]) is a tool that is designed to perform a number of operations that arise when working with underspecified descriptions. It is capable of classifying MRS structures as nets. In fact, Utool translates the MRS description to another formalism (dominance constraints), which can be done only for nets. Furthermore, Utool is able to calculate all configurations for a net. It also implements a more efficient algorithm[1] that determines whether an MRS has any configurations without actually calculating them. Utool accepts MRS representations, amongst a number of other underspecification formalism, as input. Therefore, the RMRS structures that are produced must be converted to MRS structures before the actual evaluation step, which is feasible as already mentioned.

Utool always tests the net criterion before the configuration criterion (for

[1] through the `nochart` option

reasons see [Fuchss *et al.*, 2004]). If the net criterion fails, the structure will be classified as invalid without testing the configuration criterion. If the net criterion is satisfied and the configuration criterion fails, the structure is also invalid. Only a positive outcome of both tests yields a valid MRS structure.

The implemented *pdt2mrs* script tried to map each tectogrammatical tree of PDT to an RMRS representation. Some trees were skipped, as already mentioned in section 3.4.2. Afterwards, these RMRSs were converted to MRS structures. Utool[2] then tested the net criterion and the configuration criterion of every produced MRS structure. For non-nets, one of the error classes shown in figures 4.2 and 4.3 is returned. Other ill-formed MRS structures, e.g. with free variables or format errors, are also detected by the program but did not play a role in this evaluation.

4.2 Results

Table 4.1 displays precision and recall values of 89.70 % and 81.16 % respectively. The former is calculated by dividing the number of nets with at least one configuration by the number of MRS structures predicted. For the latter the divisor is the number of trees in the tectogrammatical layer of the PDT. Table 4.2 shows the result counts for a subcorpus of the PDT. The majority of produced MRSs are nets with configurations, i.e. valid MRS representations. There were little amounts of nets without a configuration and non-nets with ill-formed island fragments and a bigger amount of non-nets with open holes. Table 4.3 lists all encountered reasons for for skipping a tree during the mapping.

[2] version 3.1.1

mapped	44725	90.48 %
skipped	4706	9.52 %
correct	40120	81.16 %
Precison	40120/44725	89.70 %
Recall	40120/49431	81.16 %

Table 4.1: Precision and Recall for valid MRS structures produced from the PDT tectogrammatical layer. The 40120 correctly mapped structures are nets with at least one configuration (see table 4.2).

Nets	have configuration	**40120**
	no configuration	22
Non-Nets	open hole	4547
	ill-formed island	36
	mapped	44725

Table 4.2: Result totals of all mapped tectogrammatical layer trees of PDT

behavior of qcomplex node undefined	1560
behavior of idiom, i.e. dphr node, undefined	1167
coap with inconsistent functional roles among members	1148
coap with inconsistent functors among members	246
multiple assignment of a valency slot	197
list with inconsistent functional roles	174
behavior of *coord_adv* functional role undefined	85
resolve or inher node has no grammatical coreference annotated	77
multiple assignment of an alternate valency slot	24
tree consists only of a technical root	12
empty functional role in non-valency position	10
empty functional role at an inner node	6
skipped	4706

Table 4.3: Reasons for skipping a tree while mapping the PDT tectogrammatical trees onto MRS structures. Section 3.4.2 lists the details of these reasons.

4.3 Discussion

The results show very satisfying performance. A big majority of the mapped structures consists of valid structures. For the other cases, this sections analyzes the source and target structures of all error classes.

4.3.1 No configurations

An underspecified MRS structure represents the set of all possible scope-resolved linkings (or configurations) that can be produced from it. If the handle constraints are inhibiting linkings so that there is not a single configuration, the MRS expresses no linguistic interpretation. Analyzing the concrete representations unfortunately did not yield a common reason for their MRS not to have a configuration. However, over half of the MRSs in question have a node with the *resolve* functional role that is coreferring to a node with the *v* functional role. This is problematic since it was assumed that *resolve* nodes represent control, reciprocity and complex predicates. In all these cases the antecedent of the coreference is typically a nominal object, not a verb.

4.3.2 Open holes

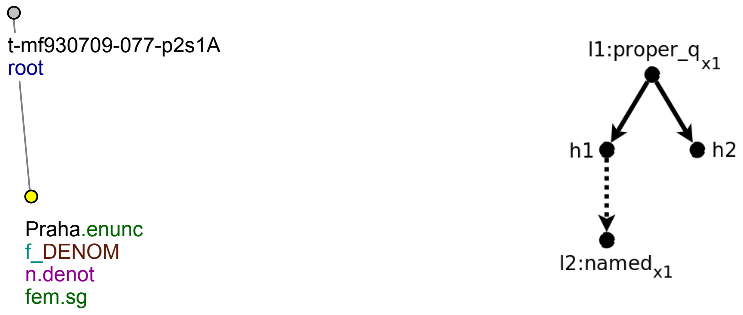
There are 4547 structures with open hole fragments. The analysis of these structures revealed two common characteristics. They involve the linguistic root node of the respective trees, meaning the child of the technical root. None of these linguistic root nodes has the PRED functor[3]. Furthermore, none of these linguistic root nodes has the functional role *v*. Table 4.4 lists all functors and functional roles at these nodes.

With exception of the 8 trees that have the *adv* functional role at the root node, all roots introduces a nominal variable. This variable is not used by any other EP in the EP bag, because the root node is not an MRS-dependent node of any other node. Hence, the graph fragment of the quantifier lacks an outgoing dominance edge, which is not allowed under the net hypothesis. The easiest example for these cases is the city names at the beginning of newspaper articles. The root functor is DENOM and the functional role of the root is *n* for

[3] For coap nodes the functor inherited from the members is meant here.

Functor	Functional Role	
independent nominative clause (DENOM)	*n*	3616
parenthetic clause (PAR)		134
independent vocative clause (VOCAT)		1
independent nominative clause (DENOM)	*coord_n_or_adj*	777
parenthetic clause (PAR)		9
independent interjectional clause (PARTL)	*adv*	8
independent nominative clause (DENOM)	*adj*	1
adjunct of expressing manner (MANN)		1
MRSs with open holes	total	4547

Table 4.4: Functors and functional roles of the linguistic root nodes (child of technical root) of all trees that translate to MRSs with open hole fragments. If the linguistic root is a coap node, the functor is inherited from the members.



(a) (b)

Figure 4.4: (a): Tectogrammatical tree for the sentence string *”Praha -”* (b): corresponding MRS graph

these structures. The EP bag of the corresponding MRS will just contain one EP and one quantifier EP for the introduced nominal variable. The restriction of the quantifier is qeq to the lexical EP. Figure 4.4 shows an example. The fragment has two nodes without outgoing dominance edge, violating the first net condition. Beyond that, if there are more nodes beneath the nominative root in the PDT tree, this problem still persists, since $x1$ will never be used by any other EP.

All these structures are invalid considering the net criterion, but from another point of view it can be argued that sentences without a main verb are in fact incomplete phrases. If the results of the remaining mappings are considered, one can speculate that the trees that do not have the PRED functor at the root are in fact correct partial MRSs. The invalidity of the 4547 MRSs with open holes is at least questionable.

4.3.3 Ill-formed islands

The analysis of the structures with ill-formed island fragments also showed a shared characteristic. All the source trees include a node with the *resolve* functional role. It corefers to one of its own ancestor nodes. The antecedent, in turn, is at least two levels above the *resolve* node. See figure 4.5 for an example subtree. In the corresponding MRS, the variable introduced for the antecedent noun ($x3$) is used at least two times (besides the _divadlo_n.denot EP introducing it): in the governing lexical EP of the antecedent node (_přispisovat_v_1) and in the governing lexical EP of the *resolve* node (_oslovit_v_1). In the corresponding MRS graph, the quantifier node of this variable therefore has two outgoing dominance edges. The targets of these edges must be connected via a hypernormal path, as stated in the second net condition. But for structures with the described subtree characteristic there is no such path, as shown in figure 4.6. Therefore, these PDT trees cannot be properly represented, at least not under the net hypothesis.

However, all these structures fall into the category of ”legitimate non-nets” identified in [Flickinger *et al.*, 2005]. There are two quantifiers that both bind variables in the restriction of the other quantifier. Hence, for these structures, their incorrectness is in question as well.

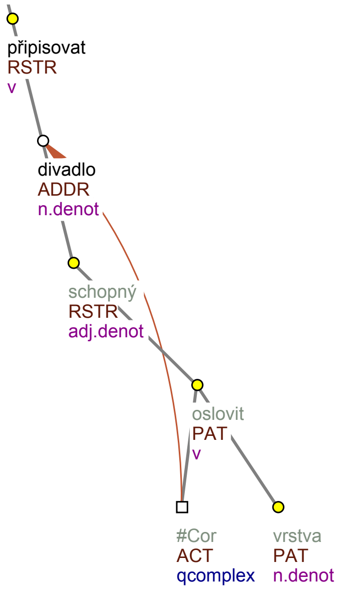


Figure 4.5: Example subtree of the tectogrammatical layer for the substring "*připisovalo divadlu - schopnému oslovit* [...] *vrstvy*". The coreference link originating at the #Cor node refers to one of its ancestors. ACT and PAT of *přispisovat* as well as MAT and RSTR of *vrstva* are omitted for simplicity.

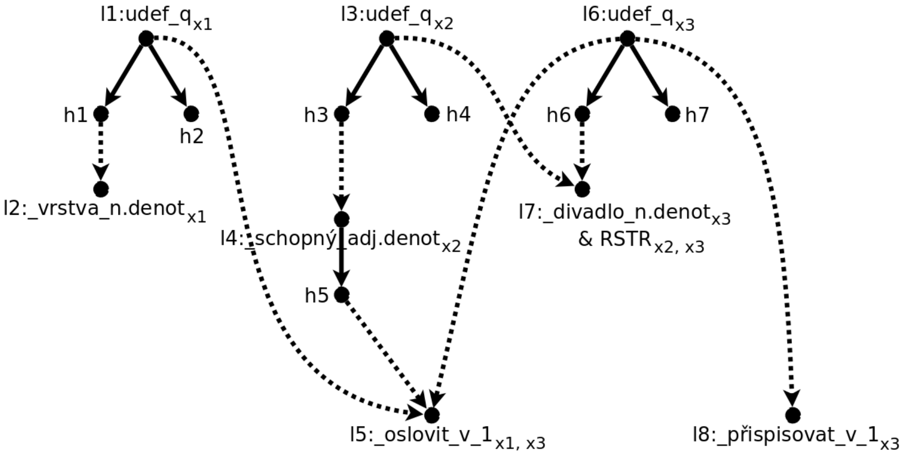

l1:udef_q$_{x1}$
h1
h2
l2:_vrstva_n.denot$_{x1}$
l3:udef_q$_{x2}$
h3
h4
l4:_schopný_adj.denot$_{x2}$
h5
l5:_oslovit_v_1$_{x1, x3}$
l6:udef_q$_{x3}$
h6
h7
l7:_divadlo_n.denot$_{x3}$
& RSTR$_{x2, x3}$
l8:_přispisovat_v_1$_{x3}$

Figure 4.6: MRS graph for the example structure in figure 4.5. The quantifier $l6$ has two outgoing dominance edges to $l5$ and $l8$ but those two EPs are not connected through a hypernormal path. This is an ill-formed island graph fragment.

To sum up the evaluation, the overall performance of the mapping is very satisfying. It proves that this work can be the basis for a useful transformation of the PDT annotation to the MRS formalism. Furthermore, there is reasonable doubt that the structures labeled as non-nets in this evaluation are in fact incorrect structures, as outlined in sections 4.3.2 and 4.3.3. This gives reason to believe that the two involved descriptions are basically compatible. Nevertheless, this evaluation involved only the automatic checking of the structural correctness of the MRS structures. A final judgment of the compatibility has to be made after a Czech linguist took a closer look at the actual represented semantics. At any rate, the amount of phenomena that are not mapped and that are skipped still has to be reduced before the produced MRSs can be used in practice.

Chapter 5

Conclusion

In this work, the task of mapping the dependency based PDT annotation scheme onto the compositional semantics formalism RMRS was considered. For this purpose, the tectogrammatical layer was chosen as source representation. The produced RMRS structures represent the tree structure and dependencies as well as grammatemes and some coreferences of the PDT trees on this layer while trying to find a middle way between the theoretical background of FGD and classical RMRS representations. Elementary predications for lexical information and dependency labels are constructed while distinguishing between valency and free modification. This is done in an iterative manner, so that all nodes have associated RMRS structures that represent the semantics for the tectogrammatical subtree rooted at the respective node. The complex relationship between the linguistic dependence and non-dependency structures (mainly coordination and apposition) in PDT and in RMRS has been analyzed in detail and implemented in the mapping algorithm.

The results show that systematically correct underspecified target structures can be obtained by a rule-based mapping approach. This gives raise to the notion that dependency structures for semantic descriptions can be reformulated as compositional semantic representations, while it is still an open question how much information of the source representation is possible to preserve. Potential benefits of the mapping can be found in several areas of deep processing, extending the range of available resources. Deep parsers utilizing RMRS could be enriched with information from manually annotated dependency trees and machine translation systems can operate on a common representation for both involved languages, just to name a couple. A possible disadvantage of the approach presented here is that the produced structures are too different from the ones produced by other MRS descriptions, for instance, by the ERG, in which the predicates are more closely related to the input tokens. However, the level of abstraction was adapted from the tectogrammatical layer, even though it would be possible to integrate lower layer information and therefore come closer to the ERG design. The developed mapping algorithm can be seen as a basic framework that can be adjusted to specific needs. Nevertheless, the results show that Czech, being typologically different from languages that used RMRS in the past, can be adequately represented in terms of compositional semantics.

5.1 Future Work

Future work on the presented basic research must necessarily be carried on by Czech linguists who have a correct and deep insights into the meaning of the mapped dependency trees. The produced RMRSs might be systematically correct considering their structure, however, they should be manually checked for their semantic correctness by a native speaker familiar with the target description. The next step should be to add lexically licensed quantifiers. Even though Czech has no determiners and therefore a lot of structures can cope with unspecified quantifiers alone, completely correct RMRS structures are given once the correspondences of *every*, *some*, *which* etc. have been added. Furthermore, the number of skipped phenomena has to be reduced, which includes many quasi-complex nodes, coordination of adverbs, idioms and other structures. Also, some of the information that is lost in the current approach can probably be incorporated into the target structures as well with similar methods than the ones used here. For instance, the direct speech node attribute could be captured using an additional EP that outscopes the content of the direct speech. Furthermore, word ordering, as an essential part of the represented free word order language, necessarily has to be included in the future. The lower annotation layers as well as explicit character position representation in RMRS could be exploited for this. Finally, a fully developed mapping algorithm of the reverse direction, RMRS structures to dependency trees, would yield similar advantages as the ones outlined in this work.

With the presented research as a starting point, a precise definition of the relation between compositional underspecification formalisms and dependency descriptions for semantics in natural language processing bears the potential to increase the available resources for followers of both orientations. Furthermore, it is encouraging to closer investigate a formulation of compositional semantics for free word order languages with rich morphology.

Bibliography

[Allen *et al.*, 2007] James Allen, Myroslava Dzikovsk, Mehdi Manshadi, and Mary Swift. Deep Linguistic Processing for Spoken Dialogue Systems. In *Proceedings of the ACL Workshop on Deep Linguistic Processing*, pages 49–56, Prague, Czech Republic, June 2007.

[Bos *et al.*, 1996] Johan Bos, Björn Gambäck, Christian Lieske, Yoshiki Mori, Manfred Pinkal, and Karsten Worm. Compositional Semantics in Verbmobil. In *In Proceedings of the 16th International Conference on Computational Linguistics*, 1996.

[Bos, 1995] Johan Bos. Predicate Logic Unplugged. In *In Proceedings of the 10th Amsterdam Colloquium*, pages 133–143, 1995.

[Copestake and Flickinger, 2000] Ann Copestake and Dan Flickinger. Open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, pages 591–598, Athens, Greece, 2000.

[Copestake *et al.*, 1995] Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. Translation using Minimal Recursion Semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, Leuven, Belgium, 1995.

[Copestake *et al.*, 2005] Ann Copestake, Dan Flickinger, and Ivan A. Sag. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3(4):281–332, 2005.

[Copestake, 20042006] Ann Copestake. Robust Minimal Recursion Semantics (working paper). http://www.cl.cam.ac.uk/∼aac10/papers/, 2004/2006.

[Copestake, 2007a] Ann Copestake. Applying Robust Semantics. In *Proceedings of the 10th Conference of the Pacific Assocation for Computational Linguistics (PACLING)*, pages 1–12, Melbourne, Australia, 2007.

[Copestake, 2007b] Ann Copestake. Semantic composition with (Robust) Minimal Recursion Semantics. In *Proceedings of the ACL Workshop on Deep Linguistic Processing*, pages 73–80, Prague, Czech Republic, June 2007.

[Dridan and Bond, 2006] Rebecca Dridan and Francis Bond. Sentence comparison using Robust Minimal Recursion Semantics and an ontology. In *Proceedings of the ACL Workshop on Linguistic Distances*, page 3542, Sydney, Australia, 2006.

[Egg *et al.*, 2001] Markus Egg, Alexander Koller, and Joachim Niehren. The Constraint Language for Lambda Structures. *Journal of Logic, Language, and Information*, 10:457–485, 2001.

[ERG, 2009] LinGO ERG. *English Resource Grammar (ERG) LOGON On-Line Demonstrator*, 2009. http://erg.delph-in.net/.

[Flickinger *et al.*, 2003] Dan Flickinger, Emily M. Bender, and Stephan Oepen. MRS in the LinGO Grammar Matrix: A Practical User's Guide. http://faculty.washington.edu/ebender/papers/userguide.pdf, 2003.

[Flickinger *et al.*, 2005] Dan Flickinger, Alexander Koller, and Stefan Thater. A new well-formedness criterion for semantics debugging. In Stefan Müller, editor, *Proceedings of the 12th International Conference on HPSG*, 2005.

[Fuchss *et al.*, 2004] Ruth Fuchss, Alexander Koller, Joachim Niehren, and Stefan Thater. Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 247–254, Barcelona, Spain, 2004.

[Goss-Grubbs, 2005] David Goss-Grubbs. An Approach to Tense and Aspect in Minimal Recursion Semantics. Master's thesis, University of Washington, USA, 2005.

[Hajič *et al.*, 2001] Jan Hajič, Barbora Vidová-Hladká, and Petr Pajas. The Prague Dependency Treebank: Annotation Structure and Support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105–114. University of Pennsylvania, Philadelphia, USA, 2001.

[Hajič *et al.*, 2003] Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57–68. Vaxjo University Press, 2003.

[Hajič *et al.*, 2006a] Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová, Václav Klimeš, Jiří Mírovský, Petr Pajas, Jan Štěpánek, Barbora Vidová Hladká, and Zdeněk Žabokrtský. *PDT 2.0 - Guide.* ÚFAL MFF UK, Prague, Czech Republic, 2006. http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/en/pdf/pdt-guide.pdf.

[Hajič *et al.*, 2006b] Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. Prague Dependency Treebank 2.0, 2006. http://ufal.mff.cuni.cz/pdt2.0/.

[Hajič, 2006] Jan Hajič. Complex Corpus Annotation: The Prague Dependency Treebank. In Mária Šimková, editor, *Insight into the Slovak and Czech Corpus Linguistics*. Veda, Bratislava, Slovakia, 2006.

[Koller *et al.*, 2003] Alexander Koller, Joachim Niehren, and Stefan Thater. Bridging the Gap between Underspecification Formalisms: Hole Semantics as dominance constraints. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, page 95202, Budapest, Hungary, 2003.

[Koller *et al.*, 2006] Alexander Koller, Stefan Thater, and Michaela Regneri. *utool: The Swiss Army Knife of Underspecification.* Computational Linguistics at Saarland University, Saarbrücken, Germany, 2006. http://www.coli.uni-saarland.de/projects/chorus/utool/.

[Kruijff, 2001] Geert-Jan M. Kruijff. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic and Information Structure.* PhD thesis, ÚFAL MFF UK, Prague, Czech Republic, 2001.

[Kučová and Hajičová, 2004] Lucie Kučová and Eva Hajičová. Coreferential Relations in the Prague Dependency Treebank. In *Proceedings of the 5th International Conference on Discourse Anaphora and Anaphor Resolution 2004*, pages 97–102, San Miguel, Spain, 2004.

[Mikulová *et al.*, 2006] Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Republic, 2006.

[Pajas and Štěpánek, 2005] Petr Pajas and Jan Štěpánek. A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague Dependency Treebank 2.0. Technical Report 29, ÚFAL MFF UK, Prague, Czech Republic, 2005.

[Panevová, 1994] Jarmila Panevová. Valency Frames and the Meaning of the Sentence. In Philip A. Luelsdorff, editor, *The Prague School of Structural and Functional Linguistics*, pages 223–243. John Benjamins Publishing Company, Amsterdam, Netherlands, 1994.

[Robinson, 1970] Jane J. Robinson. Case, Category, and Configuration. *Journal of Linguistics*, 6(1):57–80, 1970.

[Sgall *et al.*, 1986] Petr Sgall, Eva Hajičová, and Jarmila Panenová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.*

Academia/Reidel Publishing, Prague, Czech Republic/Dordrecht, Netherlands, 1986.

[Wilcock, 2005] Graham Wilcock. Information Structure and Minimal Recursion Semantics. In Antti Arppe, Lauri Carlson, Krister Lindén, Jussi Piitulainen, Mickael Suominen, Marttie Vainio, Hanna Westerlund, and Anssi Yli-Jyrä, editors, *Inquiries into Words, Constraints and Contexts: Festschrift for Kimmo Koskenniemi on his 60th Birthday*, pages 268–277. Stanford, CA: CSLI Publications, 2005.

[Žabokrtský *et al.*, 2008] Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. Tecto MT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *ACL 2008 WMT: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA, 2008. Association for Computational Linguistics.