Faculty of Informatics
Masaryk University
Brno

The Normal Translation Algorithm in Transparent Intensional Logic for Czech

PhD Thesis

Aleš Horák

Brno, November 2001

Declaration:

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Acknowledgements:

I want to give thanks to my supervising professor Karel Pala for his support and feedback during my research and to professor Pavel Materna for his very valuable comments on this work. Special thanks must also go to my colleagues Pavel Smrž and Vladimír Kadlec for their cooperation on the system.

Contents

Chapter 1

Introduction

1.1 Specification of NLP

The analysis of natural language has enticed the attention of many researches since the first realworld computer applications came in existence. The possibility of an automatic processing of the huge amount of texts that people have produced for the purpose of human to human communication is an important addition to the power of computer processing.

In computer linguistics, which has taken the leading role in the field after the direct application of formal grammars and automata has shown to be infeasible for non-artificial languages, most researches agree to splitting the process of analysis into three very basic levels — morphological, syntactic and semantic analysis (see the Figure 1.1). Each of these parts needs to have at its input the results of the previous ones. However, this does not mean that a particular implementation of the analysis must also physically separate these part into different modules, which cooperate only by means of module output/input. In the implementation these parts may even inosculate up to the processing on the particular rules level.

1.1.1 Morphological Analysis

The analysis of morphology is concentrated on single words or at most on some collocations (it is often questionable whether collocations should be analysed as soon as on this level or not). The aim of this stage is to find all the possible grammatical categories of the given word with respect to its ending, stem, prefix and other auxiliary intersegments.

In analytical languages (like English) this part of analysis is usually shifted to the task of POS (part of speech) tagging which is described e.g.

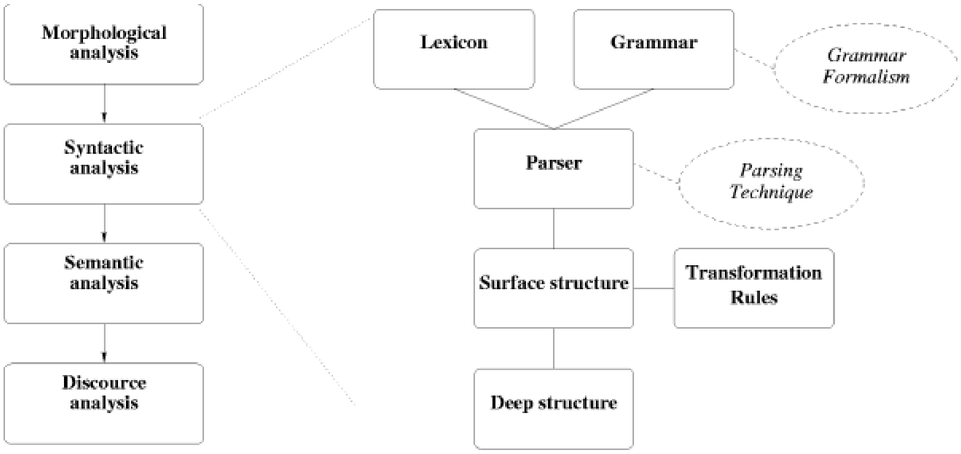


Figure 1.1: General schema of sentence analysis

in Eric Brill's work [Brill95]. His tagger has more than 97% overall accuracy and over 85% of accuracy on unknown words.

Morphological analysis of highly inflectional languages like Czech is a task at a different level of complexity with the number of possible word tags more than 2000 (in comparison to 36 Brill's tags). The commercial program Lemma by Pavel Ševeček [Sevecek96] is an example of a very good morphological analyser that is able to recognize more than 170 thousand stems of Czech words. Nowadays a completely new analyser is being developed at the NLP laboratory at the Faculty of Informatics, Masaryk University, by Marek Veber and Radek Sedláček [Sedlacek99].

1.1.2 Syntactic Analysis

Syntactic analysis of running texts plays a key role in a natural language processing. Many researches have contributed to the area of text parsing by systems that reach satisfactory or even excellent results for English [Sarkar96]. Other languages bring many more obstacles in attempts at creating a systematic description of the language by means of traditional grammars (the situation in German is discussed e.g. in [VolkSch98]). Even more problems are arising in free word order, respectively free constituent order languages. The sentence structure of such language defies to be described by a maintainable number of rules. The order of sentence constituents is designated as free, but as a matter of fact the order is driven more by a human intuition than by firmly set regulators specified by linguists. The word order plays an important role in communicative dynamism, it expresses the sentence focus. This phenomenon is intensively explored by Prague Linguistic School [HajSgSk94] in the context of Functional Generative Description.

There are many grammar formalisms for representing the natural language phenomena employing various kinds of feature structures. The most popular ones are probably HPSG [PollSag94], LFG [KapBres82] and LTAG [Schabes88]. These formalisms are well known for their linguistic adequacy, however, they suffer from problems of high complexity of parsing algorithms. To overcome this difficulty we have introduced the concept of meta-grammar [SmrHor99] that is constructed on a context free grammar backbone which enables us to use a CFG parser modified for the purpose of feature agreement fulfilment and other linguistic tests and actions essential for parsing a free word order language like Czech.

1.1.3 Semantic Analysis and Knowledge Representation

The last stage of the analysis of natural language texts suffers from a permanent problem — the adequate specification of its subject matter. To be able to capture the meaning of the discourse we first have to recognize the meaning resolution process in the human brain, for which we still lack an exact description. Thus the success of the analysis of semantics mainly lies in searching the right representations of sentence meaning for a specified purpose.

Several approaches to semantic analysis have appeared in the course of history (already since the Aristotlean times). Many authors in computationally oriented semantics work with the assumption that knowledge of the meaning of a sentence can be equated with knowledge of its truth conditions: that is, knowledge of what the world would be like if the sentence were true [Pulman96]. Traditionally, the first order predicate logic was used for the semantic description of language. As Montague [Montague74, Montague73] showed, this logic system is able to capture an important range of the constructs but the range of valid constructs in natural language is far wider. Montague and his followers have tried to overcome this weakness. However, as Tichý showed in his book [Tichy88], the Montague Semantics can run into severe problems when analysing certain kind of sentences, which are commonly used in natural language. That is why *transparent intensional logic* was designed to represent semantic structure of the language by constructions.

In the last decades many researches have been involved in searching the best knowledge representation formalism that would fulfil the need for a well behaving surrogate of the real world objects and situations. The approaches range from encoding the knowledge in some strictly specified procedures in procedural knowledge and logic representation through semantic networks and frames up to statistical and probabilistic knowledge (see [LugStub98]). However none of these approaches can claim itself to solve all the difficulties that are brought about with objects described by means of natural language.

The most often discussed conceptions are various alternatives of the logic approach, mainly due to its declarativeness. The central problem of describing the real world by means of logic lies in the low expressivity of the predicate logic. Thus, if we want to use logic for representing natural language knowledge, we need some more enhanced and sophisticated variants of logic.

TIL, or Transparent Intensional Logic, similarly as Montague Semantics, follows Frege's principle of compositionality, viz. *"The meaning of a sentence is a function of the meanings of its constituents"* [Frege1892]. The basic idea of TIL lies in the presupposition that every well-defined language displays a definite intensional base which can be explicated by an "epistemic" framework. Tichý uses an unspecified epistemic framework with objectual base E which is a set of four types that form the basis of type hierarchy. Every entity that can be discussed in a natural language has its equivalent of the appropriate type over the base E. The TIL object that represents the entity described by the analysed expression is denoted not by some sort of name but rather as a construction of the object. The construction records relations among elementary parts of the discourse (words or word groups with a special meaning as a whole). That is why constructions can be advantageously used for expressing the semantics of natural language. More information and discussion on TIL is further presented in the Chapter 4.

1.1.4 Pragmatics

The analysis of a discourse usually employs additional complex actions for processing suprasemantic information, called *pragmatic information.* Pragmatics is concerned with whatever information is relevant, over and above the linguistic properties of a sentence, to understanding its utterance. The distinction between semantics and pragmatics has served to separate strictly linguistic facts about utterances from those that involve the actions, intentions, and inferences of language users (speaker and hearers). However, there are some linguistic phenomena that seem to traverse the semantics-pragmatics boundary like the word ambiguity vs. polysemy or anaphoric and cataphoric relations.

The definition of pragmatics can be accommodated on the supposition that semantic information pertains to linguistic expressions, whereas pragmatic information pertains to utterances and facts surrounding them – the *communication situation*. The analysed semantic information about sentences is a part of sentence grammar, and it includes information about expressions whose meanings are relevant to the use rather than to truth conditions. Linguistically encoded information can pertain to how the present utterance relates to the previous, to the topic of the present utterance, or to what the speaker is doing. The business of sentence semantics cannot be only confined to giving the proposition it expresses. There are also sentences which do less than express propositions, because they are semantically incomplete.

Pragmatic information concerns the facts relevant to the meaning of a speaker's utterance of a sentence (or other expression). The hearer thereby needs to identify the speaker's intention in making the utterance. In effect, the hearer seeks to explain the fact that the speaker said what he said, in the way he said it. Because the intention is communicative, the hearer's task of identifying it is driven partly by the assumption that the speaker intends to do this. The speaker succeeds in communicating if the hearer identifies his intention in this way, for communicative intentions are intentions whose "fulfilment consists in their recognition" [BachHar79, p. 15]. Pragmatics is often divided to *external* (referring to the communication situation, expressed with words with deictic function, indexic words) and *internal* pragmatics (the users attitudes to the propositional content).

1.2 The Objectives

The work described in this thesis is a part of the project of the Normal Translation Algorithm (NTA) and the TIL Inference Machine (TIM). The aim of NTA is to describe the process of translating natural language sentences to constructions of TIL for the purpose of consecutive logical inference performed by TIM. The implementation of NTA can be divided into two main phases — the syntactic analysis with the output in the form of syntactic derivation trees and the logical analysis of the sentence/tree by means of TIL constructions.

In the thesis we describe the implementation of the first part of NTA for the Czech language as well as the detailed algorithm of the logical analysis of Czech sentence. We also provide the implementation of the second part for a selected subset of Czech sentences. Lastly, we briefly discuss the design of the representation of the knowledge base that is used in the inference mechanism TIM which is being implemented by Leo Hadacz within his PhD thesis.

One of the main results of this work is a general purpose syntactic analyser for Czech language. The exploitation of such language parser can range from machine translation, grammar checking, automatic verb valency acquisition or disambiguation of morphological tags up to a speech prosody segmentator for the purpose of high level speech synthesis. Currently, there is only one productive syntax analyser of Czech, the Vladislav Kuboň's parser [Kubon99], which puts to use a certain kind of procedural grammar. It is based on a formalism called RFODG (Robust Free-Order Dependency Grammar) developed in the LATESLAV project [Platek96]. The system encompasses about 50 complex rules for sentence syntax specification written in a Pascal-like form. The main deficiency of this parser is its unavailability outside the author's team, probably due to its possible commercial reuse.

In contrast to the procedural grammar approach, we constitute a grammar system that retains simplicity of rules and thus is a show-case of declarativeness. Herewith the maintenance of the set of grammar rules is kept under an acceptable limit, so that the modifications can be performed even by those users who do not need to have the perfect knowledge of all the internals of the grammar.

Another very important result of our work is the possibility of provision of input data (based on common natural language sentences) to the inference machine based on transparent intensional logic. Such a work can be a significant step towards natural human computer interaction.

Chapter 2

Brief Survey of Sentence Analysis Techniques

In this chapter, we will have a look at the state of the art techniques of the sentence analysis on the syntactic and semantic level.

The syntactic analysis of NL sentences, or shortly *parsing*, can be viewed at from two separate points — the point of chosen grammar formalism and the actual parsing technique used.

2.1 Grammar Formalisms

Here, we will mention three grammar formalisms that are the most frequently used formalisms engaged in the syntactic analysis of natural language sentence. These formalisms are shortly denoted as HPSG, LFG and LTAG.

2.1.1 Head-Driven Phrase Structure Grammar

Instead of transformational derivations (the sequential manipulation of complete sentential structures commonly assumed in linguistic analysis), Head-Driven Phrase Structure Grammar (HPSG) is formulated in terms of order-independent constraints. These constraints provide partial grammatical information that can be flexibly combined in a variety of language processing models.

In HPSG the notion of phrase structure is built around the concept of a lexical head — a single word whose dictionary entry specifies information that determines crucial grammatical properties of the phrase it projects.

This includes part of speech (POS) information (nouns project noun phrases, verbs project sentences, etc) and dependency relations (all verbs require subjects in English, but verbs differ systematically as to whether they select direct object complements, clause complements, and so forth). Lexical heads also encode key semantic information that is shared with their phrasal projections.

Although entries in HPSG are information-rich, the detailed lexical entries of HPSG are expressed within a multiple inheritance hierarchy. Such hierarchical lexicons allow generalizations about words to be expressed in an efficient and compact organization.

The general theoretical background of current work in HPSG is presented in considerable detail in [PollSag94].

2.1.2 Lexical Functional Grammar

The LFG formalism is strictly based on the lexicalized form of the grammar, i.e. it works with a complex lexicon of entries like

Mary, Noun
$(\uparrow \textsc{pred}) = $ '{meaning of Mary}'
$(\uparrow \textsc{num}) = -\textsc{pl}$
$(\uparrow \textsc{gend}) = +\textsc{fem}$

In LFG, there are two parallel levels of syntactic representation: constituent structure (c-structure) and functional structure (f-structure).

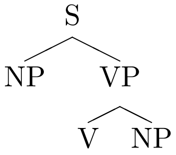
- C-structures have the form of context-free phrase structure trees.
- F-structures are sets of pairs of attributes and values; attributes may be features, such as tense and gender, or functions, such as subject and object. Its role is to integrate the information from c-structure and from the lexicon. While c-structure varies across languages, the f-structure representation, which contains the necessary information for the semantic interpretation of an utterance, is claimed to be universal.

The name of the theory emphasizes an important difference between LFG and the Chomskyan tradition from which it developed: many phenomena are thought to be more naturally analysed in terms of grammatical functions as represented in the lexicon or in f-structure, rather than on the level of phrase structure. An example is the alternation between active and passive, which rather than being treated as a transformation, is handled in the lexicon. Grammatical functions are not derived from phrase structure configurations, but are represented at the parallel level of functional structure.

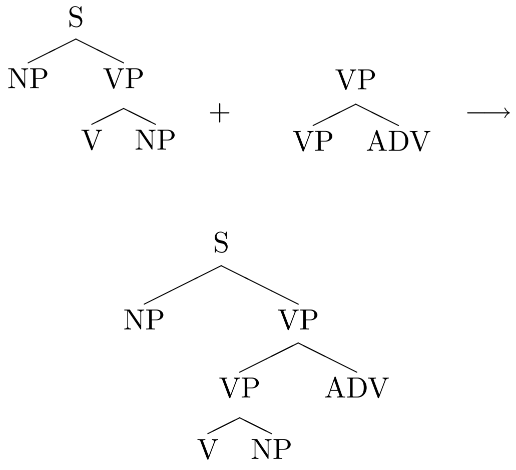
For further reading on LFG with its relations to other grammar formalisms [KapBres82] and [Neidle94] can be recommended.

2.1.3 Lexicalized Tree Adjoining Grammar

The TAG formalism does not encompass any rules — rather, there is a set of initial tree structures that describe the simplest sentences of the language, and a tree operation, called *adjoining*, that inserts one tree into another to create a more complex structure. For example, a simple initial tree is



More complex sentences are derived using auxiliary trees, which capture the minimal forms of recursion in the language. The adjunction operation involves inserting an auxiliary tree that recurses a constituent C into another tree that contains a constituent C. Adjoining the auxiliary tree that allows an adverbial in a verb phrase into the initial tree for S produces the new tree



‘Lexicalized’ grammars (see [Schabes88]) systematically associate each elementary structure with a lexical anchor. This means that in each structure there is a lexical item that is realized. We say that a grammar is *lexicalized* if it consists of

- a finite set of structures each associated with a lexical item, and
- an operation or operations for composing the structures

Each lexical item is called the *anchor* of the corresponding structure, which defines the domain of locality over which constraints are specified. The constraints are local with respect to their anchor. In the process of lexicalizing a grammar, the lexicalized grammar is required to be strongly equivalent to the original grammar, i.e. it must produce not only the same language, but the same structures or tree set as well.

2.2 Parsing Techniques

The early parsing techniques were tailor-made to the analysis of artificial (programming) languages and their main benefits lay in parsing speed. However, analysis of natural language using these techniques brings vigorous difficulties in the form of massive ambiguity of employed grammars.

In next paragraphs we present the basic ideas of two recent techniques that were devised with respect to the natural language processing.

2.2.1 Generalized LR Analysis

The GLR algorithm (introduced by Masaru Tomita [Tomita86]) is an extension of the standard LR parsing algorithm in order to efficiently handle the non determinism occurring due to the conflicting entries in the LR parsing table.

For grammars which are not LR, the parse table will have multiple action entries, corresponding to the shift/reduce and reduce/reduce conflicts. Tomita's algorithm operates by maintaining a number of parsing processes in parallel, each one behaving basically as a standard LR parser. All the parsing processes share a common graph-structured stack, the key structure for the efficiency of the algorithm.

The root (node 0) acts as a common bottom of the stack for all the processes. Each vertex of the graph represents an element (a parse state) in some of the stacks merged in the structure and each leaf of the graph corresponds to an active parsing process, and acts as its top of stack.

All processes are synchronized and behave in the following manner. On each input token, each process acts like a standard LR parser, guided by the parse table. When a process encounters a conflict in the parse table, it splits into several processes, one for every action. When two or more processes are in the same state (have the same state in their top of stack), they will exhibit the same behavior until the state is popped by some reduce action. To avoid this repetition, those processes are unified into a single one, by merging their stacks. When a process executes a reduce action, all the nodes in between its top of stack and the ancestor corresponding to the first symbol of the production must be popped. Since a vertex may have multiple parents, some reduce actions can be done on several paths in the graph-structured stack. When this happens, the process is again split into separate processes. The algorithm begins with a single initial process and follows the procedure above until all processes die, which signifies rejection, or until a process reaches an acceptance state.

The parse trees generated by the different processes are also merged in a *packed shared parse forest*. Sharing refers to the fact that if several trees have a common sub-tree, the sub-tree is represented only once in the forest, and packing refers to the fact that the top vertices of sub-trees that represent local ambiguity are merged and treated as if there were only one vertex.

2.2.2 Chart Parsing Techniques

These techniques establish the notion of a *chart*, a mechanism enabling a parser to keep a record of substructures it has already found, to avoid having to look for them again. In addition, it records information about goals it has adopted, whether unsuccessful or still under exploration.

The basic non-deterministic chart parsing algorithm can be expressed in the following way

```
function nondet-Chart-Parse($words$, $grammar$) returns $chart$
  INIT: top-down:    $chart \leftarrow [0, 0, S' \rightarrow \bullet S]$
  or    bottom-up:   for $i \leftarrow 1..\#words$ and $words_{[i]}$ of category $B$
                     add $[i-1, i, B \rightarrow words_{[i]}\bullet]$ to $chart$
while new edges can still be added do
  $edge \leftarrow$ choose $[i, j, A \rightarrow \alpha \bullet B\beta]$ in $chart$
  choose one of the three methods that will succeed:
    PREDICT:         choose $(B \rightarrow \gamma)$ in RULES[$grammar$]
                     add $[j, j, B \rightarrow \bullet\gamma]$ to $chart$
    COMPLETE:        choose $[j, k, B \rightarrow F\bullet]$ in $chart$
                     add $[i, k, A \rightarrow \alpha B \bullet \beta]$ to $chart$
    SCAN:            if $words_{[j+1]}$ is of category $B$ then
                       add $[j, j+1, A \rightarrow \alpha B \bullet \beta]$ to $chart$
end
return $chart$
```

Before parsing can begin the chart data structure is initialized to contain edges for every word from the input (for the more often used bottom-up strategy) or for the starting non-terminal (for the top-down strategy). *Edge* is a tripple of form $[i, j, A \leftarrow \alpha \bullet \beta]$, which expresses that the parser has recognized the α part in the input ranging from position i to j. If the β part is empty, the edge is called *inactive* (or *closed*), otherwise it is *active* (or *opened*).

The order in which the new edges are processed guides the efficiency of the whole parsing — the optimal parsing needs to produce no partial parses that do not reach the starting symbol. To fulfil this requirement the bottom-up strategy is supplemented with edge selecting methods called *top-down filtering*, *head driven* chart parsing or *left/head-corner* chart parsing (see [SikkAkk96]).

2.3 Semantics

The development of contemporary semantic theories may be viewed at as motivated by the deficiencies that are uncovered when one tries to take the first order predicate logic (FOPC) example further as a model for how to deal with natural language semantics. For example, the technique of associating theoretic denotations directly with syntactic units is clear and straightforward for the artificial FOPC example. But when a similar program is attempted for a natural language, whose syntax is vastly more complicated, the statement of the interpretation clauses becomes in practice extremely unwieldy.

For this reason, in most semantic theories and in all computer implementations, the interpretation of sentences is given indirectly. A syntactically disambiguated sentence is first translated into an expression of some artificial logical language, where this expression is given an interpretation by rules analogous to the interpretation rules of FOPC. This process factors out the two sources of complexity whose product makes direct interpretation cumbersome: reducing syntactic variation to a set of common semantic constructs; and building the appropriate objects to serve as interpretations.

2.3.1 Dynamic Semantics

Dynamic Semantics approach (e.g. [GroeStok91]) reflects the opinion that the standard truth-conditional view of sentence meaning deriving from the paradigm of FOPC does not do sufficient justice to the fact that uttering a sentence changes the context it was uttered in. Deriving inspiration from the work on the semantics of programming languages, dynamic semantic theories have developed several variations on the idea that the meaning of a sentence is to be equated with the changes it makes to a context.

Update Semantics approaches have been developed to model the effect of asserting a sequence of sentences in a particular context. In general, the order of such a sequence has its own significance. A sequence like:

Someone's at the door. Perhaps it's John. It's Mary!

is coherent, but not all permutations of it would be:

Someone's at the door. It's Mary. Perhaps it's John.

Dynamic predicate logic extends the interpretation clauses for FOPC (or richer logics) by allowing assignments of denotations to subexpressions to carry over from one sentence to its successors in a sequence. This means that dependencies that are difficult to capture in FOPC or other non-dynamic logics, such as that between *someone* and *it* in:

Someone's at the door. It's Mary.

can be correctly modeled, without sacrificing any of the other advantages that traditional logics offer.

2.3.2 Discourse Representation Theory

Discourse Representation Theory (DRT), see [Kamp81], as the name implies, has taken the notion of an intermediate representation as an indispensable theoretical construct, and, as also implied, sees the main unit of description as being a discourse rather than sentences in isolation. One of the things that makes a sequence of sentences to constitute a discourse is their connectivity with each other, as expressed through the use of pronouns (anaphora) and ellipsis or similar devices. This connectivity is mediated through the intermediate representation, however, and cannot be expressed without it. The kind of example that is typically used to illustrate this is the following:

A computer developed a fault.

A simplified first order representation of the meaning of this sentence might be

```
exists(X,computer(X) and develop a fault(X))
```

There is a computer X and X developed a fault. This is logically equivalent to

```
not(forall(X,not(computer(X) and develop a fault(X))))
```

It is not the case that no computer developed a fault. However, whereas the first sentence can be continued with

A computer developed a fault.
It was quickly repaired.

its logically equivalent one cannot be

It is not the case that no computer developed a fault.
It was quickly repaired.

Thus the form of the representation has linguistic consequences. DRT has developed an extensive formal description of a variety of phenomena such as this, while also paying careful attention to the logical and computational interpretation of the intermediate representations proposed. Kamp's works contain more detailed analyses of the aspects of noun phrase reference, propositional attitudes, tense and aspect, and many other phenomena.

2.3.3 Transparent Intensional Logic

TIL, or the transparent intensional logic, is a logical system, suitable (and designed) for representing meaning of a natural language expression. The system was introduced by Pavel Tichý as a parallel to Montague's logic, where TIL has the power of greater expressivity while retaining the simplicity of the basic idea. Moreover, the inference rules for TIL are well defined, thus enabling us to use constructions as an instrument for representing sentence meaning in a knowledge base system.

In this work, we have devoted a whole chapter[1] to a detailed survey of the conception of the transparent intensional logic. In that chapter, we first offer a description of the basic ideas of TIL as well as an argumentation about the advantages TIL has brought in comparison with other systems of the logical analysis of natural language. After that, we summarize the basic definitions of the main notions of TIL like type, construction or concept.

2.4 Knowledge Representation and Reasoning

If any intelligent entity (we) wishes to reason about something, it has to identify the objects of the reasoning. The problem is that the objects themselves exist only externally and the reasoning is an internall process of the intelligent entity. Inevitably, we need an appropriate delegate for the external objects in the internal environment, a suitable *knowledge representation*.

2.4.1 Procedural Knowledge

Procedurally based techniques are frequently used in database query applications, where there is a large difference in expressive power between the logical form language and the database language. In the terms of knowledge representation (KR) theory the knowledge base (here the database) consists only of the positive literals that are often without any variables. In such case it is more feasible to treat the logical forms as expressions in a query language. Each logical form language construct corresponds to a particular procedure that performs the appropriate query. For example, the query

Has every employee in department "D" received his wages this month?

---

[1] see the Chapter 4.

with the logical form

$$(\forall e.(\mathsf{Employee}\ e)\ \wedge\ (\mathsf{Department}\ e\ (\mathsf{name}\ \text{"D"}))\ \wedge\ (\mathsf{Wages}\ e\ (\mathsf{date}\ d))\\ \wedge\ (\mathsf{Month}\ d\ \text{ThisMonth}))$$

would be interpreted as a procedure like

1. Find all employees in department "D"
2. For each employee found, check if the month of his/her last wages is ThisMonth. If so, return yes; otherwise return no.

More information about procedural semantics and how to interpret logical form expressions as procedures can be found in [Allen94], the Chapter 13.

2.4.2 Semantic Networks and Frames

A semantic network is a graph, where the nodes in the graph represent concepts, and the arcs represent binary relationships between concepts. The most important relations between concepts are subclass relations between classes and subclasses, and instance relations between particular objects and their parent class. Any other relations are also allowed, such as has-part, colour etc.

The subclass and instance relations may be used to derive new information which is not explicitly represented, through the process of inheriting the information from parent classes. Semantic networks normally allow efficient inheritance-based inferences using special purpose algorithms.

Frames are a variant of nets that are one of the most popular ways of representing non-procedural knowledge in an expert system. All the information relevant to a particular concept is stored in a single complex entity, called a frame. Superficially, frames look like record data structures with the support for inheritance. They are often used to capture knowledge about typical objects or events, such as a typical bird, or a typical restaurant meal. More details about frames can be found in [LugStub98].

2.4.3 Knowledge Base of the TIL Inference Machine

The design of representation of TIL objects in computer takes its inspiration from an implemented system Adam, that is described in [Chrz84a, Chrz84b]. Adam's ontology came out of early versions of TIL, that had not used time in verb constructions and did not allow to use objects with higher order types, as it is currently in the extended type hierarchy. Even without these enhancements the system ADAM was able to catch quite complicated phenomena of natural language like the difference between implication and conditional.

The conception and implementation of the knowledge base of the TIL inference machine is an undetachable part of the prepared PhD thesis of Leo Hadacz, the work [Hadacz2001]. However, we will present a brief description of the designed knowledge base (with examples) and a short introduction to the inference mechanism of TIL. For the disccussion on these topics, the reader may see the Chapter 6.

Chapter 3

Description of the Parsing System

The system description can be divided into two main parts: the description of the employed (meta-)grammar in the next section and the description of the parser in the Section 3.2.

3.1 Grammar Forms

We bring into play three successive grammar forms. Human experts work with the meta-grammar form, which encompasses high-level generative constructs that reflect the meta-level natural language phenomena like the word order constraints, and enable to describe the language with a maintainable number of rules. This is the only part of the system which needs to be adapted when enlarging the phenomena set covered by the parsing system. The two other grammar forms are more a technical tool than an input that should be manually altered. They are generated automatically after the basic meta-grammar changes and all the information in them is deduced from the code contained in the source.

The meta-grammar serves as a base for the second grammar form which comes into existence by expanding constructs used in rule description. This grammar consists of context-free rules equipped with feature agreement tests and other contextual actions, which are translated from the meta-grammar code. This grammar can be already used for parsing, however we are testing the possibility of shifting the task of basic grammatical tests from slow post processing actions to faster and efficient CF rules.

Thus the last phase of grammar induction lies in the transformation of the feature agreement tests into standard rules of the expanded grammar with the actions remaining to guarantee the contextual requirements.

3.1.1 Meta-grammar (G1)

The meta-grammar consists of global order constraints that safeguard the succession of given terminals, special flags that impose particular restrictions to given non-terminals and terminals on the right hand side and of constructs used to generate combinations of rule elements.

Rule flags. The notation of the flags can be illustrated by the following examples:

```
ss -> conj clause

/* budu muset cist - I will have to read */
futmod --> VBU VOI VI

/* byl bych byval - I would have had */
cpredcondgr ==> VBL VBK VBLL

/* musim se ptat - I must ask */
clause ===> VO R VRI
```

We use the arrow in the rule for specification of the *rule type*. A little hint to the arrow form meaning can be expressed by 'the thicker and longer the arrow the more (complex) actions are to be done in the rule translation'.

The thin short arrow (`->`) denotes an ordinary CFG transcription. To allow discontinuous constituents, as is needed in Czech syntactic analysis, the long arrow (`-->`) supplements the right hand side with possible intersegments between each couple of listed elements. The intersegments here represent the sentence constituents that are allowed to fill *gaps* in the verb phrase. The thick long arrow (`==>`) adds (in addition to filling in the intersegments) the checking of correct enclitics order. This flag is more useful in connection with the `order` or `rhs` constructs discussed below. The thick extra-long arrow (`===>`) provides the completion of the right hand side to form a full clause. It allows the addition of intersegments in the beginning and at the end of the rule, and it also tries to supply the clause with conjunctions, etc.

Global order constraints. The *global order constraints* represent universal simple regulators, which are used to inhibit some combinations of terminals in rules.

```
/* jsem, bych, se - am, would, self */
%enclitic   = (VB12, VBK,  R)

/* byl, cetl, ptal, musel - was, read, asked, had to */
%order VBL  = {VL,   VRL,  VOL}

/* byval, cetl, ptal, musel - had been, read, asked, had to */
%order VBLL = {VL,   VRL,  VOL}
```

In this example, the `%enclitic` specifies which terminals should be regarded as enclitics and determines their order in the sentence. The `%order` constraints guarantee that the terminals `VBL` and `VBLL` always go before any of the terminals `VL`, `VRL` and `VOL`.

Combining constructs. The main *combining constructs* in the metagrammar are `order()`, `rhs()` and `first()`, which are used for generating variants of assortments of given terminals and non-terminals.

```
/* budu se ptat - I will ask */
clause ===> order(VBU,R,VRI)

/* ktery ... - which ... */
relclause ===> first(relprongr) rhs(clause)
```

The `order()` construct generates all possible permutations of its components. The `first()` and `rhs()` constructs are employed to implant content of all the right hand sides of specified non-terminal to the rule. The `rhs(N)` construct generates the possible rewritings of the non-terminal `N`. The resulting terms are then subject to standard constraints and intersegment insertion. In some cases, one needs to force a certain constituent to be the first non-terminal on the right hand side. The construct `first(N)` ensures that `N` is firmly tied to the beginning and can neither be preceded by an intersegment nor any other construct. In the above example, the `relclause` is transformed to CF rules starting with `relprongr` followed by the right hand sides of the non-terminal `clause` with possible intersegments filled in.

List expressions. In the current version, we have added two generative constructs and the possibility to define rule templates to simplify the creation and maintenance of the grammar. The first construct is formed by a set of `%list_*` expressions, which automatically produce new rules for a list of the given non-terminals either simply concatenated or separated by comma and co-ordinative conjunctions:

```
/* (nesmim) zapomenout udelat -
   (I have not) to forget to do */
%list_nocoord vi_list
vi_list -> VI

%list_nocoord_case_number_gender modif
/* velky cerveny - big red */
modif -> adjp

/* krute a drsne - cruelly and roughly */
%list_coord adv_list
adv_list -> ADV

%list_coord_case_number_gender np
/* krasny pes - beautiful dog */
np -> left_modif np
```

The endings `*_case`, `*_number_gender` and `*_case_number_gender` denote the kinds of agreements between list constituents. The incorporation of this construct has decreased the number of rules by approximately 15%.

RHS grouping. A significant portion of the grammar is made up by the verb group rules. Therefore we have been seeking for an instrument that would catch frequent repetitive constructions in verb groups. The obtained addition is the `%group` keyword illustrated by the following example:

```
%group verb={
    V:head($1,intr)
      add_verb($1),
    VR R:head($1,intr)
         add_verb($1)
         set_R($2)
}
```

```
/* ctu - I am reading */
/* ptam se - I am asking */
clause ====> order(group(verb),vi_list)
```

Here, the group `verb` denotes two sets of non-terminals with the corresponding actions that are then substituted for the expression `group(verb)` on the right hand side of the `clause` non-terminal.

Rule templates. Many rules, e.g. those prescribing the structure of a `clause`, share the same rule template — they have the same requirements for intersegments filling and the enclitics order checking as well as the right hand side term combinations. To avoid the exigency of repeated usage of the same arrow operator and the `order` construct, we provide the *template* mechanism — instead of the rules

```
/* budu cist - I will read */
clause ====> order(VBU,VI)

/* budu se ptat - I will ask (reflexive) */
clause ====> order(VBU,R,VRI)
```

we define a `clause`-template and specify the rules as template-following by using '`%`' in the arrow:

```
%template clause ====> order(RHS)

clause %> VBU VI
clause %> VBU R VRI
```

Rule levels. Some grammatical phenomena occur very sparsely in common texts. The best way to capture this sparseness is to train rule probabilities on a large data bank of derivation trees acquired from corpus sentences. Since preparation of such corpus of adequate size (at least tens of thousands of sentences) is a very expensive and tedious process, we have for now overcome this difficulty with defining *rule levels*. Every rule without level indication is of level 0. The higher the level, the less frequent the appropriate grammatical phenomenon is. Rules of higher levels can be set on or off according to the chosen level of the whole grammar.

```
3:np -> left_modif
  propagate_case_number_gender($1)
```

In the above example the rule is of level 3, thus when we turn the grammar level to at least 3, we allow adjective groups to form a separate intersegment. When analysing with grammar of level 0 the rule '`np -> left_modif`' is not seen as a part of the grammar at all.

3.1.2 The Second Grammar Form (G2)

As we have mentioned earlier, several pre-defined grammatical tests and procedures are used in the description of context actions associated with each grammatical rule of the system. We use the following tests:

- grammatical case test for particular words and noun groups

  ```
  noun-gen-group  ->  noun-group noun-group
     test_genitive($2)
     propagate_all($1)
  ```

- agreement test of case in prepositional construction

  ```
  prep-group  ->  PREP noun-group
     agree_case_and_propagate($1,$2)
  ```

- agreement test of number and gender for relative pronouns

  ```
  ng-with-rel-pron  ->  noun-group ',' rel-pron-group
     agree_number_gender_and_propagate($1,$3)
  ```

- agreement test of case, number and gender for noun groups

  ```
  adj-ng  ->  adj-group noun-group
     agree_case_number_gender_and_propagate($1,$2)
  ```

The collection of contextual actions denoted with `propagate_all` and `agree_*_and_propagate` take care of the transportation of all relevant grammatical information from the non-terminals on the right hand side to the one on the left hand side of the rule.

Actions. During the process of design and implementation of our system, we started to distinguish four kinds of contextual actions, tests or constraints:

1. rule-tied actions
2. agreement fulfilment constraints
3. post-processing actions
4. actions based on derivation tree

Rule-tied actions are quite rare and serve mainly as special counters for rule-based probability estimation or as rule parameterization modifiers. Agreement fulfilment constraints are used in generating the G3 expanded grammar, in G2 they serve as chart pruning actions. In terms of [MaxKap91], the agreement fulfilment constraints represent the functional constraints, whose processing can be interleaved with that of phrasal constraints. The post-processing actions are not triggered until the chart is already completed. They are used, for instance, in the packed dependency graph generation (see the Section 3.3.1). On the other hand, there are some actions that do not need to work with the whole chart structure, they are run after the best or n most probable derivation trees are selected. These actions do not prune anything, they may be used, for example, for outputting the verb valencies from the input sentence.

Rule heads. Apart from the common generative constructs, the metagrammar comprises feature tagging actions that specify certain local aspects of the denoted (non-)terminal. One of these actions is the specification of the head-dependent relations in the rule — the `head()` construct:

```
/* prvni clanek - first article */
np -> left_modif np
  head($2,$1)

/* treba - perhaps */
part -> PART
    head(root,$1)
```

In the first rule, `head($2,$1)` says that (the head of) `left_modif` depends on (the head of) `np` on the right hand side. In the second example, `head(root,$1)` links the `PART` terminal to the root of the resulting dependency tree. More sophisticated constructs of this kind are the `set_local_root()` and `head_of()`, whose usage is demonstrated in the following example:

```
/* ktery ... - which ... */
relclause ===> first(relprongr) rhs(clause)
    set_local_root(head_of($2))
```

Here, the heads in `rhs(clause)` are assigned as specified in the derivation rules for `clause`. This way we obtain one head of the `rhs(clause)` part and can link all yet unlinked terms to this head.

3.1.3 Expanded Grammar Form (G3)

Context-free parsing techniques are well suited to be incorporated into real-world NLP systems for their time efficiency and low memory requirements. However, it is a well-known fact that some natural language phenomena cannot be handled with the context-free grammar (CFG) formalism. Researchers therefore often use the CFG backbone as the core of their grammar formalism and supplement it with context sensitive feature structures (e.g., [PollSag94], [Neidle94]). The mechanism for the evaluation of feature agreement is usually based on unification. The computation can be either interleaved into the parsing process, or it can be postponed until the resulting structure which captures all the ambiguities in syntax has been built [LavRos2000].

In our approach, we have explored the possibility of shifting the task of feature agreement fulfilment to the earliest phase of parsing process — the CFG backbone. This technique can lead to a combinatorial expansion of the number of rules, however, as we show in this work, it does not need to cause serious slow-down of the analysis.

In a certain sense, we investigate the interface between phrasal and functional constraints as described in [MaxKap91]. They compare four different strategies — interleaved pruning, non-interleaved pruning, factored pruning, and factored extraction and see the fundamental asset in the factoring technique. On the other hand, we use a special structure for constraint evaluation. This structure stores all the possible propagated information in one place and allows to solve the functional constraints efficiently at the time of the chart edge closing. Therefore, factoring cannot play such key role in our system.

[MaxKap91] further discussed the possibility of translating the functional constraints to the context-free (CF) phrasal constraints and vice versa and noted that “many functional constraints can in principle be converted to phrasal constraints, although converting all such functional constraints is a bad idea, it can be quite advantageous to convert some of them, namely, those constraints that would enable the CF parser to prune the space of constituents". To date, the correct choice of the functional constraints selected for conversion has been explored mostly for English. However, these results cannot simply be applied in morphologically rich languages like Czech, because of the threat of massive expansion of the number of rules. Our preliminary results in answering this question for Czech suggest that converting the functional constraints to CF rules can be valuable for noun phrases, even if the number of rules generated from one original rule can be up to 56 (see below). An open question remains, how to incorporate the process of expansion to other agreement test checking, especially the subject–predicate agreement and verb subcategorization. Here, the cause of problems are the free word order and discontinuity of constituents omnipresent in Czech. Moreover, ellipses (deletions) interfere with the expansion of verb subcategorization constraints and even of the subject–predicate agreement tests (subject can be totally elided in Czech).

The chart parsing techniques for extended CFGs are often underspecified with respect to the way how and when the rule constraints are evaluated. An elegant solution is a conversion of the agreement fulfilment actions to the CF rules. For instance, a grammar rule

```
pp -> prep np
  agree_case_and_propagate($1,$2)
```

is transformed to

```
pp1 -> prep1 np1
pp2 -> prep2 np2
pp3 -> prep3 np3
...
```

In Czech, similar to other Slavic languages, there are 7 grammatical cases (nominative, genitive, dative, accusative, vocative, locative and instrumental), two numbers (singular and plural) and four genders (masculine in two forms — animate and inanimate, feminine and neuter). Thus in the process of expanding a G2 rule to a set of G3 rules, we may get up to 56 possible variants for a full agreement between two constituents.

3.2 Parser

The parser design and implementation has passed through a long development and has undergone a lot of changes during its lifetime. The only thing that remains stable is the source code language used — we have chosen C/C++ for its efficiency and portability.

3.2.1 History

Our first parser builder was based on the public domain parser generator BtYacc developed by Chris Dodd and Vadim Maslov [BtYacc98], which is an open source program written in C programming language and is designed and carefully tuned for efficiency and portability.

BtYacc processes a given context free grammar and constructs a C program capable of analysing input text according to the grammar rules. Natural language processing involves manipulation with grammars, that allow more than one possible analysis of the input sentence. BtYacc enables the processing of ambiguous grammar that in case of ordinary LALR analysis causes shift-reduce or reduce-reduce conflicts, which are in deterministic systems solved by choosing only one variant according to predefined precedences. For the purpose of working with ambiguous grammar we have implemented an intelligent backtracking support for BtYacc that is combined with routines which take care of successive formation of the derivation tree.

This approach had brought feasible parsing times for quite a large percentage of natural language phenomena. However, it ran into problems when analysing sentences with a high degree of ambiguity. Since the system allowed only serial processing of multiple analyses, the running time could (for specific sentences) rise up-to several hours (usual sentences were parsed within seconds).

Now we keep the possibility of that kind of analysis mainly for a side-effect feature, *animated LALR analysis*. For the purpose of prototyping and debugging, we let the analyser output every partial derivation tree and display it with a script `tree.tcl`. This way, we can demonstrate very graphically the process of LALR analysis with backtracking.

3.2.2 Chart Parsing vs. GLR

In our work, we have successively tried several different techniques for syntactic analysis. We have tested the top-down and bottom-up variants of the standard chart parser. For more efficient natural language analysis, several researchers have suggested the concept of head-driven parsing (e.g., [Kay89], [Noo97]). Taking advantage of the fact that the head-dependent relations are specified in every rule of our grammar to enable the dependency graph output, the head-driven approach has been successfully adopted in our system. Currently, we are testing the possibility of incorporating the Tomita's GLR parser [Tomita86, HNS91] for the sake of comparing the efficiency of the parsers and the feasibility of implanting a probabilistic control over the parsing process to the parser.

The number of rules may differ significantly in particular grammars. Extreme values are reached in grammars that are obtained automatically from corpus processing [Moore2000]. Our parser is designed in order to cope with drastic increases in number of rules without the loss of its speed. We use as an experiment a grammar of about 35000 rules that were expanded from the base rules plus the unification actions and the rise of analysis time is negligible. Even several hundred thousand rules (created by multiplying the ruleset) are no challenge for the analyser.

Since the number of rules that we need to work with is fairly big (tens of thousands), we need efficient structures to store the parsing process state. The standard chart parser implementation used in our experiments (see [Kadlec2001]) employs 4 hash structures — one for open edges, one for closed edges, one hash table for the grammar rules (needed in the prediction phase) and one for all edges in the agenda or in the chart (the hash key is made of all the attributes of an edge — the rule, the dot position and the surface range). In the case of a head-driven chart parser, we need two hashes for open edges and also two hashes for closed edges.

The gain of this rather complex structure is the linear dependency of the analysis speed on the number of edges in the resulting chart. Each edge is taken into consideration twice — when it is inserted into the agenda and when it is inserted into the chart. The overall complexity is therefore $2k$, where k is the number of edges in the resulting chart.

The number of chart edges that are involved in the appropriate output derivation structure is related to:

a) the number of words in the input sentence, and

b) the ambiguity rate of the sentence.

The output of the chart parser can be presented in the form of a packed shared forest, which is also a standard product of the generalized LR parser. Thus, it enables the parser to run the postprocessing actions on a uniform platform for the different parsers involved.

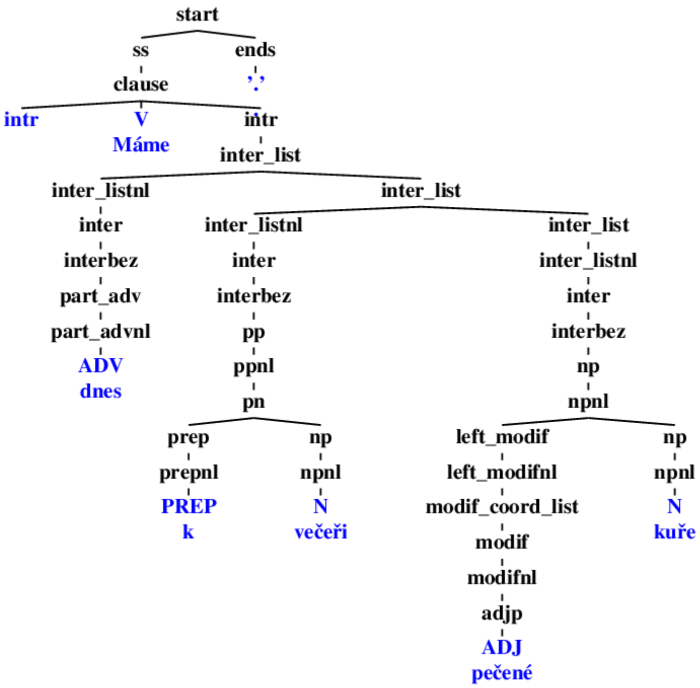


Figure 3.1: An example of resulting derivation tree for sentence '`Máme dnes k večeři pečené kuře.`' (*We have a roast chicken for dinner today.*)

3.3 System Output

The main benefits of the system are gained from the resulting chart or the packed shared forest that both display all the possible derivation trees that properly correspond to the input sentence and the grammar. The most natural way to display this result is the listing of all the trees in a form like in the Figure 3.1.

The raw form of the tree is laid down by the appropriate CF grammar rules. Thus it contains a substantial number of technical nodes, which are not directly relevant to the syntactic structure (`*nl`, `inter_list`). They help to express phenomena like grouping of constituents into coordinated or non-coordinated lists or propagating the grammatical information from one nonterminal to another. Thus, the only grievance to the output of our parser from the linguistic point of view regarded the complexity of derivation trees, namely the high number of levels in trees. We have therefore provided the possibility to postprocess the output with the aim to specify the importance of selected nonterminals that are then displayed in the result (see the Figure 3.4 or the examples in the Appendix A).

However, displaying the results in the form of list of derivation trees is feasible only for less ambiguous inputs, since for large and highly ambiguous sentences the number of obtained derivation trees can go to hundreds of millions due to the exponential behaviour of the linguistic attachment problems (this is not a unique result of this kind, cf. eg. [Kaplan2000]).

Even if result ordering is not a common part of a parser and some authors of a syntactic analyser leave this work to the user, we are looking for ways that enable us to reduce the number of output analyses and to select the most probable (according to a linguistic norm) derivation. To be able to work with the high number of obtained analysis in expanded form (i.e. not in chart or packed shared forest), we are providing other forms of output as well as the possibilities of decreasing the probability of an analysis by means of lexico-semantic constraints. However, a definite solution to the problem of result ordering will be in a form of full probabilistic prelexicalized chart parsing, whose implementation in our system is currently in testing stage and is the main stream of our future directions.

3.3.1 Packed Dependency Graph

A common approach to acquiring the statistical data for the analysis of syntax employs learning the values from a fully tagged tree-bank training corpus. Building such corpora is a tedious and expensive task and it requires a team cooperation of linguists and computer scientists. At present, the only source of Czech tree-bank data is the Prague Dependency Tree-Bank (PDTB) [Haj98], which contains dependency analyses of about 98000 Czech sentences.

The linguistic tradition of Czech syntactic analysis is constituted by distinguishing the role of head and dependent and describes the relations between a head and its dependents in terms of semantically motivated dependency relations. In order to be able to exploit the data from PDTB, we have supplemented our grammar with the dependency specification for constituents. Thus, the output of the analysis can be presented in the form of a pure dependency tree. At the same time, we unify classes of derivation trees that correspond to one dependency structure. We then define a canonical form of the derivation to select one representative of the class which is used for assigning the edge probabilities.

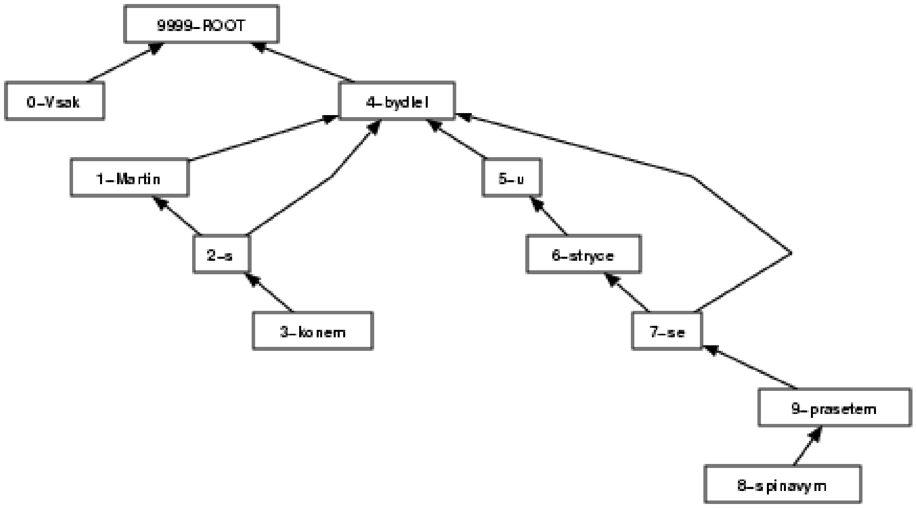


Figure 3.2: Dependency graph.

The dependency structures for all possible analyses are stored in the form of a packed dependency graph. Every "non-simple" rule (that has more than one term on the right hand side) is extended by a denotation of the head element and its dependents. Thus, the dependency is often given as a relation between non-terminals, which cover several input words. However, the basic units of the dependency representation are particular surface elements (words). To be able to capture the standard dependency relations, we propagate the information about a "local head" from the surface level through all the processed chart edges up to the top. A simplified case that captures only one possible derivation of sentence '`Máme k večeři kuře.`' (*We have a chicken for dinner.*) can be described by the tree in the Figure 3.4.

During the evaluation of post-processing actions, every head-dependent relation is then recorded as an edge in the graph (without allowing multi-edges). An example of the graph for the sentence '`Však Martin s koněm bydlel u strýce se špinavým prasetem.`' (literally: *However, Martin with a horse lived with his uncle with a dirty pig.*) is depicted in the Figure 3.2. Two examples of unpacked derivation trees that are generated from the graph are illustrated in the Figure 3.3.

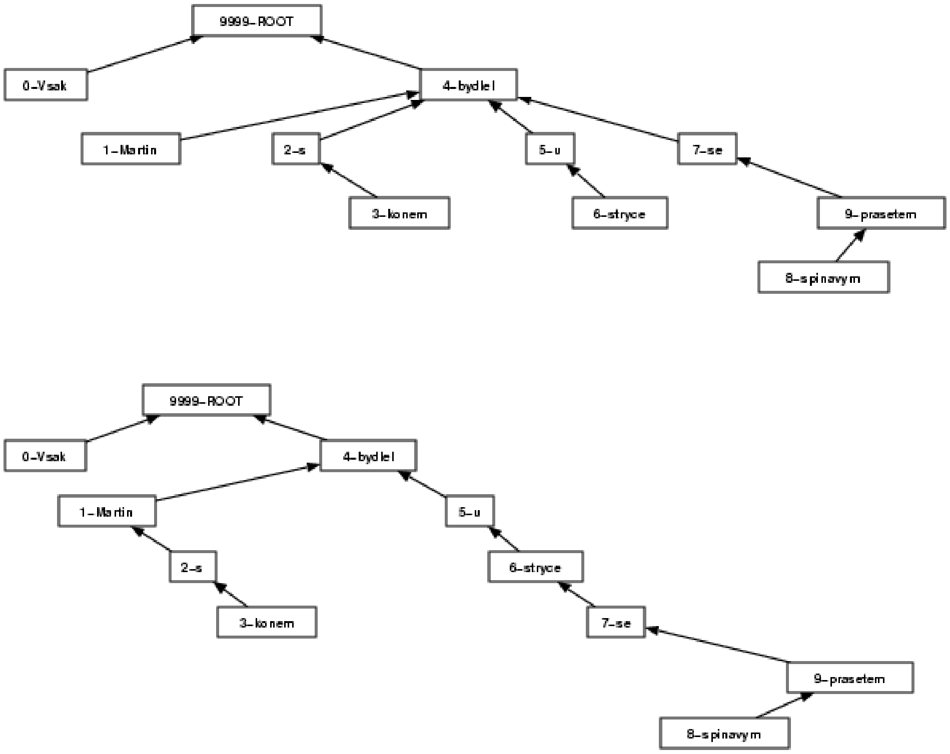


Figure 3.3: Two of the four possible dependency trees.

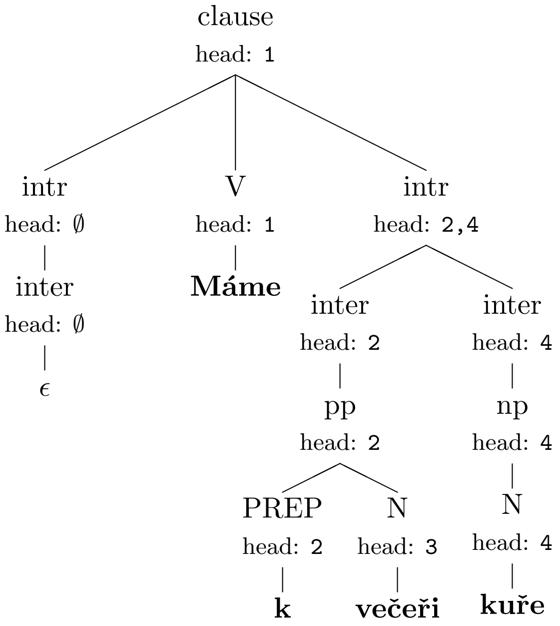


Figure 3.4: The head propagation up the tree.

The packed dependency graph enables us to recover all the possible standard dependency trees with some additional information gathered during the analysis. The example graph represents four dependency trees only, however, in the case of more complex sentences, especially those with complicated noun phrases, the saving is much higher.

3.3.2 Lexico-semantic Constraints

The analysis is supported by a set of commonly used grammatical tests that are described in the Section 3.1.2. In addition to these tests we have extended the valency test functions with lexico-semantic constraints. The constraints take advantage of an ontological hierarchy of the same type as in Wordnet [WordNet90]. They enable us to impose a special request of compatibility with selected class or classes in the hierarchy to each valency expression. In current version we use a very limited subset of the complete hierarchy and we plan to connect the system to the results of Czech part of the Eurowordnet 2 project [EWN98].

An example of the constraints in action can be demonstrated by the following phrase:

$$\begin{array}{ccc} \text{Leaseholder} & \text{draws} & \text{beer.} \\ \underbrace{\text{Nájemce}}_{\texttt{k1gMnSc1245,k1gMnPc4}} & \text{čepuje} & \underbrace{\text{pivo.}}_{\texttt{k1gNnSc145}} \end{array}$$

```
čepovat
= sb.<HUMAN> & st.<LIQUID>
```

The lexico-semantic constraints that are found in the valency list of the verb `čepovat` (draw) makes it possible to distinguish the word `pivo` (beer) as an object and `nájemce` (leaseholder) as the subject. Considering metonymy and other forms of meaning shifts we do not regard this feature so strictly to throw out a particular analysis. We use it rather as a tool for assigning preferences to different analyses.

The part of the system dedicated to exploitation of information obtained from our list of verb valencies [PaSe97, Horak98] is necessary for solving the prepositional attachment problem in particular. During the analysis of noun groups and prepositional noun groups in the role of verb valencies in a given input sentence one needs to be able to distinguish free adjuncts or modifiers from obligatory valencies. We are testing a set of heuristic rules that determine whether a found noun group typically serves as a free adjunct. The heuristics are also based on the lexico-semantic constraints described above.

$$\overset{\text{At the meeting}}{\text{Na }\underbrace{\text{schůzi}}_{\texttt{<ACTIVITY>}}}\qquad \overset{\text{Peter angered}}{\text{se Petr rozhněval}}\quad \overset{\text{with Charles}}{\text{na }\underbrace{\text{Karla}}_{\texttt{<HUMAN>}}}$$

$$\overset{\text{about the lost advance}}{\text{kvůli ztracené záloze}}\quad \overset{\text{for payroll}}{\text{na }\underbrace{\text{mzdu.}}_{\texttt{<RECOMPENSE>}}}$$

In this example, the expression `na Karla` (with Charles) is denoted as a verb argument by the valency list of the verb `rozhněvat se` (anger), while the prepositional noun phrase `na schůzi` (at the meeting) is classified as a free adjunct by the rule specifying that the preposition `na` (at) in combination with an `<ACTIVITY>` class member (in locative) forms a location expression. The remaining constituent `na mzdu` (for payroll) is finally recommended as a modifier of the preceding noun phrase `záloze` ([about the] advance).

Chapter 4

Transparent Intensional Logic

Since the times of old Greek philosophers, many thinkers have always been in quest of the explication of *meaning* (of a natural language expression). In the contemporary philosophy of meaning, among the most often discussed explanations are the ideas of Frege[1], Russell[2] and Quine.[3] In computer science researchers often follow up the works of Church[4] and Montague.[5] In our work we want to emphasize and discuss the conception of Tichý[6] and his followers (esp. Materna[7]), named the *transparent intensional logic* (TIL).

Our discussion here comes out primarily from the monographs by Tichý and Materna [Tichy88, Materna95, Materna98] and numerous Tichý's articles, esp. [Tichy80a, Tichy80b, Tichy94b, Tichy94a]. The asset of this thesis lies in the explication of the *logical analysis of natural language* by means of (semi)automatic assigning/searching of the representation of meaning of a natural language (NL) sentence. Moreover, the feasibility of the proposed algorithm is verified by implementation of its main parts.

In the next section, we present and argue for the main views of Tichý's TIL also with informal discussion of Materna's definition of *concept*. This discussion is then formalized in the (summary) definitions of the main notions of TIL (type, construction, variable, . . .) in the following section.

---

[1] Gottlob Frege, *1848 – †1925. esp. [Frege1892]

[2] Bertrand Russell, *1872 – †1970. esp. [Russel1903]

[3] Willard Van Orman Quine, *1908 – †2000.

[4] Alonzo Church, *1903 – †1995.

[5] Richard Montague, *1930 – †1971. esp. [Montague74]

[6] Pavel Tichý, *1936 – †1994. esp. [Tichy88]

[7] Pavel Materna, *1930. esp. [Materna98]

Our algorithm of the NL logical analysis is in detail described in the Chapter 5, where we apply the ideas presented and sketched throughout the whole work of Tichý and Materna as well as many of our own proposals to analysis of expressions that have not been mentioned in any text, so far. Tichý was about to publish most of them in his cogitated book *The Analysis of Natural Language.* However, he managed to write only the first out of the intended twelve chapters [Tichy94a], and thus has left a lot of particular phenomena of NL without the prescription of their proper analysis (in TIL). In this work, we want to offer an exact representation for the most frequent language phenomena and to open a (welcomed) discussion leading to a complete logical analysis algorithm.

4.1 Basic Ideas

There is a fundamental predicament about meaning, that should be followed by all meaning explications, known as Frege's Functionality Principle:[8]

> The meaning of a (compound) NL expression is a *function* of the meanings of its constituents.

It is not only the case, that this rule is consistent with intuition, moreover, if any of the meaning representations fails to obey this principle, it consequently leads to serious flaws and paradoxes.[9]

When seeking for the elucidation of meaning, Tichý thoroughfully examined the most promising theories and, since all of them suffered from unacceptable inconsistencies in various places, he has introduced the transparent intensional logic with the fundamental conception of construction as a possible meaning naming tool.

4.1.1 Expression-Meaning Relationship

One of the most important parts of Frege's logic was his conception of meaning. In contrast to other logicians of his times, Frege realized the three-fold character of the expression-meaning relation — with meaning differentiated to *sense* and *reference.*[10] These two components of understanding process combine together with a NL expression according to the diagram in the Figure 4.1a).

---

[8] known also as the *Compositionality Principle.*

[9] like Montague's hat '$^\wedge$' symbol, see the Section 4.1.2.

[10] the often cited original denotations are *Sinn* and *Bedeutung*

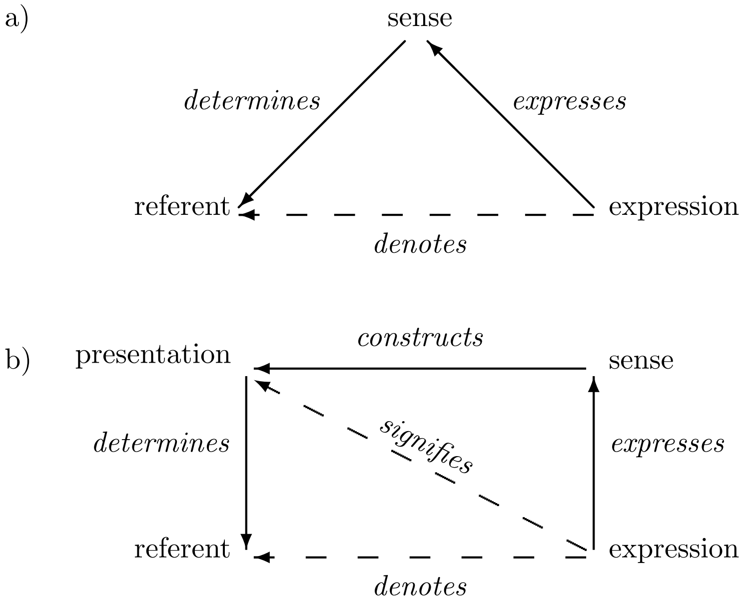


Figure 4.1: a) Frege's three-fold expression-meaning relation and b) its four--fold amendment by Tichý.

This theory allowed for resolution of many paradoxes that were ungraspable with meaning analysed as a bare truth-value. Frege's sense is typically a compound, i.e. the sense of a compound expression has as its *parts* the senses of the expression's constituents. What Frege missed in the expression--meaning relation is the way *how* those constituents' senses combine. In order to complete the diagram, Tichý amended it by changing it into a four--fold one (see the Figure 4.1b)) with adding the relation *constructs* between sense and the mode of presentation of the referent. As an example, we may take the instantiated diagram of the expression 'the author of Hamlet' in the Figure 4.2. In the diagram, we use the abbreviations

Ao	function that takes every drama to its author
H	Hamlet, the drama
AH	the intensional role the author of Hamlet

Even after Tichý offered this new diagram as a correction of the original one from Frege, he saw that the idea that a NL expression (like 'the author of Hamlet') denotes the extension of the mode of presentation (i.e. William Shakespeare in this case) is unacceptable. For if it were so, the

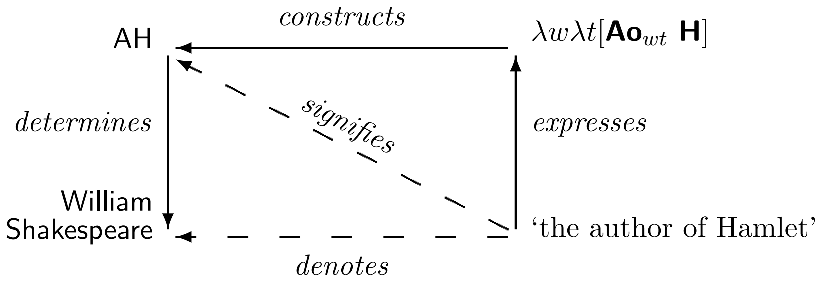


Figure 4.2: An instantiated diagram of Tichý's amendment of the Frege's expression-meaning relation — the meaning of the expression 'the author of Hamlet'.

object the author of Hamlet would have to (in some "magic" way) comprise the particular individual William Shakespeare. With this, no one could understand (identify the meaning of) the expression 'the author of Hamlet' without knowing that it was William Shakespeare who wrote the drama. This obvious discrepancy led Tichý to the final conception of expression--meaning relation as it is represented in TIL (see the Figure 4.3a)). Here, the meaning of an expression is completely separated from the (empirically obtained) extension of the object in mind.

Tichý's diagram was further extended by Materna[11] who has linked together equivalent constructions into one abstract entity — the *concept*. This collection of constructions can be *generated* by any of its elements and as a whole it represents the (unambiguous) way of constructing the identified object. Some more discussion on concepts can be found in the Section 4.2.5 of this work, where we also summarize the formal definition of concept.

Frege's thesis lay in representing meaning of a NLE in the form of a *Function*.[12] If we set aside the fact that with that apparatus Frege was not at any means able to capture the higher order phenomena (like belief), the main discrepancy of his thesis consisted in the explanation of a subject

---

[11] see schema (S) in [Materna98].

[12] In his early work, Frege explicitly noted that his Function is not a mere mapping, Function is rather a *(structured) entity* designated by a Functional expression.

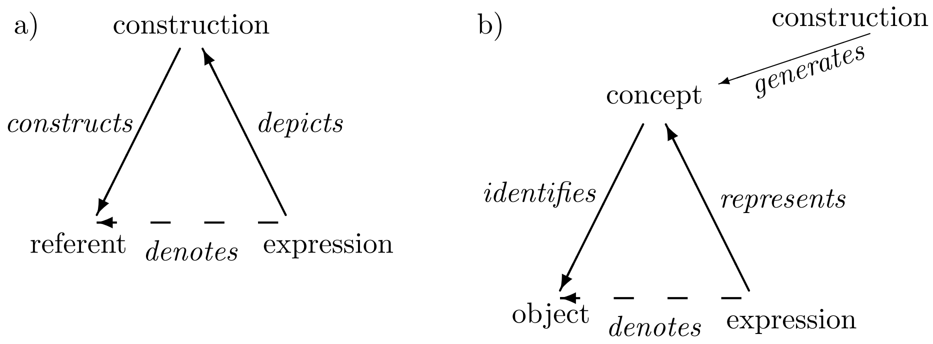


Figure 4.3: a) the expression-meaning relation in TIL and b) with Materna's conceptual approach.

matter of intensional expressions. The well known example for this is the statement about morning/evening star[13]

$$\text{The morning star is the evening star.} \tag{4.1}$$
$$\text{The morning star is the morning star.} \tag{4.2}$$

The difference in understanding of those sentences lies in that two different *modes of presentation* point to one and the same object in (4.1) (while (4.2) is a mere tautology). According to Frege's conception still no part of (4.1) refers to either of these presentations, the only topic spoken about is the planet of Venus. And this is really a flaw, since certainly no fact in (4.1) connects the meaning of this sentence with any particular individual at all. It is perfectly imaginable that the *role* of the morning/evening star could be played by another celestial body and the characteristics of the proposition (4.1) (viz. truthness) would stay the same.

Tichý sets this inconsistence down to Frege's unsound interpretation of the distinction between an *application of a mapping* (function) to an argument and a *composition* as a construction of the application of such mapping. The proper analysis of propositions (4.1) and (4.2) in TIL may look like

$$\lambda w \lambda t[\mathbf{MS}_{wt} = \mathbf{ES}_{wt}]^{14} \tag{4.1'}$$
$$\lambda w \lambda t[\mathbf{MS}_{wt} = \mathbf{MS}_{wt}], \tag{4.2'}$$

---

[13] "morning star"=the brightest celestial body at the day-break (Jitřenka in Czech), "evening star"=the same at the dusk (Večernice in Czech). Both of them have been discovered by astronomers to be the planet of Venus.

where the meaning of the statement (4.1) is analysed rather as a construction of equivalence between individuals that at a specific possible world w and time t play the role of being the brightest object on the morning/evening sky.

4.1.2 Logical Analysis through the Looking-Glass

The crucial point of many of Tichý's arguments fueling his claims can be expressed in two items:

a) the logical analysis cannot be a *translation* of NL expression (NLE), it should rather *name* whatever is depicted by the expression.[15]
b) the logical analysis cannot find more facts than those that are really stated in the NL sentence/expression and are assignable *a priori*.[16]

Many authors amiss analyse meaning of NL expressions by *translating* them into a formal language (a "toy language" for Tichý) and carry out all successive processing by means of that instrumentality. However, what is usually the missing point of their theories is the (often very difficult) demonstration that the formal language really *is* in all means equivalent to NL and that any proofs and operations we conduct on the surrogate really apply (the same as) in the case of NL itself. Thus, when we know that something holds in the extent of the formal language, we may know nothing about the validity or counter-validity of the same fact in NL.[17]

Moreover, the formal language is often a source of paradoxes that do not have its reflection in NL — usually caused by oversimplification of the representation of some NL phenomena such as *intension/extension* difference, *temporal aspect* of events or *belief* attitudes.

In the light of these claims, we follow an alternative approach (viz. TIL) where no other *language* is brought into play, instead the NL expressions themselves are the subject of study and what they *mean* is analysed as *constructions* of the objects denoted by the expressions. This fact is so

---

[14] here we use the abbreviation notation of typesetting the trivialization construction in bold font, i.e. $^0\mathsf{A}$ is written as $\mathbf{\mathsf{A}}$. For the explanation of trivialization see the Section 4.1.5.

[15] see e.g. [Tichy94b]

[16] esp. the *pragmatic content* of an expression (such that depends on the *communication situation*) cannot be the meant topic whose knowledge is postulated by our understanding of the NLE.

[17] Tichý offers an analogy for Montague's intensional logic (IL) and the English language. If we want to understand how English works by means of Montague's IL, it is analogous to explaining the functionality of an electronic digital watch by unveiling the mechanism of springs and cogs in an old grandfather clock.

important for TIL, that is is emphasized even in its name — TIL as a tool that works with a natural language is completely *transparent* in the sense that we always work directly with the NL in the logical analysis, not with any extra formal language.

Among the contemporary logicians one of the most often quoted solutions of the problem of *intensions* and *extensions* is the approach introduced by Montague.[18] In his Intensional Logic (IL) he defines two operators:[19]

$^\wedge\alpha$ … expression denoting (or having as its extension) the intension of the expression α; and
$^\vee\alpha$ … denotes the extension of α and as such is meaningful only if α is an expression that denotes an intension or sense.

However, such simplification may bring (and really does so) more dissonance than clarification into the system. The main source of semantic misinterpretation is the '$^\wedge$' operator, since it is not at all clear what is the *structure* of the term $^\wedge\alpha$. If we suppose that α in it stands for an extension, e.g. the individual Venus (the planet), than which of the infinitely many intensions that may refer to this individual ('the morning star', 'the evening star', 'the second planet of the Solar system', …) is actually represented by $^\wedge\alpha$? Clearly this cannot be the way the operator could work in IL. But, if we regard the noncontextual reading of α, the '$^\wedge$' operator would not follow even the basic Functionality Principle.

In TIL this problem is transparently resolved by compositional construction of a mapping that abstracts from the dependency on the actual possible world and present time and thus correctly postpones the pragmatic "extensification" to empirical inquiry in outside of the system of the logical analysis.

4.1.3 Possible Worlds

The notion of a *possible world*[20] is often very unamenable as a subject to precise definition. In TIL, we use this conception for capturing the *modal* and *temporal variability* of facts. For this purpose the term "world" does not mean a collection of particular existing objects and things, of course. The conception rather comes out from findings similar to those formulated in [Materna98, pp. 25f]:

1. a possible world is a collection of thinkable facts;
2. it is consistent and maximum of such sets;
3. a possible world is objective (individually independent)

---

[18] see [Montague74]

[19] see the discussion in [Tichy88, pp. 131,151]

[20] usually ascribed to Gottfried Wilhelm von Leibniz, *1646 – †1716.

	being corpulent			
being slender	{Laurel, Hardy}	{Laurel}	{Hardy}	$\emptyset$
{Laurel, Hardy}	×	×	×	w_1
{Laurel}	×	×	$\boxed{w_2}$	w_3
{Hardy}	×	w_4	×	w_5
$\emptyset$	w_6	w_7	w_8	w_9

Table 4.1: Determination systems representing possible worlds ($w_1, \ldots, w_9$) of a toy reality with 2 individuals and 2 features (in one moment of time).

One fact arises consequently of these claims: among the possible worlds there is exactly one world the is *actual*, i.e. that completely and perfectly reflects the state of our reality. Nevertheless, nobody is able to point it out, since knowing the collection of maximum consistent and veritable facts about the reality would certainly require an omniscient being, which is very uneasy to find.

However, as most humans are able to capture sentence meaning without the necessity to determine its truth-value, the logical analysis of a sentence does not need to regard the actuality of its content.

A possible world, in accordance with the above claims, is in TIL defined as a *determination system* that, for each of the intuitively, pre-theoretically given features from intensional base, contains an assignment of all possible (consistent) distributions. As an example, let us imagine a very limited reality with just two individuals Laurel and Hardy and only two features in the intensional base, viz. being slender and being corpulent. All possible distributions of these two features between Laurel and Hardy are summarized in the Table 4.1. The fields marked with × correspond to inconsistent collections of facts,[21] the fields w_k denote determination systems that represent possible worlds interpretations in one particular time moment. If the moment belonged to the time interval in which the two famous entertainers of the black-and-white film era set millions of onlookers in a roar, than the field $\boxed{w_2}$ could represent our actual world (its part projected to this toy reality, of course).

---

[21] In the toy reality, inconsistent collections of facts are those distributions where one and the same individual is at the same time slender and corpulent (belongs to the intersection of the classes determined by properties being slender and being corpulent at the same time and world).

For a real-world property, we (naturally) do not try to enumerate it in the way we presented in the Table 4.1. Instead, we refer to one of the possible worlds (determination systems) and one specific time moment with variables (w, t) with the interpretation[22]

being slender	... an object of type $(((o\iota)\tau)\omega)$, a mapping from a possible world and a time moment to a class of individuals
w	... a variable of type ω, a possible world
t	... a variable of type τ,[23] a time moment
[**being slender** $w\,t$]	... constructs an $(o\iota)$-object[24], a class of all the individuals (given by its characteristic function) that at the world w and time t have the property of being slender.

In case we apply only the first of the pair of variables w, t to an object of type $\xi_{\tau\omega}$,[25] we obtain a $\xi\tau$-object that can be interpreted as a *chronology* of the extension of type ξ, i.e. the course of values (ξ-objects) throughout the whole time axe. For instance the *intensional role* American president ($\iota_{\tau\omega}$-object) has (a part of) its chronology in the actual world w_{act} of this form:

American president$_{w_{act}}$ (shortly $\mathbf{P}_{w_{act}} \ldots \iota_\tau$):

The relationship between the class of all possible worlds and the class of time moments, as it is used in the denotation of intensional objects, is depicted in the Figure 4.4.

4.1.4 TIL Types

The basic differentiation of objects in TIL is based on the theory of types introduced by Church.[26] In this theory every object has its *type* which is

---

[22] for the description of the TIL types $\{o, \iota, \tau, \omega\}$ see the Section 4.1.4.

[23] we write A/ξ as a short for an object A of type ξ.

[24] we write $C \ldots \xi$ (and $x \ldots \xi$) as a short for a construction C that constructs an object of type ξ (a variable x of type ξ is also a construction that constructs an object of type ξ).

[25] a type of the form $((\xi\tau)\omega)$ is shortly written as $\xi_{\tau\omega}$.

[26] and rectified by Tichý in [Tichy88].

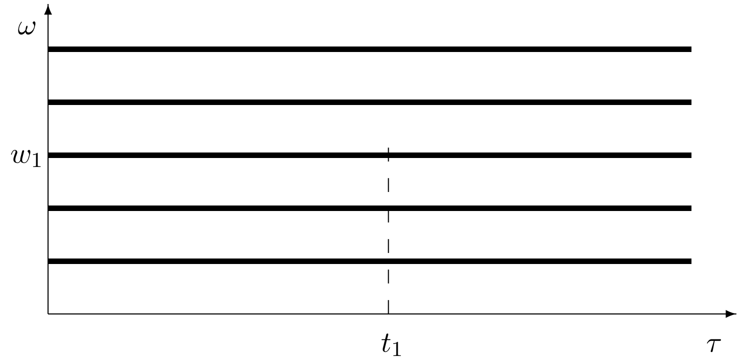


Figure 4.4: The relationship between possible worlds and time moments.

defined over a firmly set type base. Every object (which is not a construction) is assigned either one of the *basic types*, or a type that is formed by a *mapping* from one type to another type. Within this framework, we can obtain an infinitely nested hierarchy of types, i.e. mappings of basic types, mappings of mappings, mappings of mappings of mappings, etc. Nevertheless, how difficult soever the mapping is, the object of the respective type is still "flat". The flatness of mappings is predicated upon the way mappings are treated — as collections of $(n+1)$-tuples (in case of an n-adic mapping). This means that the mapping is represented by a table of values without any possibility to find out the way[27] (a procedure) which leads to those values. This is also one of the reasons why mere mappings cannot serve as surrogates for meaning — mappings lack a constructing *structure*.

The idea of logical analysis of NL with TIL lies in the presupposition that every language has a definite *intensional base* — a collection of fundamental properties[28] (colours, heights, attitudes, ...) of objects, that are capable (without any need for other extra techniques) of describing a (thinkable) state of the world.

In TIL, such an intensional base of a NL is rigorously explicated in an *epistemic framework*, i.e. a typed system based on a set of four basic types $\{o, \iota, \tau, \omega\}$, called a type *base*, together with an explicit interpretation of its members:

o ... a set of two items representing the *truth-values* True (T) and False (F). These two objects behave exactly the same as their

---

[27] For every mapping there are infinitely many (structural) ways of obtaining one and the same table of values.

[28] not only $(o\iota)_{\tau\omega}$ but any world and time depending relations among objects.

counterparts in the standard predicate logic especially in combination with standard logic operations such as conjunction, disjunction, implication or negation. These predicate logic operations can be represented as objects of type (oo) or (ooo), i.e. functions with one or two arguments of type o returning a value of type o.

ι ... a class of *individuals*. The designation "individual" must not entice us to imagine the members of this class as beings with all their properties. In TIL, the notion of an individual is best interpreted as a mean of a numerical identification of an (type unstructured[29]) entity. Any individual properties are ascribed to an object of type ι only by means of asserting a statement that contains the ascribing as its part — in the proposition 'Peter is a tall man,' we use the ι-object Peter[30] as an identification of an entity that is ascribed the property being a tall man (TM)

$$\lambda w \lambda t[\mathbf{TM}_{wt}\mathbf{Peter}]$$

The individual Peter itself is carrying no *a priori* properties, it serves as an identifier of a further unspecified object and is mainly used for references to this object.

τ ... a class of *time moments*. Due to the continuity feature of this class, it may be regarded identical with the class of real numbers, in case we specify a fixed zero point and a unit. Functions working with arguments of type τ are usually used for expressing the *temporal dependency* of an entity.

ω ... a class of *possible worlds*.[31] Its members, the determination systems, are intended for a transparent representation of *modal dependency* of described objects.

---

[29] i.e. whose type is not decomposable as a function to the types of the functional arguments and the type of its return value.

[30] we take Peter here (and on other places in the text) as an example of an individual despite the fact that proper names often need to be analysed as a pragmatically anchored expression (see the Section 5.2.4).

[31] see the Section 4.1.3 for more detailed explanation of the possible worlds conception in TIL.

Any object in TIL has been assigned a type, which is either one of the basic types $\{o, \iota, \tau, \omega\}$ or a mapping of inductively derived types (see the Definitions 2 and 7). The most frequent derived types are summarized in the Table 4.2. On the other hand, the Table 4.3 provides a survey of the operators that appear frequently as basic elements of many constructions stated in the present text.

4.1.5 Construction

The essential motto of the transparent intensional logic is the notion of construction understood as an *abstract procedure* describing a way how to get (by a train of thought) to an object of an idea. By means of such procedure, we can then represent the meaning of an assertion or of any of its self-reliant components. In the diagram in the Figure 4.3a), we can see that construction stands for a mean of capturing the *compositional character* of the denotation of the (otherwise unstructured) object denoted by the NL expression.

So, what exactly is a construction? A precise answer is given in the formal Definition 5, but the basic idea of construction can be described at the moment. Since construction serves as a mechanism to describe the *structure* of something, then the most convenient (if not necessary) method of explanation of how such a mechanism looks like is the specification of how any building step in forming a construction can look like, i.e. the inductive description.

We start with the simplest case of a construction — actually there are two nearly equally simple cases. One of them is the well-known notion of a *variable* (although in TIL we have a specific and precise definition of this notion available). Maybe, the idea of a variable as a procedure is somewhat unusual, but there *is* such a procedure behind it — it says "take the value of this variable and return it". In this way, we can obtain a mean of constructing an object of a specific type, but without any other limitations.

The other simple case is the construction of a particular (known and named) object. The procedure of constructing an object denoted directly by its name consists only in the very single step — returning the object itself. Yes, so trivial it is — that is why such construction is called a *trivialization* of an object.[32]

---

[32] The notation of the trivialization consists in stating a small 0 before the object such as $^0 1$ (constructs the number 1) or ^{0}Peter (constructs the individual Peter), or, in a short way, typesetting the object name in a **bold font** like **Peter** (unlike Peter, the object itself). In this text, we use the bold font notation for objects named with words (i.e. the reader

Type	**Notation**	**Object**
$((o\tau)\omega)$	$o_{\tau\omega}$, π	a *proposition* — an assertion whose truth-value depends on the world and time. **Example:** 'Peter is ill.'
$((\iota\tau)\omega)$	$\iota_{\tau\omega}$	an *intensional role* — an individual office that may be engaged by different individuals in different worlds or times. **Example:** 'the American president'
$(((o\iota)\tau)\omega)$	$(o\iota)_{\tau\omega}$	a *property* — an atomic intensional feature that an individual either has or has not according to a chosen world and time. Such feature is then represented by a class (its characteristic function) of those individuals that have the feature in a certain world and time moment. **Example:** 'being red'
$(\xi\tau)$	ξ_τ	a *ξ-chronology* — a mapping that specifies the flow of changes of a ξ-object in time. **Example:** 'yesterday' is a time interval of 24 hours that ends at the last midnight, thus it represents a different interval every day.
$((\xi\tau)\omega)$	$\xi_{\tau\omega}$	an *intension* — expresses the dependency of the related ξ-object on the selected possible world and time moment. The application of an intension to the world and time (i.e. composition of a $\xi_{\tau\omega}$-object with argument of type ω and then with argument of type τ resulting in a construction of a ξ-object) is called the *intensional descent.* If ξ is not itself an intension, it is called an *extension* and represents an entity whose value does not change with world and time. All mathematical objects (numbers, operations, axioms) correspond to extensions.

Table 4.2: The most frequent derived types in TIL (ξ in the table stands for any type).

Oper.	**Type**	**Notation**	**Comment**
$\wedge, \vee,$ $\supset, \ldots$	(ooo)	$A\,{}^0\!\wedge B$	the *logical operations* are working on truth-values and are usually written in the infix notation instead of the λ-calculus prefix notation $[{}^0\!\wedge AB]$
$\neg$	(oo)	$[{}^0\neg A]$	the *logical negation*
$\subset_\xi$	$(o(o\xi)(o\xi))$	$[A\,{}^0\!\subset_\xi B]$	the *subset* relation between classes of ξ-objects.
$\Rightarrow$	$(o\pi\pi)$	$[A\,{}^0\!\Rightarrow B]$	the *implication* relation between two propositions — $A \Rightarrow B$ if and only if B is true in all the worlds and times where A is true.
Π^ξ	$(o(o\xi))$	$\forall x \ldots$	*universal quantifier* is written in the usual notation instead of $[{}^0\Pi^\xi\ \lambda x \ldots]$
Σ^ξ	$(o(o\xi))$	$\exists x \ldots$	*existentional quantifier* is written in the conventional notation instead of $[{}^0\Sigma^\xi\ \lambda x \ldots]$
I^ξ	$(\xi(o\xi))$	$\imath x \ldots$	*singularizer* means ‘the only x that $\ldots$,’ written in the short notation instead of $[{}^0 I^\xi\ \lambda x \ldots]$

Table 4.3: The most frequent extensional operators in TIL (ξ in the table stands for any type).

The two remaining cases of a construction are actually the building techniques of all complicated constructions — their use always includes two or more other constructions. The first one is called a *composition* or application, we use it for obtaining a return value of an application of an object-function constructed by a head of the composition to one or more arguments (objects constructed by the corresponding constructions). Thus, for instance, if sinus is the well-known object-function, then **sinus** is the construction (trivialization) of the function and [**sinus** ^{0}0] is the composition of **sinus** and ^{0}0, whose result is the number 0 (because sinus$(0) = 0$).

The other building technique of a construction (and the last kind of constructions in our survey) is named *closure* or λ-abstraction. As the name suggests, this technique comes from the λ-calculus and it serves for

---

can easily detect the font change) and the zero notation for other objects like numbers and compound constructions.

closing a construction with regard to some *variables*. Let us take e.g. the construction $[x \times {}^0 2]$ which constructs a number that is twice the value assigned to x. Such construction is somewhat strange in the sense that we cannot foretell what number will be constructed by it without the extra knowledge about the value of the variable x. We call such construction an *open* construction. However, with the help of the closure technique, we can make this construction closed to the variable x, which is symbolized as $\lambda x[x \times {}^0 2]$, and since x was the only free variable in the open construction then the whole construction $\lambda x[x \times {}^0 2]$ is denoted simply as *closed*. Moreover, the closed construction now constructs not just a number, but the whole function $f(x) = x \times 2$, so we have also changed the type of the constructed object (the same holds for the counterpart technique, the composition).

This is all what can such a construction be built with. When working with constructions, we always have to remember that constructions are *not* any kind of *symbolic formulas*, however it may seem so from the similarity of their notation with the notation of the λ-calculus. Constructions just *use* the notation as a specification of the corresponding abstract procedure about which we have talked at the beginning.

4.2 Definitions

After we have interpreted and advocated the introductory notions of TIL in a ‘friendly way’, we are going to corroborate them by formal definitions. In the next sections, we explicate the TIL *types* (simple and higher-order), the central point of the whole theory — *construction*, present Tichý’s conception of *variable* and finally explain and discuss the term *concept* as defined by Materna.

4.2.1 Simple Type

The first definition involves one part of the definition of a *type* — it defines a type of *order 1*, called also a *simple type*. These types cover objects that do range over basic types and over mappings of simple types, they do not comprise entities involving constructions.

Definition 1 (base). *B is a* base *if and only if B is a set of non-empty and pairwise disjoint classes.* □

Definition 2 (simple type). *Let B be a base.*

(t_1-a) *Every member of B (basic type) is a* simple type over B.

(t_1-b) *Let $n > 0$ and $\alpha, \beta_1, \ldots, \beta_n$ be simple types over B. Then the set $(\alpha\beta_1 \ldots \beta_n)$ of all (total and partial) n-ary mappings from $\beta_1 \times \cdots \times \beta_n$ into α is a* simple type over B.

(t_1-c) *Nothing is a* simple type over B *unless it is specified so in* (t_1-a) *or* (t_1-b).

□

Remark 1 ▷ Simple types are also called *types of order 1.* ◁

Remark 2 ▷ The attribute 'over B' is often left out, since we usually assume the types over the TIL objectual base $\{o, \iota, \tau, \omega\}$. ◁

4.2.2 Variable and Valuation

Variables are the only atomic way of constructing an object in TIL. The notion of variable is explicated by transposing the linguistic approach where the variables are unified with letters, into the objectual approach that forms one of the corner-stones of TIL.

Definition 3 (variable). *If ξ is a type, then any sequence $\mathsf{X}_1, \mathsf{X}_2, \ldots$ of objects of type ξ is called a* ξ-sequence. *For any natural number $n > 0$ let $\overset{\xi}{\bigvee}_n$ be a construction that constructs an object by taking it from the n-th place of some ξ-sequence. Constructions $\overset{\xi}{\bigvee}_n$ are called* variables *of type ξ.*
□

Variables are *incomplete* constructions in the sense that they do not construct a fixed object — the object behind a variable depends on the ξ-sequence used for anchoring the variable. Such selection of the sequences for variables of all types is a *valuation.*

Definition 4 (valuation). *Let all the types over the TIL objectual base $\{o, \iota, \tau, \omega\}$ be arranged into a sequence and let ξ_i be the i-th type. Further let $\mathsf{X}^i{}_1, \mathsf{X}^i{}_2, \ldots$ be a ξ_i-sequence.*

An array v of selected ξ_i-sequences for all $i = 1, 2, 3, \ldots$

$$\begin{array}{c} \mathsf{X}^1{}_1, \mathsf{X}^1{}_2, \mathsf{X}^1{}_3, \ldots \\ \mathsf{X}^2{}_1, \mathsf{X}^2{}_2, \mathsf{X}^2{}_3, \ldots \\ \mathsf{X}^3{}_1, \mathsf{X}^3{}_2, \mathsf{X}^3{}_3, \ldots \\ \vdots \end{array}$$

is called a valuation. *A variable* $\overset{\xi_i}{\bigvee}_n$ *constructs* according to a valuation v *(shortly v-constructs) the object* $\mathsf{X}^{i}{}_{n}$.

□

Remark ▷ For the sake of readability, we denote variables in the usual way by lowercase letters with subscripts and we always specify the type of the variable. Thus, we write $x_n \dots \xi$ (or y_n or another letter) instead of $\overset{\xi}{\bigvee}_n$. ◁

4.2.3 Construction

The following definition specifies in an inductive way what parts may a construction consist of.

Definition 5 (modes of construction). *Let α be a type (of order n), v be a valuation.*

- (c-a) *If x is a* ***variable*** *ranging over α, then it is also a (simple)* construction *that constructs an object of type α (an α-construction). It v-constructs the value that is assigned to the variable x by valuation v.*
- (c-b) *If* A *is an object of type α, then the* ***trivialization*** ${}^{0}\mathsf{A}$ *(also typeset as* $\mathbf{\mathsf{A}}$*) is an (α-)*construction. *It constructs the object* A *independently on any valuation.*
- (c-c) *If $m > 0$ and $X, Y_1, \dots, Y_m$ are constructions, then the* ***composition***[33] $[XY_1 \dots Y_m]$ *is a* construction. *If X v-constructs an m-ary mapping f that is defined on the values $y_1, \dots, y_m$ v-constructed by $Y_1, \dots, Y_m$, then the composition v-constructs the value of the mapping $f(y_1, \dots, y_m)$. Otherwise the composition is v-improper.*
- (c-d) *If $m > 0$, Y is an α-construction and $x_1, \dots, x_m$ are distinct variables of types $\xi_1, \dots, \xi_m$, then the* ***closure***[34] $[\lambda x_1 \dots x_m Y]$ *is a* construction. *If* $\mathsf{X}_1, \dots, \mathsf{X}_m$ *are objects of types $\xi_1, \dots, \xi_m$ and v' is a valuation that is the same as v except for assigning* X_1 *to x_1,...,* X_m *to x_m, then the closure v-constructs a mapping $f : \xi_1 \times \dots \times \xi_m \to \alpha$ with the value of* $f(\mathsf{X}_1, \dots, \mathsf{X}_m)$ *being the object that is v'-constructed by Y or undefined if Y is v'-improper. In no case is a closure a v-improper construction.*

---

[33] also denoted as *application*

[34] also referred to as *abstraction* or *λ-abstraction*

(c-e) *Nothing is a* construction *unless it is specified so in* (c-a)*,* (c-b)*,* (c-c) *or* (c-d)*.*

□

Remark 1 ▷ In the definition of construction, we must always realize that all the modes of forming a construction are not just notational expressions that denote the corresponding mapping or its value, but that they rather represent *abstract procedures* or recipes on how to come to an object. ◁

Remark 2 ▷ An application of an $\alpha_{\tau\omega}$-object A onto the possible world w and time moment t is often written as $\mathbf{A}_{wt}$ instead of $[[\mathbf{A}w]t]$. ◁

Remark 3 ▷ Besides the four modes of construction stated in our Definition, Tichý in [Tichy88, pp. 63f] introduces two more modes of forming a construction — the *execution* and *double execution.* The idea of executions lies in that we need a way to distinguish between just referring to a construction and running or executing it to obtain whatever it constructs. The explanation is more clear in case of variables, e.g. $x \ldots \tau$. By the trivialization ^0x, we obtain the construction of the variable x, while by its execution ^1x we get what the construction constructs, i.e. the number assigned to x by a valuation. The double execution then means the execution of the result of execution, i.e. execution of a construction constructed by another construction.

By execution of a construction C we obtain the same construction ($^1C = C$) and by execution of a non-construction we get an undefined value. In case, we know that C is a construction, we may replace its execution with just C. Thus even in his book Tichý nearly never uses executions and in the following literature (as well as in this work) the notion of executions is usually not included in the definition of a construction. ◁

For the definition of a higher-order type we need to have at hand the notion of a construction of order n.

Definition 6 (construction of order n). *Let B be a base, $n \geq 1$ and α be a type of order n over B.*

(c$_n$-a) *Every variable ranging over α is a* construction of order n over B.

(c$_n$-b) *If A is an object of type α, then the trivialization $^0\mathsf{A}$ (also typeset as $\mathbf{A}$) is a* construction of order n over B.

(c$_n$-c) *If $m > 0$ and $X, Y_1, \ldots, Y_m$ are constructions of order n over B, then the composition $[XY_1 \ldots Y_m]$ is a* construction of order n over B.

(c_n-d) *If* $m > 0$*,* Y *and distinct variables* $x_1, \ldots, x_n$ *are constructions of order* n *over* B*, then the closure* $[\lambda x_1 \ldots x_n Y]$ *is a* construction of order n over B*.*

(c_n-e) *Nothing is a* construction of order n over B *unless it is specified so in* (c_n-a)*,* (c_n-b)*,* (c_n-c) *or* (c_n-d)*.*

□

Remark ▷ The attributes 'of order n' or 'over B' are left out in cases where the order of construction is not important or the base is the TIL objectual base. ◁

4.2.4 Higher-Order Type

Now, after we have the definitions of type of order 1 (simple type) and construction of order n ready, we can complete the definition of type with the specification of higher-order type.

Definition 7 (higher-order type). *Let* B *be a base and* $*_n$ *be the class of all constructions of order* n *over* B*.*

(t_{n+1}-a) $*_n$ *is a* type of order $n+1$ over B*.*

(t_{n+1}-b) *Every type of order* n *over* B *is a* type of order $n+1$ over B*.*

(t_{n+1}-c) *Let* $m > 0$ *and* $\alpha, \beta_1, \ldots, \beta_m$ *be types of order* $n+1$ *over* B*. Then the set* $(\alpha\beta_1 \ldots \beta_m)$ *of all (total and partial)* m*-ary mappings from* $\beta_1 \times \cdots \times \beta_m$ *into* α *is a* type of order $n+1$ over B*.*

(t_{n+1}-d) *Nothing is a* type of order $n+1$ over B *unless it is specified so in* (t_{n+1}-a)*,* (t_{n+1}-b) *or* (t_{n+1}-c)*.*

□

Remark 1 ▷ When we further talk about a type (without attributes), we mean a type of order n over the TIL objectual base for any natural number n. ◁

Remark 2 ▷ Another useful denotation system of higher-order types comes from [Hadacz2001]. In his classification, the higher-order types are denoted with $*_\xi$ which is the class of all constructions that construct an object of type ξ. Hence, the notation $C \ldots \xi$ (construction C which constructs a ξ-object) is equivalent to the notation of $C \ldots *_\xi$ (a variable/object of type $*_\xi$). Such denotation is advantageously applicable in the analysis of a NL expression where we often need to specify the type of the constructed object within a type of a compound. We might also take the type $*_\xi$ (esp. when used as an argument or return value type for a mapping) as a short for Tichý's $*_n$ with the extra condition on the type of the object constructed by the appropriate construction. ◁

4.2.5 Concept

The Tichý's expression-meaning relation diagram, as is depicted in the Figure 4.3a), was changed by Materna[35] by replacing the TIL central conception of construction on its place of the meaning sustainer with CONCEPT[36] (the Figure 4.3b)). In this section we summarize the definition of CONCEPT (and CONCEPT∗) and provide some comments to this approach.

Before we can define CONCEPT, we first need to explicate the term of free and bound variables.

Definition 8 (variable free/bound in a construction). *Let x be a variable and C a construction with at least one occurrence of x.*

(fv-a) *If C is x, then x is* free in C.

(fv-b) *If C is in the form of* ${}^0\mathsf{X}$, *then x is* 0-bound in C.

(fv-c) *If C is in the form of* $[XY_1 \ldots Y_n]$, *then x is* free in C *if and only if x is free in at least one of* X, $Y_1, \ldots, Y_n$.

(fv-d) *If C is in the form of* $[\lambda x_1 \ldots x_n X]$, *then x is* free in C *if and only if x is distinct from* $x_1, \ldots, x_n$ *and x is free in X.*

(fv-e) *If C is in the form of* $[\lambda x_1 \ldots x_n X]$, *then the variables* $x_1, \ldots, x_n$ *are* λ-bound in C *if and only if they are not 0-bound in X.*

(fv-f) *The variable x is* free, 0-bound *or* λ-bound in C *only in cases specified in* (fv-a), (fv-b), (fv-c), (fv-d) *or* (fv-e).

□

Definition 9 (CONCEPT∗). *A* CONCEPT∗ *is a construction that does not contain any free variables (a closed construction).* □

Remark ▷ We may also talk about a CONCEPT∗ *of order n*, in which case we refer to the order of the closed construction, i.e. we talk about a closed construction of order n. ◁

---

[35] see [Materna95, Materna98]

[36] we use the typesetting CONCEPT and CONCEPT∗ instead of Materna's 'concept' and 'concept∗' in order to clearly distinguish the places where we talk about a concept in Materna's approach.

We can see that CONCEPT* is just another name for a closed construction that is the meaning sustainer in Tichý's conception. However, Tichý never specified how to cope with *equivalent* constructions in his conception. An intuition says that if we talk about a certain concept (e.g. a winged horse), we do not take into account possible variants of it,[37] i.e. all constructions that are α-equivalent (identical up to renaming of λ-bound variables) or β-equivalent (construction extended by an unneeded composition and closure of the same variable) with each other.[38] Such constructions are called *quasi-identical* and in Materna's approach they form a CONCEPT.

Definition 10 **(**CONCEPT**).** *Let C and D be* CONCEPT*s∗. C and D are* quasi-identical *if and only if there exist* CONCEPT*s∗ $X_1, \ldots, X_n$ such that $X_1 = C$, $X_n = D$ and for $i = 1, \ldots, n-1$ every X_i and X_{i+1} are α- or β-equivalent.*

The equivalence relation of quasi-identity (for CONCEPT*s∗ of order n) is a $(o \ast_n \ast_n)$-object and is denoted Quid^n.*

The collection of all CONCEPT*s∗ is divided into classes of equivalence by the Quid^n relation. If C is* CONCEPT∗ *of order n and $\underline{C}$ the class of equivalence that contains C, then $\underline{C}$ is a* CONCEPT of order n generated by C. □

Remark 1 ▷ We can leave out the attribute 'of order n' in case the order of concept is not important. ◁

Remark 2 ▷ In [Materna98, pp. 97] the author provides a definition of CONCEPT that is actually different from our definition:

> Let c, d be variables ranging over $\ast_n$ for some n. A CONCEPT of order n is *the function constructed by* $\lambda c \lambda d[{}^0\mathsf{Quid}^n\ c\ d]$.

and continues

---

[37] if $\mathsf{WH}/(o\iota)_{\tau\omega}$ is the object of a winged horse, then there are infinitely many constructions that are equivalent not only in the generated object but also in the way they construct the object. Examples of such constructions are

$$\begin{array}{lll}
\mathbf{WH} & [\lambda w \mathbf{WH}_w] & [\lambda w_1 \mathbf{WH}_{w_1}] \\
 & [\lambda w \lambda t \mathbf{WH}_{wt}] & [\lambda w_1 \lambda t_1 \mathbf{WH}_{w_1 t_1}] \\
 & [\lambda w \lambda t \lambda x \mathbf{WH}_{wtx}] & [\lambda w_2 \lambda t_2 \mathbf{WH}_{w_2 t_2}] \\
 & \ldots & \ldots
\end{array}$$

[38] in [Materna95], the γ-equivalence (a singularizer $\imath x$ added in the form of $\imath x[x = C]$) is the third relation that forms the quasi-identity relation. In [Materna98] the γ-equivalent constructions are no more counted as quasi-identical because of the difference in the *content* (the new construction contains ${}^0I^\xi$ and ${}^0{=}$ in addition to the content of C). For more detailed definitions of α- and β-equivalence see [Materna98, pp. 93-96].

> In other words, a CONCEPT of order n associates every open construction of order n with an empty class, and every CONCEPT$*$ with the class of those CONCEPTs$*$ that are Quid-related with it.

By these words Materna defines a CONCEPT as a *function* that is able to assign the sought class of equivalence to every (closed) construction, i.e. to a CONCEPT$*$. Moreover, the construction that defines the concept function is β-equivalent with ${}^{0}\mathsf{Quid}^{n}$, thus the function is truly identical with the Quid^{n} relation.

However, on many other places Materna writes

> The relations Quid^{n} ... induce, for every CONCEPT$*$, an equivalence class. CONCEPTs can be construed as these equivalence classes.
>
> ...
>
> The CONCEPT$*$ ${}^{0}\mathsf{0}$ generates the singleton [CONCEPT] $\{{}^{0}\mathsf{0}\}$.

from where it is clear that CONCEPTs are classes. We thus suppose that a correct wording of Materna's definition of CONCEPT should be

> Let C be a CONCEPT$*$ and d be a variable ranging over $*_n$ for some n. A CONCEPT *of order n generated by C* is the *class of constructions* ($(o*_n)$-object) constructed by $\lambda d[{}^{0}\mathsf{Quid}^{n}\ C\ d]$.

In this form the definition is consistent with our Definition 10 and fully corresponds to the rest of Materna's book. ◁

A question remains whether this approach to concepts in the form of classes of equivalence is the right way to cope with the problem of quasi--identical constructions. On many places of his book[39] Materna provides claims about the nature of concepts which point out that a concept is very well represented by a construction since it provides many of the vital properties that the meaning sustainer must have.[40] On the other hand, a *class* certainly does not answer the purpose with the same qualities as the construction does. Materna partly overcomes this problem with shifting the required properties from the concept generator CONCEPT$*$ up to its container CONCEPT.

This however does not prevent us from encountering problems when we connect the meaning directly with a CONCEPT. It is rather counter

---

[39] e.g. [Materna98, pp. 83]

[40] one of the most important of them is the accordance with Frege's Functionality Principle.

intuitive to imagine that in the process of uttering an expression which represents a concept the speaker has in mind a whole class of constructions (CONCEPT). To resolve this problem with the corresponding CONCEPT$*$ does not provide a feasible solution, since it would require to change the schema in the Figure 4.3b) into a different one, where a construction (or CONCEPT$*$) would again gain its place on the meaning-way from expression to the identified object and the CONCEPT would either stay aside or perform a different function in the central diagram.

The whole conception of CONCEPT is inspired by two claims that may form a source of discrepancies in the original Tichý's expression-meaning relation:

($*$-a) when constructing a higher-order object (e.g. a propositional attitude), we cannot make a difference between quasi-identical constructions in place of the argument.

($*$-b) the meaning sustainer must follow the Functionality Principle (and other qualities that are reflected in the design of a construction).

To preserve the CONCEPT's conformability with the claim ($*$-a) while offering a better concordance with ($*$-b), we bring forward the following (sketchy) definition of *concept* via the notion of a *concept normal form* (CNF).

Definition 11 (concept normal form). *Let us suppose a fixed ordering of all types, i.e. let ξ_i be the i-th type over the TIL objectual base and let $\overset{\xi_i}{\bigvee}_j$ be the j-th variable of the type ξ_i for two natural numbers i and j.*

An α-normal form *of a construction C is the construction $\mathsf{NF}^{\alpha}(C)$ that ensues from construction C in the following way — the structure of the construction is exactly the same except that every free or λ-bound variable is consistently renamed to a first unused variable of the corresponding type (we parse the construction from left to right).*

A β-normal form *of a construction C is the construction $\mathsf{NF}^{\beta}(C)$, where for $n > 0$ there exist constructions $D_1, \ldots, D_n$ such that $D_1 = C$ and $D_n = \mathsf{NF}^{\beta}(C)$ and for each $i = 1, \ldots, n-1$ every D_i is reduced to its β-equivalent construction D_{i+1} and D_n cannot be reduced to other β-equivalent construction any more.*

A normal form *of a construction C is the construction $\mathsf{NF}(C)$ such that*

$$\mathsf{NF}(C) \overset{df}{\equiv} \mathsf{NF}^{\alpha}(\mathsf{NF}^{\beta}(C))$$

□

Remark 1 ▹ It is a trivial consequence of the definition that $\mathsf{NF}^{\alpha}(C)$ is α-equivalent to C and $\mathsf{NF}^{\beta}(C)$ is β-equivalent to C. ◃

Remark 2 ▹ Actually, CNFs represent a whole class of functions NF_i of type $(*_i *_i)$. Since the process of identification of a normal form does not change the order of the argument construction, and for the sake of better readability of the following text, we do not explicitly state the order of the type of the normal forms used in the text and we believe that it does not cause any misunderstandings. ◃

Before we can proceed to the definition of a concept that uses the just defined concept normal form, we need two auxiliary claims about important features of the normal forms.

Claim 1. *Let C, D be* CONCEPT*s∗ and let us denote the α-equivalence with $\overset{\alpha}{\cong}$. Then*

$$C \overset{\alpha}{\cong} D \quad \Leftrightarrow \quad \mathsf{NF}^{\alpha}(C) = \mathsf{NF}^{\alpha}(D)$$

Proof: Let us do the prove in the two directions of implications.

1. First let us prove that $C \overset{\alpha}{\cong} D \supset \mathsf{NF}^{\alpha}(C) = \mathsf{NF}^{\alpha}(D)$.
   If $C \overset{\alpha}{\cong} D$, then C and D differ only in consistently renamed λ-bound variables. Let $x \ldots \xi$ be the first λ-bound variable (from left to right) in C that corresponds to variable $y \ldots \xi$ in D and $x \neq y$. Then during the process of constructing the $\mathsf{NF}^{\alpha}(C)$ and $\mathsf{NF}^{\alpha}(D)$ both x and y are necessarily (consistently) replaced with the same variable $\bigvee^{\xi}_{n}$, since the left prefixes of both constructions parsed so far are identical and so is the type of both x and y. Inductively, every two corresponding distinct λ-bound variables in C and D are necessarily replaced with the same variable in the $\mathsf{NF}^{\alpha}(C)$ and $\mathsf{NF}^{\alpha}(D)$. Thus $\mathsf{NF}^{\alpha}(C) = \mathsf{NF}^{\alpha}(D)$.

2. Now it remains to prove that $\mathsf{NF}^{\alpha}(C) = \mathsf{NF}^{\alpha}(D) \supset C \overset{\alpha}{\cong} D$.
   If $\mathsf{NF}^{\alpha}(C) = \mathsf{NF}^{\alpha}(D)$ then $C \overset{\alpha}{\cong} \mathsf{NF}^{\alpha}(C) = \mathsf{NF}^{\alpha}(D) \overset{\alpha}{\cong} D$ and thus $C \overset{\alpha}{\cong} D$.

□

Claim 2. *Let C, D be* CONCEPT*s∗ and $\underline{C}$, $\underline{D}$ the appropriate classes of* Quid^{n} *equivalence. Then*

$$\underline{C} = \underline{D} \quad \Leftrightarrow \quad \mathsf{NF}(C) = \mathsf{NF}(D)$$

Proof: Again, we will prove the claim in two steps which will prove the implications in both directions.

1. $\underline{\mathrm{C}} = \underline{\mathrm{D}} \supset \mathsf{NF}(C) = \mathsf{NF}(D)$
   If $\underline{\mathrm{C}} = \underline{\mathrm{D}}$ then C and D are quasi-identical. Hence, there exist CONCEPTS∗ $X_1, \ldots, X_n$ such that $X_1 = C$, $X_n = D$ and for $i = 1, \ldots, n-1$ every X_i and X_{i+1} are α- or β-equivalent. Let us take any two neighbours C_i and C_{i+1} and think of the two cases where

   (a) $C_i \overset{\beta}{\cong} C_{i+1}$
   Then $\mathsf{NF}^\beta(C_i) = \mathsf{NF}^\beta(C_{i+1})$ and because NF^α is a mapping then $\mathsf{NF}^\alpha(\mathsf{NF}^\beta(C_i)) = \mathsf{NF}^\alpha(\mathsf{NF}^\beta(C_{i+1}))$, i.e. $\mathsf{NF}(C_i) = \mathsf{NF}(C_{i+1})$.

   (b) $C_i \overset{\beta}{\ncong} C_{i+1}$
   Thus $C_i \overset{\alpha}{\cong} C_{i+1}$ and since they are not β-equivalent, $C_i = \mathsf{NF}^\beta(C_i)$, $C_{i+1} = \mathsf{NF}^\beta(C_{i+1})$. From this it follows that $\mathsf{NF}(C_i) = \mathsf{NF}^\alpha(C_i)$ and $\mathsf{NF}(C_{i+1}) = \mathsf{NF}^\alpha(C_{i+1})$ and since according to the Claim 1 $\mathsf{NF}^\alpha(C_i) = \mathsf{NF}^\alpha(C_{i+1})$, thus $\mathsf{NF}(C_i) = \mathsf{NF}(C_{i+1})$.

   Hence, for any $i = 1, \ldots, n-1$, we have that $\mathsf{NF}(C_i) = \mathsf{NF}(C_{i+1})$, and thus also $\mathsf{NF}(C) = \mathsf{NF}(D)$.

2. $\mathsf{NF}(C) = \mathsf{NF}(D) \supset \underline{\mathrm{C}} = \underline{\mathrm{D}}$
   Let us suppose that $\mathsf{NF}(C) = \mathsf{NF}(D)$. This means that $\mathsf{NF}^\alpha(\mathsf{NF}^\beta(C)) = \mathsf{NF}^\alpha(\mathsf{NF}^\beta(D))$. Thus $\mathsf{NF}^\beta(C) \overset{\alpha}{\cong} \mathsf{NF}^\beta(D)$ and since both C and D are β-equivalent to their β-normal forms, we obtain a chain $C \overset{\beta}{\cong} \mathsf{NF}^\beta(C) \overset{\alpha}{\cong} \mathsf{NF}^\beta(D) \overset{\beta}{\cong} D$, which indicates that C and D are quasi-identical to each other and thus $\underline{\mathrm{C}} = \underline{\mathrm{D}}$.

□

According to the Claim 2, we know that the normal forms of CONCEPTS∗ are isomorphic with the corresponding classes of equivalence, viz CONCEPTS, so we can proceed to the following definition of concept.

Definition 12 (concept). *Let C be any* CONCEPT∗ *and let D be the* CONCEPT∗ *constructed by* $\mathsf{NF}(C)$*. We call D a* concept *and we say that C* points *to the* concept *D.* □

With this definition of a concept, we suppose that it follows both concept claims ($*$-a) and ($*$-b) and thus offers an acceptable (intuitive) solution to the explication of the notion of concept as the meaning sustainer in Materna's expression-meaning relation schema.

Chapter 5

Normal Translation Algorithm in TIL

In this section, we specify the particular steps for logical analysis of natural language, viz. Czech. The Normal Translation Algorithm (NTA) provides a way of describing the analysed meaning content of a sentence by means of the transparent intensional logic that was described and discussed in the previous chapter.

The first part of the algorithm,[1] lies in the syntactic analysis of Czech sentence. Its particular implementation is, as a part of this thesis, presented in the Chapter 3. During this part, we obtain syntactic derivation trees of the sentence ordered by their estimated probability. In this chapter, we always suppose in the description of the algorithm that we have already selected one (fixed) syntactic analysis encoded in one derivation tree.

The logical analysis itself consists in assigning the appropriate (sub)constructions to analysed (sub)constituents by employing the *lexicon* and in the *type checking* which makes it possible to prune the contingencies that cannot be resolved on a lower level of the derivation tree.

A special attention is payed to the central expressive device in the sentence, the *verb phrase*. The analysis of verb tenses is based on Tichý's articles [Tichy80b, Tichy80a] and their summarization in [Koukol88]. In this work, the approach is put into concordance with the temporal TIL and the extended type theory, as presented in the Chapter 4 as well as with the analysis of other language phenomena. Some inspiration for the analysis was also taken from [Tichy94b, Tichy94a], e.g. the analysis of quantifica-

---

[1] however, not necessarily detached from the other parts in code or time during the implementation.

tional phrases. The algorithm as a whole is an original part of this work and its usability is demonstrated by a partial implementation described in the Chapter 7. In the design of the algorithm, we have concentrated on the most frequent phenomena appearing in the analysis of Czech sentence. Hence, the presented algorithm needs to be seen as the first version that tries to systematically cover the spread of a natural language, with the proviso that some points have to be postponed until the successive versions.

Since, in this chapter, we present the algorithm with regard to the Czech language, the examples of NL expression will be stated in Czech with the English (verbatim) translation in a footnote.

5.1 Verb Phrase

In a sentence, the verb phrase (with the verb in its finite form) represents the predicative skeleton of every clause. In our syntactic analysis, the particular verb group is always represented by a '`clause → ...`' rule, which enables us to process it consistently in one place.[2] Thus, in this section, we can concentrate on how to capture the meaning of a single clause with one verb group. We start with the simplest case, the verb in present tense, and further describe how to cope with other tenses, verb aspects and various kinds of verb meaning modificators.

Tichý comes with a dissociation of significant verbs into two groups according to the classification of their meaning:

1. *attributive verbs* express what *qualities* the attributed objects *have* or what the objects *are*. An example of such verb is in the sentence

   Radnice stojí na náměstí.[3] (5.1)

   The appropriate analysis of the attributive verbs lies in a proposition that ascribes the alluded property to the subject. A simplified analysis of (5.1), which does not take into account the complex character of

---

[2] some Prague School researchers propose to simplify the analysis by cutting the verb group rules into several levels of the derivation tree, i.e. into different grammar rules. This could possibly decrease the number of rules needed for capturing the language syntax, however, it would also drastically increase the complexity of the logical analysis of the resulting verb phrase.

[3] 'The town hall stands in the square.'

the predicate 'stojící na náměstí'[4] and which regards 'radnice'[5] as an individual, may look like this

$$\lambda w \lambda t[\textbf{stojící\_na\_náměstí}_{wt}\textbf{radnice}] \tag{5.1'}$$

Attributive verbs are typically expressed in the form of the verb 'být'[6] in combination with a property, usually as an adjective such as 'být červený' or 'být zralý'[7].

2. *episodic verbs*, on the other hand, express *actions*, they tell what the subject *does*. As an example, we may instance the sentence 'Kočka chodí po střeše.'[8] In this sentence the verb 'chodit'[9] cannot be included in the cat's state description in any moment of time, it rather describes an *episode* of walking that is practiced by the cat at the certain moment (and necessarily some time before that moment plus the expectation that it will last also in the next few moments, at least).

The main difference between attributive and episodic verbs consists in their time consumption — the attributive verbs do not take the time dimension into account, they just describe the *state* of the subject in the very one moment by saying that the subject has (or has not) a certain property.

5.1.1 Episodic Verb

The episodic verbs bring much more complications with their analysis than the attributive verbs. In this section, we will present the approach to logical analysis of the episodic verbs by means of events and episodes.

5.1.1.1 Events

In the episodic verbs conception, an *event* can be described as a snapshot of an action. In order to offer an explication of such snapshot, we define it as conjunction of specific propositions (atomic assertions) together with the time specification of the whole process. An atomic assertion, called a basic proposition, is then defined with the use of *primary properties*, viz. the members of the TIL intensional base.

---

[4] 'standing in the square'

[5] 'town hall'

[6] 'to be'

[7] 'to be red', 'to be mature'

[8] 'A cat walks on the roof.'

[9] 'to walk'

Definition 13 (primary property). *Let ξ be a type. Then the members of the intensional base that form the class* Pr^{ξ} *of ξ-properties*[10] *are called the* primary ξ-properties. *Similarly, let* $\mathsf{Pr}^{\xi_1,\ldots,\xi_n}$ *denote a class of* primary $(\xi_1,\ldots,\xi_n)$-relations.[11] □

Remark ▷ According to the definition of the intensional base, we can say that primary properties are those properties, whose change is an atomic process, i.e. the change of one property is not a necessary reason of (and is not inevitably caused by) a change of any other primary property. ◁

E X A M P L E Typical primary properties are features like color, height or absolute position (being white, being 2 meters tall). On the other hand, a property like being next to a white box cannot be counted among the primary properties, since it necessarily depends on the color of another object — Tichý calls it a parasitic property. □

Definition 14 (basic proposition). *Let P be a proposition (a $o_{\tau\omega}$-object denoted also as a π-object). We call P a* basic proposition *if and only if it can be generated by a construction of the form* $\lambda w\lambda t[\mathbf{R}_{wt}X_1\ldots X_n]$ *or* $\lambda w\lambda t[{}^0\neg[\mathbf{R}_{wt}X_1\ldots X_n]]$ *, where* $\mathsf{R}\in\mathsf{Pr}^{\xi_1,\ldots,\xi_n}$ *and $X_1,\ldots,X_n$ are objects of the types $\xi_1,\ldots,\xi_n$. Let also* Ba *denote a class of all basic propositions.* □

For the definition of an event, we need several auxiliary functions that are defined in the Table 5.1.[12] Now, nothing is hindering us in defining an event.

Definition 15 (event). *An* event *is a conjunction of a class of propositions that consists of exactly one time-proposition and a number of shifts of basic propositions. The characteristic function* $\mathsf{Ev}/o\pi$ *of the class of all events is defined as*[13]

---

[10] i.e. $(o\xi)_{\tau\omega}$-objects

[11] i.e. $(o\xi_1\ldots\xi_n)_{\tau\omega}$-objects

[12] So as to offer a mathematical explication of the presented functions, we use the $\overset{df}{\Leftrightarrow}$ operator with the same meaning as in [Tichy80a]. By saying that $\mathbf{A}\,b\,c \overset{df}{\Leftrightarrow} D$, we mean that the two constructions $\mathbf{A}\,b\,c$ and D are equivalent in the sense of constructing the same object for the same assignments of the values to the free variables b and c. By this explication, we do not define the left side as the short for the right side, we rather rigorously specify which objects are constructed by $\mathbf{A}$.

[13] The original Tichý's definition of an event (see [Tichy80b, pp. 273]) differed from ours in one part — we have added the singularizer $p = \imath q[\,e_q \wedge q = \mathbf{Tm}_t]$. This adding safeguards the fact that an event has exactly one time-proposition. The previous definition

Func.	Definition	Description
$\mathsf{Tm}/\pi\tau$	$[\mathbf{Tm}\ t_0]_{wt} \overset{df}{\Leftrightarrow} t = t_0$ $(t, t_0 \ldots \tau; w \ldots \omega)$	Let $t_0 \in \tau$ be a time moment. The proposition saying 'The current time is t_0.' is called a *time-proposition.* Let Tm be the function of type $\pi\tau$ that takes every time moment to the corresponding time-proposition.
$\mathsf{Sh}/\pi\pi\tau$	$[\mathbf{Sh}\ p\ k]_{wt} \overset{df}{\Leftrightarrow} p_{w[t+k]}$ $(p \ldots \pi; k \ldots \tau)$	Let P be a proposition and $k \in \tau$ be a number. The proposition saying that P will be true in k seconds is called the *k-shift of P.* Let Sh denote the function which takes any proposition P and number k to the k-shift of P.
$\mathsf{Cj}/\pi(o\pi)$	$[\mathbf{Cj}\ e]_{wt} \overset{df}{\Leftrightarrow} (\forall p)[e_p \supset p_{wt}]$ $(e \ldots o\pi; p \ldots \pi)$	Let E be a class of propositions. The proposition saying that at world w and time t all the propositions from E are true is called the *conjunction* of E. Let Cj be the function that takes every class of propositions to their conjunction.

Table 5.1: Auxiliary functions needed for the definition of an event.

$$\begin{aligned}
\mathbf{Ev}\ c \overset{df}{\Leftrightarrow} (\exists e)\Big[& c = \mathbf{Cj}_e \wedge \\
& \wedge (\exists t)(\forall p)\Big[e_p \supset \big[p = \imath q[e_q \wedge q = \mathbf{Tm}_t] \vee \\
& \qquad \vee (\exists b)(\exists k)[\mathbf{Ba}_b \wedge p = \mathbf{Sh}_{kb}]\big]\Big]\Big]
\end{aligned}$$

$$(c, b, p, q \ldots \pi; e \ldots o\pi; k \ldots \tau)$$

□

---

allowed an absurd event with none or more than one time specifications. This discrepancy was probably also discovered by Koukolíková, since in [Koukol88, pp. 26] she adds a note that Tichý's definition of an event does not fully correspond to his informal definition. Koukolíková also offers a replacement for the definition, however, it does not properly solve the problem.

Func.	**Definition**	**Description**
Rg/ $(o\tau)\pi$	$\textbf{\textsf{Rg}}\ c \overset{df}{\Leftrightarrow} \lambda t(\exists t_0)\Big[c \Rightarrow \textbf{\textsf{Tm}}_{t_0} \wedge (\exists b)\big[\textbf{\textsf{Ba}}_b \wedge c \Rightarrow \textbf{\textsf{Sh}}\, b\, [t - t_0]\big]\Big]$ $(c, b \ldots \pi; t, t_0 \ldots \tau)$	The collection of those time moments, in which the event's basic propositions must be true for the event to be true, is called the *temporal range* of the event. Let $\textsf{Rg}$ be the function that takes events to their ranges.
Due/ $(o\pi\pi)_{\tau\omega}$	$\textbf{\textsf{Due}}_{wt}\ p\ c \overset{df}{\Leftrightarrow} \textbf{\textsf{Ev}}_c \wedge c_{wt} \wedge c \Rightarrow p$ $(p, c \ldots \pi)$	Let p be a proposition and c an event. We say that p is true in world w and time t *due to* c if and only if c is true at w, t and c implies p. Let $\textsf{Due}$ denote the due-to-relation between propositions and events.
Hab/ $(o(o\tau)\pi)_{\tau\omega}$	$\textbf{\textsf{Hab}}_{wt}\ s\ p \overset{df}{\Leftrightarrow} (\exists c)\big[[\textbf{\textsf{Due}}_{wt} c\, p] \wedge s = [\textbf{\textsf{Rg}}\, c]\big]$ $(s \ldots o\tau; p, c \ldots \pi)$	Let p be a proposition and c an event and let p be true in w at t due to c. Further let $s = [\textbf{\textsf{Rg}}\, c]$. Then we call s the *temporal habitat* of p at w, t and denote the appropriate relation $\textsf{Hab}$.
Sg/$\pi\pi\tau$	$\textbf{\textsf{Sg}}\ c\ t \overset{df}{\Leftrightarrow} \textbf{\textsf{Cj}}\, \lambda p(\exists t_0)\Big[c \Rightarrow \textbf{\textsf{Tm}}_{t_0} \wedge \Big[p \Rightarrow \textbf{\textsf{Tm}}_{t_0} \vee (\exists b)(\exists k)\big[\textbf{\textsf{Ba}}_b \wedge [t_0 + k \leq t] \wedge p \equiv \textbf{\textsf{Sh}}\, k\, b \wedge c \Rightarrow p\big]\Big]\Big]$ $(c, p, b \ldots \pi; t, t_0, k \ldots \tau)$	Let c be an event, t a time moment. Let d be the event which is the same as c except those shifts of basic propositions from c that exceed the time moment t. Then d is called a *t-segment of c*. Moreover, d is called a *proper segment* of c if $d \neq c$.

Table 5.2: Functions representing important properties of or operations over events.

In [Tichy80b], the author defines several operations that work with events and represent important properties of propositions and events. Since we will use those operations in the following text, we present their definitions in a synopsis in the Table 5.2.

5.1.1.2 Episodes

The relationship between an event and an episode can be liken to the relationship between a snapshot and a video sequence. An episode consists of specific events — such that fully describe the whole action in every moment in which the action runs, i.e. not only the part of the action that is true at the specific moment, but also all other parts of it including those that may precede the moment and those that will follow.

Definition 16 (episode). *Let c, d be events. We call them (mutually)* variant *if and only if c and d differ only in their time-proposition. Let* $\mathsf{Var}/o\pi\pi$ *be the function-relation between variant events:*

$$\begin{aligned}\textbf{\textsf{Var}}\ c_1\ c_2 \overset{df}{\Leftrightarrow} \textbf{\textsf{Ev}}_{c_1} \wedge \textbf{\textsf{Ev}}_{c_2} \wedge (\exists t_1)(\exists t_2)\Big[&[c_1 \Rightarrow \textbf{\textsf{Tm}}_{t_1}] \wedge [c_2 \Rightarrow \textbf{\textsf{Tm}}_{t_2}] \wedge \\ &\wedge (\forall b)(\forall k)\Big[\textbf{\textsf{Ba}}_b \supset \big[[c_1 \Rightarrow \textbf{\textsf{Sh}}\, k\, b] \Leftrightarrow [c_2 \Rightarrow \textbf{\textsf{Sh}}\, [k + t_1 - t_2]\, b]\big]\Big]\Big]\end{aligned}$$

$$(c_1, c_2, b \ldots \pi; t_1, t_2, k \ldots \tau)$$

A class of events that are pairwise variants is called an episode. *Let* $\mathsf{Ep}/o(o\pi)$ *denote the class of all episodes:*

$$\textbf{\textsf{Ep}}\ e \overset{df}{\Leftrightarrow} (\forall c_1)(\forall c_2)\big[[e_{c_1} \wedge e_{c_2}] \supset \textbf{\textsf{Var}}\ c_1\ c_2\big]$$

$$(c_1, c_2 \ldots \pi; e \ldots o\pi)$$

Let w be a possible world and t a time. We say that an episode occurs *in w if and only if all events that form the episode are true in w at some time. Let* $\mathsf{Occ}/(o(o\pi))_\omega$ *be the function that assigns to a world the class of episodes that occur in it:*

$$\textbf{\textsf{Occ}}_w\ e \overset{df}{\Leftrightarrow} \textbf{\textsf{Ep}}\, e \wedge (\forall c)\big[e_c \supset (\exists t) c_{wt}\big]$$

$$(c \ldots \pi; e \ldots o\pi)$$

By the episode's running time*, we understand the collection of time moments in which the episode is in progress. Let* $\mathsf{Ru}/(o\tau)(o\pi)$ *be the function that assigns to an episode its running time:*

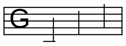
$$\mathbf{Ru}\ e \overset{df}{\Leftrightarrow} \lambda t(\exists c)[e_c \wedge c \Rightarrow \mathbf{Tm}_t]$$

$$(e \ldots o\pi; c \ldots \pi)$$

□

EXAMPLE

Let us have a look at an example of an episode — 'brnkání na kytaru'[14]. In order to form a particular episode we may, for instance, take the sentence

> Petr brnká na kytaru melodii  v časovém intervalu od t_1 do t_2.[15] (5.2)

Let T_1, T_2, T_3 denote the relative times of Peter's playing the first, second and third tone of the tune. Then the episode (5.2) will be the class of events of the form

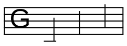
> Je právě čas t ∧
> za $(t_1 + T_1 - t)$ sekund Petr zahraje tón 1 ∧
> za $(t_1 + T_2 - t)$ sekund Petr zahraje tón 2 ∧
> za $(t_1 + T_3 - t)$ sekund Petr zahraje tón 3.[16] (5.3)

where $t \in <t_1; t_2>$ and if $(t_1 + T_i - t) \leq 0$ then the corresponding proposition changes to 'právě hraje' or 'před ... sekundami zahrál.'[17]□

An important part of an episode is its protagonist (or protagonists), the leading actor. The fact that an individual is the leading actor of an episode is necessarily contained in the episode in the way that each basic proposition within events of the episode is an affiliation of a certain property to the individual. We call this a *by*-relation between episodes and individuals and express it with function $\mathsf{By}/o\iota(o\pi)$

---

[14] 'strumming a guitar'

[15] 'Peter is strumming a tune  at a guitar at the time interval from t_1 to t_2.'

[16] 'The time is exactly t ∧ within $(t_1 + T_1 - t)$ seconds Peter plays the tone 1 ∧ within $(t_1 + T_2 - t)$ seconds Peter plays the tone 2 ∧ within $(t_1 + T_3 - t)$ seconds Peter plays the tone 3.'

[17] 'is just strumming' or 'has strummed ... seconds ago'

$$\begin{aligned}\textbf{\textsf{By}}\ x\ e \overset{df}{\Leftrightarrow} \textbf{\textsf{Ep}}_e \wedge (\forall c)(\forall b)(\forall k)\Big[[e_c \wedge \textbf{\textsf{Ba}}_b \wedge c \Rightarrow \textbf{\textsf{Sh}}\, k\, b] \supset\\ \supset (\exists q)\big[\textbf{\textsf{Pr}}^{\iota}\, q \wedge b = \lambda w \lambda t[q_{wt}\, x]\big]\Big]\end{aligned}$$

$$(x \ldots \iota; e \ldots o\pi; c, b \ldots \pi; k \ldots \tau; q \ldots (o\iota)_{\tau\omega})$$

A similar relation (with function $\textsf{Byp}/o(o\iota)(o\pi)$) is also defined for the plural *by*-linkage between episodes and classes of individuals

$$\begin{aligned}\textbf{\textsf{Byp}}\ z\ e \overset{df}{\Leftrightarrow} \textbf{\textsf{Ep}}_e \wedge (\forall c)\Bigg[e_c \supset \Bigg[\Bigg[(\forall b)(\forall k)\Big[[\textbf{\textsf{Ba}}_b \wedge c \Rightarrow \textbf{\textsf{Sh}}\, k\, b] \supset\\ \supset (\exists q)(\exists x)\big[z_x \wedge \textbf{\textsf{Pr}}^{\iota}\, q \wedge b = \lambda w \lambda t[q_{wt}\, x]\big]\Big]\Bigg] \wedge\\ \wedge (\forall x)\Big[z_x \supset (\exists q)(\exists k)\Big[\textbf{\textsf{Pr}}^{\iota}\, q \wedge c \Rightarrow \textbf{\textsf{Sh}}\, k\, \lambda w \lambda t[q_{wt}\, x]\Big]\Big]\Bigg]\Bigg]\end{aligned}$$

$$(z \ldots o\iota; e \ldots o\pi; c, b \ldots \pi; k \ldots \tau; q \ldots (o\iota)_{\tau\omega}; x \ldots \iota)$$

An action is thus led *by* a group of individuals if each event of the episode consists of affiliation of a property to a member of this class.

Now, we can eventually define the central function of the logical analysis of a verb phrase — the *does*-relation. We say that an individual is in the *does*-relation to a class of episodes if the individual currently takes the leading role in one of the episodes. We define two functions $\textsf{Does}/(o\iota(o(o\pi)))_{\tau\omega}$ and $\textsf{Do}/(o(o\iota)(o(o\pi)))_{\tau\omega}$ that represent the does-relation and its plural counterpart:

$$\begin{aligned}\textbf{\textsf{Does}}_{wt}\ x\ u &\overset{df}{\Leftrightarrow} (\exists e)\big[u_e \wedge \textbf{\textsf{By}}\, x\, e \wedge \textbf{\textsf{Occ}}_w e \wedge \textbf{\textsf{Ru}}\, e\, t\big]\\ \textbf{\textsf{Do}}_{wt}\ z\ u &\overset{df}{\Leftrightarrow} (\exists e)\big[u_e \wedge \textbf{\textsf{Byp}}\, z\, e \wedge \textbf{\textsf{Occ}}_w e \wedge \textbf{\textsf{Ru}}\, e\, t\big]\end{aligned}$$

$$(x \ldots \iota; z \ldots o\iota; u \ldots o(o\pi); e \ldots o\pi)$$

Remark ▷ The does-relation links individuals to classes of episodes and not to single episodes in order to reflect the subject-predicate relation in a clause. In the sentence ‘Petr zpívá,’[18] we do not talk about a particular episode of Peter’s singing, we rather mean that Peter is now playing the leading role in one of *all* the singing episodes. Thus, for the sentence to be true, we mean that Peter bears the does-relation to the class of currently happening singing episodes. ◁

---

[18] ‘Peter sings.’

EXAMPLE

We are now fully equipped for a logical analysis of a present tense indicative clause with an episodic verb. Let us take the above mentioned sentence

$$\text{Petr zpívá.} \tag{5.4}$$

Here, the expression 'zpívá'[19] identifies the class of those episodes that express singing. However, what counts to singing is not an invariable fact, it depends on the respective world.[20] Thus, the expression 'zpívá' denotes not a particular class of episodes, but rather a special property of episodes — a function of type $(o(o\pi))_\omega$ that assigns the class of singing episodes to every world. Let zpívá be such function. Then the sentence (5.4) is analysed as

$$\lambda w\lambda t[\textbf{\textsf{Does}}_{wt}\textbf{\textsf{Petr}}\ \textbf{\textsf{zpívá}}_{w}] \tag{5.4'}$$

The proposition constructed by this construction is true in world w and time t if Petr is in the *does*-relation to the class of episodes returned by the function zpívá at w, i.e. Petr is the leading actor of one such episode and that episode is in progress in time t. □

5.1.1.3 Verbal Object

Verbs are very often not describing just one (kind of an) episode. Let us take the verb 'zničit krtka.'[21] If we want to analyse a particular case where Peter kills off the mole, then we cannot simply specify one kind of episodes covering killing off the mole — we need to differentiate the episodes that have as their result the killing off the mole from those episodes that involve the mole's dying.

Definition 17 (verbal object). *Every episodic verb expresses a relationship between the episode that forms the observable part of the action, called a* labour episode*, and the achieved state, called an* upshot episode*. Thus, an episodic verb is a world-dependent linkage between a labour episode and an upshot episode, i.e. an object of type* $(o(o\pi)(o\pi))_\omega$*, called a* verbal object. □

---

[19] 'sings'

[20] For instance, if a heavy-metal singer performed his show in the world of the 18th century, nobody would ever dare to count it as 'singing.'

[21] 'to kill off the mole'

Verbal object	**Labour episode**	**Upshot episode**
zničit krtka	putting a poison into the mole's tunnel, flooding the tunnel with water, ...	the mole's death
zpívat Ódu na radost*	the singing of the Ode to Joy	the singing of the Ode to Joy
potěšit Marii*	writing a letter to Mary, singing a song, tackling Mary, ...	Mary's being pleased
chodit*	a series of completed steps	the same series of completed steps

* 'to sing the Ode to Joy', 'to please Mary', 'to walk'

Table 5.3: Examples of verbal objects with labour and upshot episodes.

EXAMPLE A description of a few episodic verbs with their labour and upshot episodes is displayed in the Table 5.3. As a matter of fact, the labour and upshot episodes that are stated in the table represent only particular instances of such episodes. The actual class of all the appropriate labour and upshot episodes is often much larger, e.g. the number of ways of 'pleasing Mary' is infinite, since Mary could be potentially pleased by any kind of behaviour on the side of the protagonist. □

In [Tichy80b], the author differentiates the verbal objects according to their labour and upshot episodes to *achievement* verbs (whose labour episode and upshot episode are materially disjoint, e.g. 'zabít králíka' or 'uvařit oběd'[22]) and *performance* verbs (whose labour episode does not differ from the upshot episode, e.g. 'zpívat Ódu na radost' or 'chodit'[23]).

5.1.1.4 Verb

The verbal objects we have investigated are expressed either by simple expressions — 'chodit' or 'zpívat,'[24] or by a compound of a verb phrase with its arguments like 'zničit krtka' or 'potěšit Marii.'[25] The first expressions express directly the verbal $(o(o\pi)(o\pi))_\omega$-objects. The other expressions have

---

[22] 'to kill a rabbit', 'to make a dinner'

[23] 'to sing the Ode to Joy', 'to walk'

[24] 'to walk', 'to sing'

[25] 'to kill off the mole', 'to please Mary'

as their counterpart a construction consisting of the *transitive* verb and its argument(s).

Definition 18 (verb). *In general, a* verb *expression identifies an object of type* $(o(o\pi)(o\pi))_\omega\xi_1\ldots\xi_n$*, where* $n \geq 0$ *and* $\xi_1,\ldots,\xi_n$ *are types of the verb arguments. Thus for* $n = 0$ *a verb is directly an* $(o(o\pi)(o\pi))_\omega$*-object, otherwise a verb is a function that assigns a verbal object to its arguments.* □

E X A M P L E

Let zničit be the $(o(o\pi)(o\pi))_\omega\iota$-object of the verb 'zničit' (to kill off) and Krtek/ι be the particular individual mole that has to be destroyed. Then the verb phrase 'zničit krtka' is analysed as the construction of a verbal object [**zničit Krtek**] $\ldots (o(o\pi)(o\pi))_\omega$. Thus, the TIL type of a particular verb phrase depends on the *verb frame* schema that is instantiated in the sentence. □

5.1.2 Verb Aspect

With the specification of other verb categories, we are now going deeper into the explication of how a verb construction may look like. As we have seen, verbs denote objects of type $(o(o\pi)(o\pi))_\omega$ and the present indicative expression like 'zpívá' (sings) from the sentence (5.4) denote objects of type $(o(o\pi))_\omega$.

Most verbal objects can appear in one of the two aspects — the *imperfective* and *perfective* aspect. An aspect is thus best construed as (a world-dependent) function from $(o(o\pi)(o\pi))$-objects into $(o(o\pi))$-objects.

The main difference between the two verb aspect is as follows — if V is a verbal object and someone is connected with an imperfective of the verb V, then it means that the person is *engaged* in V-ing, and if he is connected with a perfective of the verb V, then he has just *completed* V-ing.

Definition 19 (imperfective aspect). *Let s be an interval of time and e an episode. By an s* transform *of e, we understand the episode which is just like e except that its running time is s. Let* Trs$/(o\pi)(o\tau)(o\pi)$ *be the transform function:*

$$\textbf{Trs}\ s\ e \overset{df}{\Leftrightarrow} \lambda c\Big[\textbf{Ev}_c \wedge (\exists t)\big[s_t \wedge [c \Rightarrow \textbf{Tm}_t] \wedge (\exists c_1)[e_{c_1} \wedge \textbf{Var}\, c_1\, c]\big]\Big]$$

$$(s \ldots o\tau; e \ldots o\pi; c, c_1 \ldots \pi)$$

Let V be a $(o(o\pi)(o\pi))$-object and w a world. If a class E of episodes is the value of the imperfective *operation* $\mathsf{Imp}/((o(o\pi))(o(o\pi)(o\pi)))_\omega$ *in w at V, then it means that for each $e \in E$ there exist two episodes l and u that occur in w and are linked by V and e is the s transform of l, where s is the intersection of l's and u's running times.*

$$\textbf{\textsf{Imp}}_w\ v \overset{df}{\Leftrightarrow} \lambda e\Big[(\exists l)(\exists u)\Big[\textbf{\textsf{Occ}}_w\, l \wedge \textbf{\textsf{Occ}}_w\, u \wedge \wedge\, v\, l\, u \wedge e = \textbf{\textsf{Trs}}\big[\lambda t[\textbf{\textsf{Ru}}\, l\, t \wedge \textbf{\textsf{Ru}}\, u\, t]\big]\, l\Big]\Big]$$

$$(v \ldots o(o\pi)(o\pi); e, l, u \ldots o\pi)$$

□

EXAMPLE

Now, we can shed more light to the relationship between the object $\mathsf{zpívá}/(o(o\pi))_\omega$ (see the analysis (5.4')) representing the expression '$\mathsf{zpívá}$' (sings) and the appropriate verbal object $\mathsf{zpívat}/(o(o\pi)(o\pi))_\omega$.[26] With the use of the Imp function we can construe the object $\mathsf{zpívá}$ with the construction $\lambda w[\textbf{\textsf{Imp}}_w\textbf{\textsf{zpívat}}_w]$. So that (5.4) in the imperfective aspect depicts $\lambda w\lambda t\big[\textbf{\textsf{Does}}_{wt}\textbf{\textsf{Petr}}\,[\lambda w[\textbf{\textsf{Imp}}_w\textbf{\textsf{zpívat}}_w]\, w]\big]$ which after the reduction gives

$$\lambda w\lambda t\big[\textbf{\textsf{Does}}_{wt}\textbf{\textsf{Petr}}\,[\textbf{\textsf{Imp}}_w\textbf{\textsf{zpívat}}_w]\big] \qquad (5.4")$$

□

Definition 20 (perfective aspect). *First, we need to define the function* $\mathsf{End}/(o\tau)(o\tau)$ *which for a time interval returns its subinterval consisting only of its last point*

$$\textbf{\textsf{End}}\ s \overset{df}{\Leftrightarrow} \lambda t\Big[\neg(\exists t_1)\big[s_{t_1} \wedge t < t_1\big] \wedge (\forall t_1)\big[t_1 < t \supset (\exists t_2)[s_{t_2} \wedge t_1 < t_2 \leq t]\big]\Big]$$

$$(s \ldots o\tau; t, t_1, t_2 \ldots \tau)$$

Let V be a $(o(o\pi)(o\pi))$-object and w a world. If a class E of episodes is the value of the perfective *operation* $\mathsf{Perf}/((o(o\pi))(o(o\pi)(o\pi)))_\omega$ *in w at V, then it means that for each $e \in E$ there exist two episodes l and u that*

---

[26] '$\mathsf{zpívat}$' = 'to sing'

occur in w and are linked by V and e is the s transform of l, where s is the singleton of the last point of u's running time.

$$\textsf{\textbf{Perf}}_w\ v \overset{df}{\Leftrightarrow} \lambda e\Big[(\exists l)(\exists u)\Big[\textsf{\textbf{Occ}}_w\, l \wedge \textsf{\textbf{Occ}}_w\, u \wedge v\, l\, u \wedge \\ \wedge\, e = \textsf{\textbf{Trs}}\big[\textsf{\textbf{End}}[\textsf{\textbf{Ru}}\, u]\big]\, l\Big]\Big]$$

$$(v \ldots o(o\pi)(o\pi); e, l, u \ldots o\pi)$$

□

5.1.3 Verb Tense

The contemporary Czech language has three verb tenses — the present tense, the past tense and the future tense. However, we cannot simply convert the Czech tenses one to one to the corresponding three tenses in English (for which Tichý defined the appropriate functions). We can probably adopt the thorough definition of the past and future tenses, but the present tense requires more introspection then it was accorded e.g. in [Koukol88].

5.1.3.1 Present Tense

Unlike English the Czech verbs have the capability of expressing the verb aspect built directly into their grammatical category — every verb is exactly in either the imperfective or the perfective form. A special property that goes along with the perfective aspect is that these verbs do not form the present tense — they have only the past tense and the future tense forms.[27] The present tense of the perfective verbs is usually expressed by their imperfective counterparts.

On the other hand, the Czech present tense sentence can often have one of two quite different readings. Let us take the sentence

Petr nakupuje v supermarketu.[28] (5.5)

This sentence can mean either that

Peter is shopping in a supermarket right now. (5.5a)

---

[27] e.g. 'začal/začne' (he started/he will start), 'zabil/zabije' (he killed/he will kill) or 'udělal/udělá' (he did/he will do).

[28] 'Peter goes shopping into a supermarket.'

or we can understand it as

Peter usually goes shopping into a supermarket. (5.5b)

Without any auxiliary mean of directing us towards one of those two readings, we have no possibility of preferring either of it. However, the speaker may help the analysis with supply of (auxiliary) adverbs like 'právě', 'teď'[29] for the reading (5.5a) and 'obvykle', 'vždycky' or 'pravidelně'[30] for the respective reading (5.5b).

As we can see at the first (English) reading of the sentence (5.5), the Czech present tense with the meaning of 'is doing now' shall be analysed not as the English (simple) present tense, but rather as the present progressive tense.

The main difference between the simple and progressive forms from the logical point of view is the dependence of the truthness of the appropriate proposition on the finishing of the whole episode. While the progressive form sentence is either true or false at the moment, the truthness of the (imperfective) present tense postulates that the whole action will be finished. For instance, the English sentence 'John gets up' (with the imperfective reading 'John is involved in getting up') is true at the moment t_0 only in case that at a moment $t_1 \geq t_0$ John will finish his getting up with being up. If, however, in the middle of his getting up John thought better of it and remained lying, then we could not talk about John being involved in getting up at t_0 since there would be (in the actual world) no getting up in which John could take a part at that moment.

Hence, we need a mean of capturing the *modal dimension* of the progressive tense — a sentence in the progressive is true or false no matter whether the mentioned action really finishes in the reference world or not. For the sentence to be true it is sufficient that the corresponding action finishes in another possible world that is the same as the reference world up to the events that were the cause of the cancellation of the action in progress.

Definition 21 (progressive). *The progressive is construed as a property of propositions. Let p be a proposition, w a world and t a time moment. We say that p is* in progressive *in w at t if and only if there is*

1. *an interval $<t_1; t_2>$ of time that includes t,*
2. *an event c whose t-segment is true in w and*

---

[29] 'at the moment', 'now'

[30] 'usually', 'always', 'regularly'

3. *a world w_1 in which it is due to c that p is true throughout the whole interval $<t_1; t_2>$.*

Let $\mathsf{Prog}/(o\pi)_{\tau\omega}$ *denote such property:*

$$\mathbf{Prog}_{wt}\ p \overset{df}{\Leftrightarrow} (\exists t_1)(\exists t_2)\Bigg[[t_1 \leq t < t_2] \wedge (\exists c)\Bigg[[\mathbf{Sg}\, c\, t]_{wt} \wedge \\ \wedge (\exists w_1)\Big[\mathbf{Due}_{w_1 t}\Big[\lambda w \lambda t (\forall t_0)\big[[t_1 \leq t_0 < t_2] \supset p_{wt_0}\big]\ c\Big]\Big]\Bigg]\Bigg]$$

$$(p, c \ldots \pi; t, t_0, t_1, t_2 \ldots \tau; w, w_1 \ldots \omega)$$

□

The logical analysis of Czech present tense in the reading ‘is doing now’ is then achieved by means of the Prog function on the place of a modificator of the imperfective sentence. As an example, the analysis of the sentence (5.5) in the reading (5.5a) can look like

$$\lambda w \lambda t\Bigg[\mathbf{Prog}_{wt} \lambda w \lambda t (\exists x)\Big[\Big[\mathbf{Does}_{wt}\, \mathbf{Petr}\big[\mathbf{Imp}_w[\mathbf{Nv}\, x]_w\big]\Big] \wedge \\ \wedge \Big[\mathbf{SM}_{wt}\, x\Big]\Big]\Bigg] \tag{5.5a'}$$

$$(\mathsf{Petr}/\iota; \mathsf{Nv}/\mathsf{nakupovat\_v}/(o(o\pi)(o\pi))_{\omega}\iota; \mathsf{SM}/\mathsf{supermarket}/(o\iota)_{\tau\omega})$$

On the other hand, the TIL logical analysis of (5.5) in the reading (5.5b) has, up to our best knowledge, never been discussed in the literature. The problem of such sentence lies primarily in the vagueness of its verb tense determination — even if the grammatical tense is in the present form, the actual content of the proposition is talking about what the subject *is used to* do or *usually* does. So it discloses about the protagonist that he has done the thing before, may be even doing it right now and probably will do it again in the (near) future.

Hence, the analysis we suggest here leans on the similarity of the discussed Czech present tense reading with the English present perfect tense. The sentence (5.5) will thus be analysed as

Peter has been shopping in a supermarket. $(5.5\mathrm{b}_2)$

The analysis in TIL then looks like

$$\lambda w \lambda t \Big[\mathsf{\textbf{Pf}}_t \big[\mathsf{\textbf{Onc}}_w [\ldots (5.5\text{a'}) \ldots] \big] \ \mathsf{\textbf{Anytime}} \Big] \tag{5.5b$_2$'}$$

where the function $\mathsf{Pf}/(o(o(o\tau))(o\tau))\tau$ (the present perfect) and the function $\mathsf{Onc}/((o(o\tau))\pi)_\omega$ (the adverb ‘at least once’) will be defined in the next section on the past and future tenses and $\mathsf{Anytime}/o\tau$ is the whole time axe ($\mathsf{\textbf{Anytime}} \overset{df}{\Leftrightarrow} \lambda t[t = t]$).

5.1.3.2 Past and Future Tenses

The verb tenses that allow us to assert propositions with respect to the past or to the future can be looked at as mirror images of each other. Therefore, we will first concentrate just on the definition of the simple past tense[31] and after that we will analogously explicate the corresponding object for the future tense.

In the analysis, we understand the past tense as a certain operation working over

1. the *underlying proposition* in the (English) present tense form and
2. the *reference time span*
3. with regard to an *assertion moment*

As and example, let us take the sentence

Petr je rozlobený.[32] (5.5)

and its past tense variant

Petr byl rozlobený.[33] (5.6)

What is the difference between the two propositions? If we fix the reference world to be w, then the proposition (5.5) is in various intervals either true or false according to Peter's mood. A part of the proposition's chronology (the o-chronology generated by p_w where p is the proposition) may look like that one in the Figure 5.1a).

However, the proposition (5.6), with the reference time span being indefinite, embodies quite different time chronology — the proposition is true in

---

[31] denoted also as the *preteritum* in Czech

[32] ‘Peter is angry.’

[33] ‘Peter was angry.’

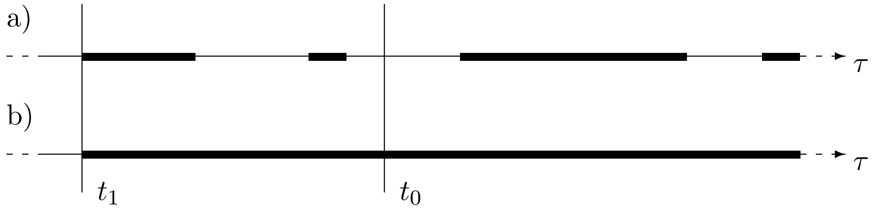


Figure 5.1: A chronology of a proposition in a) present tense and b) past tense (t_0 is an assertion moment, t_1 is the start of the first interval of the value of True in the present tense).

any moment since the first time when Peter was angry (t_1), e.g. at t_0 (5.5) happens to be false and (5.6) is true. Moreover, if we specify the reference time span of the past tense sentence as in

Petr byl rozlobený během celého 1. ledna 2001,[34] (5.6b)

we obtain a chronology where the truth-value depends, in addition, on the moment of assertion. If (5.6b) were given out before Jan 1st 2001, we cannot say whether it is true or false — its truth-value is undefined. The reason for this is that, except for the information contained in (5.5), the past tense sentence presupposes the fact that this information is related to a *past* interval, as well. When the assertion moment moves after the start of Jan 1st 2001, we can arbitrate the proposition's truthness according to the intersection of its chronology with the referred day and the past interval.

The meaning of a past tense proposition is often connected not only with a certain reference time span (in the indefinite case the object Anytime) but also with a *frequency adverb* specifying how many times the proposition happened to be true. A frequency adverb is analysed as a $((o(o\tau))\pi)_\omega$-object, i.e. as a world-dependent operation that takes a proposition p to the class of time intervals that have the requested qualities regarding the chronology of p. For instance, the adverb 'dvakrát'[35] takes every proposition p to a class of time intervals that have exactly two distinct intersections with the chronology of p (the explications of several frequency adverbs are presented in the Table 5.4). If the frequency adverb is not specified in the sentence, we assume that the frequency of the proposition is 'at least once' (object Onc) in its time span, which means that we do not limit the sentence's time span in that case.

---

[34] 'Peter was angry throughout Jan 1st 2001.'

[35] 'twice'

Adverb	**Definition**
během celého [*]	$\mathbf{Thr}_w\, p \overset{df}{\Leftrightarrow} \lambda s[s \subset p_w]$ $(\mathsf{Thr}/((o(o\tau))\pi)_\omega; p \dots \pi; s \dots o\tau)$
(alespoň) jednou v [*]	$\mathbf{Onc}_w\, p \overset{df}{\Leftrightarrow} \lambda s\big[[s \cap p_w] \neq \emptyset\big]$
právě jednou v [*]	$\mathbf{EOnc}_w\, p \overset{df}{\Leftrightarrow} \lambda s(\exists r)\big[r = [s \cap p_w] \wedge [\mathbf{NInt}^{**} r] = {}^0 1\big]$ $(\mathsf{NInt}/t(o\tau); r \dots o\tau)$
nejvýše jednou v [*]	$\mathbf{MOnc}_w\, p \overset{df}{\Leftrightarrow} \lambda s(\exists r)\big[r = [s \cap p_w] \wedge [\mathbf{NInt}\, r] \leq {}^0 1\big]$
ani jednou v [*]	$\mathbf{NoOnc}_w\, p \overset{df}{\Leftrightarrow} \lambda s\Big[\neg\big[[\mathbf{Onc}_w\, p]\, s\big]\Big]$
(alespoň) dvakrát v [*]	$\mathbf{Twc}_w\, p \overset{df}{\Leftrightarrow} \lambda s(\exists r)\big[r = [s \cap p_w] \wedge [\mathbf{NInt}\, r] \geq {}^0 2\big]$
poprvé v [*]	$\mathbf{Fst}_w\, p \overset{df}{\Leftrightarrow} \lambda s\big[[\mathbf{Start}^{***} p_w] \subset s\big]$

---

[*] ‘throughout’, ‘(at least) once in’, ‘exactly once in’, ‘at most once in’, ‘not even once in’, ‘(at least) twice in’, ‘for the first time in’

[**] NInt takes a class of time moments to the number of distinct continuous intervals in it.

[***] Start takes an interval to the singleton of its startpoint (see End in the Definition 20).

Table 5.4: Definitions of several frequency adverbs.

Thus, if P denotes the past tense function a general schema of the logical analysis of a past tense sentence looks like

$$\mathbf{P}\Big(\text{<frequency adverb>}\big(\text{<proposition>}\big),\ \text{<reference time span>}\Big)$$

The simple past is best seen as a time-dependent relation between $(o(o\tau))$-objects and $(o\tau)$-objects which holds for those cases where the (past part of the) reference time span belongs to the acceptable classes of time moments as obtained by the frequency modification of the proposition’s chronology.

Definition 22 (past tense). *Let* $\mathsf{P}/(o(o(o\tau))(o\tau))\tau$ *denote the* simple past *tense function.* P *is defined as*

$$\mathbf{P}_t\, a\, s \overset{df}{\Leftrightarrow} (\imath i)\Big[(\exists t_1)\big[s_{t_1} \wedge t_1 < t\big] \wedge i = \big[a\, \lambda t_2[s_{t_2} \wedge t_2 < t]\big]\Big]$$

$$(a \dots o(o\tau); s \dots o\tau; i \dots o)$$

□

In the definition, the singularizer $(\imath i)$ safeguards the important property of the P function — if s is the time span, s' is its intersection with the past before the reference time t and a the class of time moments obtained as the result of the frequency adverb, then the value of P is

- undefined if and only if $s' = \emptyset$
- else defined as $\begin{cases} \mathsf{True}, & \text{if } s' \in a \\ \mathsf{False}, & \text{otherwise} \end{cases}$

Now, we can construct the past tense proposition identified by the sentence (5.6b). Let $\mathsf{S}^{\mathsf{Jan\ 1^{st}\ 2001}}/o\tau$ denote the one day interval.

$$\lambda w \lambda t \Big[\mathbf{P}_t \big[\mathbf{Thr}_w \lambda w_1 \lambda t_1 [\mathbf{rozlobený}_{w_1 t_1} \mathbf{Petr}]\big]\ \mathbf{S}^{\mathrm{Jan\ 1^{st}\ 2001}}\Big] \tag{5.6b'}$$

Other verb tenses may be now defined analogously to the simple past tense. In addition to the future tense, we also define the English present perfect for the sake of the second reading of the Czech present tense (see the Section 5.1.3.1).

Definition 23 (future tense). *Let* $\mathsf{F}/(o(o(o\tau))(o\tau))\tau$ *denote the* future *tense function. We define* F *as*

$$\mathbf{F}_t\, a\, s \overset{df}{\Leftrightarrow} (\imath i)\Big[(\exists t_1)\big[s_{t_1} \wedge t < t_1\big] \wedge i = \big[a\, \lambda t_2[s_{t_2} \wedge t < t_2]\big]\Big]$$

$$(a \ldots o(o\tau); s \ldots o\tau; i \ldots o)$$

□

The difference between the simple past and the present perfect is in their demands to the reference time span — the present perfect links a class a of $(o\tau)$-objects with a reference time interval s only in cases, where s is an interval that starts in the past and goes (at least) up to the assertion moment t. If s does not agree with this requirement, then the present perfect tense function is undefined for it. Whenever s is of the right kind, a is linked with it according to whether s's past part is among the intervals in a or not.

Definition 24 (present perfect tense). *If* $\mathsf{Pf}/(o(o(o\tau))(o\tau))\tau$ *denotes the* present perfect *tense function, the behaviour of* Pf *is defined as*

$$\mathbf{Pf}_t\, a\, s \overset{df}{\Leftrightarrow} (\imath i)\Big[(\exists t_1)\big[\lambda t_2[t_1 < t_2 \leq t] \subset s\big] \wedge i = \big[a\, \lambda t_3[s_{t_3} \wedge t_3 \leq t]\big]\Big]$$

$$(a \ldots o(o\tau); s \ldots o\tau; i \ldots o)$$

□

The reference time interval is often expressed in a more complicated way than the exact specification 'v tu a tu dobu' or 'během intervalu'[36]. Let us take the sentence

$$\text{Petr byl rozlobený (alespoň jednou) před lednem 2001.}^{37} \qquad (5.6c)$$

In this sentence, the reference time is determined by the compound expression with the preposition 'před' (before). The analysis of such temporal prepositions is done by means of a $(o\tau(o\tau))$-object that relates a moment of time to a time span. Let $\mathsf{Bef}/(o\tau(o\tau))$ denote the relation corresponding to the expression 'před' and Aft the same for the preposition 'po' (after). The definitions of those two functions then look like

$$\begin{aligned} \mathbf{Bef}\, t\, s &\overset{df}{\Leftrightarrow} \neg\big[(\exists t_1)[s_{t_1} \wedge t_1 \leq t]\big] \\ \mathbf{Aft}\, t\, s &\overset{df}{\Leftrightarrow} \neg\big[(\exists t_1)[s_{t_1} \wedge t \leq t_1]\big] \end{aligned}$$

$$(t, t_1 \ldots \tau; s \ldots o\tau)$$

and the corresponding analysis of the sentence (5.6c) with the use of these functions is as follows

$$\lambda w \lambda t \Big[\mathbf{P}_t \big[\mathbf{Onc}_w \lambda w_1 \lambda t_1 [\mathbf{rozlobený}_{w_1 t_1} \mathbf{Petr}]\big] \quad \lambda t_2 [\mathbf{Bef}\, t_2\, \mathbf{S}^{\text{January 2001}}]\Big] \qquad (5.6c')$$

Another common way of expressing the reference time is represented by functions taking a certain time moment to a corresponding time interval or time moment (i.e. $((o\tau)\tau)$- and $(\tau\tau)$-objects) conveyed by adverbs like 'včera', 'letos' or 'nyní.'[38] The appropriate functions (Yd, Ty, Now) do not denote a particular time interval (day, year, ...) but rather a *chronology* of such intervals.

5.1.4 Active and Passive Voice

All the example sentences presented so far contained the active voice form of the respective verb. Let us recall the verb 'zničit'[39] in the sentence 'Petr

---

[36] 'at that time', 'during an interval'

[37] 'Peter was angry (at least once) before January 2001.'

[38] 'yesterday', 'this year' or 'now'

[39] 'to kill off'

zničil krtka'[40] with a possible analysis of

$$\lambda w\lambda t\left[\mathbf{P}_t\left[\mathbf{Onc}_w\lambda w_1\lambda t_1\left[\mathbf{Does}_{w_1t_1}\,\mathbf{Petr}\left[\mathbf{Perf}_{w_1}[\mathbf{zničit}\;\mathbf{Krtek}]_{w_1}\right]\right]\right]\mathbf{Anytime}\right]$$

But what if we want to express the mole's killing off from the point of view of the mole? Let us have a look at the sentence

$$\text{Krtek je zničen.}^{41} \tag{5.6}$$

With being killed off, we denote a property of the appropriate individual. Let us first suppose the function $\mathsf{K}/(o(o\pi))\iota$ that takes every individual x to the class of all the episodes which represent the killing off of x in world w. Then K may be constructed by $\lambda x[\mathbf{Perf}_w[\mathbf{zničit}\,x]_w]$ $(x\ldots\iota)$. Now, let $\mathsf{Pass}/((o\iota)((o(o\pi))\iota))\tau$ be the function that takes, at a given moment t, any such object like K to the class of individuals that are killed off in w at t.

$$\mathbf{Pass}_t\,f \overset{df}{\Leftrightarrow} \lambda x(\exists e)\left[[\mathbf{Ru}\,e\,t]\wedge f\,x\,e\right]$$

$$(t\ldots\tau; f\ldots(o(o\pi))\iota; x\ldots\iota; e\ldots o\pi)$$

Now, the sought property, that needs to be ascribed to the mole, of 'being killed off' is constructible by $\lambda w\lambda t\left[\mathbf{Pass}_t\lambda x[\mathbf{Perf}_w[\mathbf{zničit}\,x]_w]\right]$ and the sentence (5.6) can be analysed as

$$\lambda w\lambda t\left[\left[\mathbf{Pass}_t\lambda x\left[\mathbf{Perf}_w[\mathbf{zničit}\,x]_w\right]\right]\mathbf{Krtek}\right] \tag{5.6'}$$

5.1.5 Adverbial Modification

The meaning of a verbal object may undergo a modification by one of the four kinds of adverbial phrases (AP) in Czech:

1. a *locational* AP — answers questions 'kde', 'kam', 'odkud' or 'kudy'[42]
2. a *temporal* AP — questions 'kdy', 'odkdy', 'dokdy', 'jak dlouho' or 'jak často'[43]
3. a *modal* AP — questions 'jak', 'kolik', 'spolu s čím' or 's čí pomocí'[44]

---

[40] 'Peter killed off the mole.'

[41] 'The mole is killed off.'

[42] 'where', 'where to', 'where from' or 'which way'

[43] 'when', 'from when', 'until what time', 'how long' or 'how often'

[44] 'how', 'how many', 'together with what' or 'with the help of what'

4. a *causal* AP — questions ‘proč’, ‘za jakým účelem’, ‘za jaké podmínky’ or ‘i v kterém případě’[45]

An adverbial phrase may be represented by an adverb (a one-word expression), a clause, or a prepositional noun phrase (see the Section 5.2.2) or even a common noun phrase in the instrumental case.

Except for the cases where the adverbial phrase rather fits into the verb frame (e.g. the phrase representing an answer to the question ‘kolik’ — how many), the phrase influences the verb meaning in a certain way according to its functioning. Hence, generally, we can analyse adverbial phrases according to their classification:

- an analysis of a temporal AP has already been described in the Section 5.1.3.2 — such AP is usually formed by a combination of a frequency adverb and the reference time interval and is thus analysed as a part of the verb tense in the sentence analysis.
- in case of modal and locational APs, we can see that the resulting activity remains within the borders defined by the verb phrase without the modal or locational AP. The adverbial phrase just determines that not all of the episodes that can be regarded as the original activity serve the purpose well. Hence, the new verbal object is specified by a new *property of the verbal object.* As such, we analyse a modal or locational AP as a $(o(o(o\pi)(o\pi)))_\omega$-object — a function that in a world w selects those original verbal objects which agree with the corresponding AP expression of ‘how’ or ‘where’ the activity takes place.

EXAMPLE

Thus, if we take the sentence ‘Petr zuřivě zničil krtka’[46] (with the modal AP ‘zuřivě’ — furiously), we can analyse its central present tense part as

$$\ldots\lambda w\lambda t\Big[\textsf{\textbf{Does}}_{wt}\textsf{\textbf{Petr}}\big[\textsf{\textbf{Perf}}_{w}v\wedge v=[\textsf{\textbf{zničit Krtek}}]_{w}\wedge \\ \wedge\,\textsf{\textbf{zuřivě}}_{w}v]]\Big]\ldots$$

$$(\textsf{zuřivě}/(o(o(o\pi)(o\pi)))_\omega)$$

□

- the causal APs differ from the other adverbial phrases in that they do not modify the meaning of the verbal object directly, but rather

---

[45] ‘why’, ‘for what purpose’, ‘under what condition’ or ‘even in what case’

[46] ‘Peter furiously killed off the mole.’

capture the causality relationship between two propositions, the original one expressed by the sentence with the causal AP left out, and a proposition that matches with the causal AP. A problem arises with a very common case of elliptical form of the causal APs — by an adverbial phrase 'kvůli Petrovi,'[47] we mean a certain proposition that is currently described only by the fact that its leading actor is Peter. The causal AP thus represents a relationship between the proposition containing the AP and another underspecified proposition.

E
X
A
M
P
L
E

Analysis of a causal adverbial phrase may be demonstrated on the sentence 'Petr se odstěhoval kvůli Karlovi'[48] with the causal AP 'kvůli Karlovi' (for Charles'es sake). Again, for the sake of brevity, we present just the present tense part of the whole construction:

$$\ldots(\exists p)\Big[\textbf{kvůli}_{wt}\, p\, \lambda w \lambda t\big[\textbf{Does}_{wt}\textbf{Petr}[\textbf{Perf}_{w}\textbf{odstěhovat\_se}_{w}]\big] \wedge \\ \wedge \big[\textbf{About}\, p\, \textbf{Karel}\big]\Big] \ldots$$

$$(\mathsf{kvůli}/(o\pi\pi)_{\tau\omega})$$

where $\mathsf{About}/o\pi\iota$ links a proposition and an individual in case the individual (here Karel — Charles) takes the leading role in the proposition, i.e. the proposition must assign a certain property to Charles or say that something is done by Charles.

□

5.1.6 Auxiliary and Modal Verbs

The finite verb position in the sentence may be occupied not only by a self-reliant verb but also by a verb group consisting of an *auxiliary verb* ('mít', 'být'[49]) in combination with a noun phrase or a *modal verb*, such as 'muset', 'smět' or 'moci,'[50] together with an infinitive of a self-reliant verb.

The analysis of the first case, the auxiliary verb plus a noun phrase, has already been discussed in the beginning of the Section 5.1, because this is exactly the form of the attributive verbs.

---

[47] 'for Peter's sake'

[48] 'Peter moved away for Charles'es sake.'

[49] 'to have' (in the auxiliary meaning usually not translated) and 'to be'

[50] 'to have to' (must), 'to be allowed to' (may) or 'to be able to' (can).

EXAMPLE

Hence, an analysis of the sentence ‘Petr má chřipku.’[51] is conducted just in the following way.

$$\lambda w \lambda t[\textsf{\textbf{mít\_chřipku}}_{wt}\textsf{\textbf{Petr}}]$$

$$(\textsf{mít\_chřipku}/(o\iota)_{\tau\omega})$$

□

The case of a modal verb with an infinitive of a self-reliant verb might be treated as in the systems of *modal logic* — the modality of the activity, that is denoted by the infinitive after the modal verb, could be analysed as outside of the sentence. Thus the assertion

$$\text{Petr musí zničit krtka.}^{52} \tag{5.7}$$

might be analysed in the same way as

$$\text{Je nutné, že Petr zničí krtka.}^{53} \tag{5.8}$$

However, if we take ‘je nutné’[54] as the universal necessity constructible as $\textsf{\textbf{Nec}}\ p \overset{df}{\Leftrightarrow} (\forall w)(\forall t)[p_{wt}]$, then a sentence with ‘je nutné’ expresses a universal truth. As such, we can hardly ever use Nec for analysis of the verb ‘musí’ (must), since by (5.7) we certainly do not mean that it is a universal truth that Peter kills off the mole. It is even quite practicable that, despite (5.7), Peter will never do any harm to the mole. The only thing, we know from the assertion, is that the speaker ascribes a certain *attitude* to Peter towards his killing off the mole.

Therefore, we analyse modal verbs as relations between an individual (or a class of individuals) and a class of episodes,[55] i.e. $(o\iota(o(o\pi)))_{\tau\omega}$-objects (or $(o(o\iota)(o(o\pi)))_{\tau\omega}$-objects), and denote them by functions Must (Mustp), May (Mayp) and Can (Canp) of the respective types. The construction represented by the sentence (5.7) is then

$$\lambda w \lambda t\Big[\textsf{\textbf{Must}}_{wt}\textsf{\textbf{Petr}}\big[\textsf{\textbf{Perf}}_{w}[\textsf{\textbf{zničit Krtek}}]_{w}\big]\Big] \tag{5.7'}$$

---

[51] ‘Peter is ill with influenza.’

[52] ‘Peter must kill off the mole.’

[53] ‘It is necessary that Peter kills off the mole.’

[54] ‘it is necessary’

[55] cf. the Do and Does functions in the Section 5.1.1.2.

5.1.7 Infinitive

The ability of binding an infinitive form of another verb is not limited to the domain of modal verbs only. Several tens of self-reliant verbs, e.g. the phase verbs such as ‘začít’ or ‘přestat,’[56] include an infinitive in their verb frame. In such case, we analyse the infinitive as the world-instantiated verbal object (object of type $o(o\pi)(o\pi)$) of the corresponding verb.

EXAMPLE

Let us take the sentence ‘Petr začal nakupovat v supermarketu.’[57] In this sentence the finite form predicate is represented by the verb ‘začít’ (to start) and the infinitive belongs to the verb ‘nakupovat’ (to shop). The logical analysis of the sentence then looks like

$$\lambda w \lambda t \Bigl[\mathbf{P}_t \Bigl[\mathbf{Onc}_w \lambda w_1 \lambda t_1 \bigl[\mathbf{Does}_{w_1 t_1} \mathbf{Petr} \bigl[\mathbf{Perf}_{w_1} [\textbf{začít}\, v]_{w_1} \bigr] \wedge$$

$$\wedge\, (\exists x) \bigl[v = [\mathbf{Nv}\, x]_{w_1} \wedge [\mathbf{SM}_{w_1 t_1}\, x] \bigr] \bigr] \Bigr] \mathbf{Anytime} \Bigr]$$

$$(\mathsf{Petr}/\iota; \mathsf{začít}/(o(o\pi)(o\pi))_\omega(o(o\pi)(o\pi)); v \ldots o(o\pi)(o\pi);$$
$$\mathsf{Nv}/\mathsf{nakupovat\_v}/(o(o\pi)(o\pi))_\omega\iota; \mathsf{SM}/\mathsf{supermarket}/(o\iota)_{\tau\omega})$$

□

In sentences that contain the auxiliary verb ‘je’ (is) and certain adjective or noun phrase, like ‘je nutné’, ‘je žádoucí’ or ‘hlavní věc(í) je,’[58] the infinitive which follows such phrase may also stand for the sentence *subject*. However, in such kind of sentences, we do not take the infinitive as being attributed a certain property (which would be the case of an individual standing in that position). We rather respell the sentence from, e.g., ‘Je nutné začít něco dělat’ to ‘Je nutné, aby <u>někdo</u> začal něco dělat’[59] which keeps the meaning of the original sentence and does not contain an infinitive in the subject position. Instead, the resulting sentence then contains a subjective subordinate clause, whose analysis is described in the Section 5.3.1.

5.1.8 Verb Valency

As we have seen in the Definition 18, the TIL type of the object that is denoted by the verb in the finite form depends on the actual verb frame

---

[56] ‘to start’, ‘to stop’

[57] ‘Peter started to shop in a supermarket.’

[58] ‘is necessary’, ‘is desirable’ and ‘the main thing is’

[59] ‘(It) is necessary to start to do something’ and ‘It is necessary for <u>somebody</u> to start to do something.’

instantiated in the sentence. Each of the verb arguments may be assigned a different type from the *lexicon*. In the list of verb valencies for Czech,[60] we record the *syntactic surface structure* of the sentence constituents in contrast to their *semantic function*.[61] During the logical analysis in TIL, we need to identify yet another level of the denotation of a verb argument — its *meaning function*. On this level, we enter the construction of the TIL object represented by the corresponding NL expression.

The distinction of the three levels of verb frame representation may be demonstrated on the example of the verb 'dávat'[62] with a valency 'něco někomu.'[63] The three levels then can look like

1. syntactic surface structure:

   $\text{dávat něco}_{\text{non-human.NP, accus., no prep.}}\ \text{někomu}_{\text{human.NP, dat., no prep.}}$

   This level reflects those properties of constituents that can be derived following the morphological and syntactical analysis of the sentence.

2. semantic function:

   dávat Patiens Addressee

   The semantic function denotes the role of the verb arguments in the activity expressed by the verb — Patiens, the one that is acted on, Addressee, the one that is receiving something.

3. meaning function:

   $\mathsf{dávat}/(o(o\pi)(o\pi))_\omega\iota\iota \qquad x\ldots\iota \qquad y\ldots\iota : s_{wt}y, s\ldots(o\iota)_{\tau\omega}$

   On this level, we try to find the construction of the object that is represented by the corresponding constituent — $x\ldots\iota$, an individual, $y\ldots\iota : s_{wt}y, s\ldots(o\iota)_{\tau\omega}$, an individual from a class of individuals or an individual with a specified property.

The semantic function of the verb is thoroughly studied in the works of the Prague School researchers, where the roles of the verb arguments are described on the tectogrammatical level or the level of underlying representation (see [SgHaPa86, Sgall67]). The distinction of several levels of the natural language is a part of the Functional Generative Description (FGD) as the theoretical basis of the language formalism for automated systems.

---

[60] see [Horak98]

[61] in the conception of [SgHaPa86] the semantic function corresponds to the linguistic meaning

[62] 'to give'

[63] 'something to somebody'

In the conception of FGD, the level of surface structure corresponds to the the first stage of the NTA in our work. The output of the level of surface structure should be in successive steps translated into a tectogrammatical structure. The correspondence between the tectogrammatical level and the level of the logical analysis in the NTA is a matter of further research. The apparent advantage of the tectogrammatical level over the surface level offers to take up the results of that level and build the appropriate logical analysis based on this information. The main drawback of this approach lies in the complexity (and currently also underspecification) of the algorithms needed for an automatic analysis that would provide the output in the tectogrammatical representation. Nevertheless, the techniques described in our work are not confined to the surface level approach, they can readily exploit any higher-level representation which offers more input data to the system.

Another reason for "skipping" the tectogrammatical level in this version of the analysis consists in the fact that many of the phenomena commonly used in the natural language can be correctly analysed even within the range of the surface level analysis. Problems that arise when handling the irregularities like non-projectivity and elliptical constituents (which should be correctly resolved in the tectogrammatical structure) cannot be automatically analysed in the current phase of the implementation of the NTA. Solutions to the logical analysis of those less common phenomena are being sought together with the potential reuse of the awaited automatic tectogrammatical sentence parser.

In the analysis of the verb valency frame in the NTA, we need to find the appropriate translation from the syntactic structure to the meaning function. The particular construction and type that appears in the resulting sentence analysis depends on (at least):

1. the actual *input lexical items* the constituent consists of — their analysis has to be found in the lexicon.
2. the *context* — the lexicon often offers more than one possible analysis of the lexical item. However, on the upper level the surrounding lexical items may provide more details to the specification of the subject and so enable to select only the appropriate analyses of the item.

The basic guide-post for the list of valencies of Czech verbs that keeps the syntactic structure of the verb valency should route the translation of a valency expression (i.e. a specification of a verb argument) in the following way:

a *noun group* (with/without preposition).

The analysis of noun phrases is discussed in the Section 5.2.

an *adverbial phrase*

The constructions of adverbial phrases has been depicted in the Section 5.1.5.

a *subordinate clause*

The sentence building includes the description of analysis of relative and other subordinate clauses in the Section 5.3.1.

an *infinitive*

How to handle the infinitive form of a verb in the position of a verb argument has already been explained in the Section 5.1.7.

Following these guidelines, we construct the items in the lexicon for the corresponding verbs and verb frames.

5.2 Noun Phrase

A noun phrase is usually formed by a core, a noun, which is preceded by adjuncts in the form of an adjective, a pronoun or a numeral or a combination of such items. In the simplest case, the noun phrase consists of just one noun, whose analysis has to be found in the lexicon. Examples of common analyses of a noun are presented in the Table 5.5.

A special comment should belong to the analysis of $\iota_{\tau\omega}$-objects called *individual roles*. The use of this object brings many problems to the automatic analysis, since it is very difficult (if not beyond possibility) to find out that a noun phrase refers to an individual role and not just to a member of a class of individuals (i.e. to a *property* of individuals). Each example of the individual role analysis that is presented in the literature[64] may be given a counterexample in which the same expression is analysed in a different way ('americký prezident' — 'Bylo to nečekané setkání dvou amerických prezidentů,'[65] 'nejvyšší muž světa' — 'Na soutěž přijeli všichni nejvyšší muži světa.'[66])

---

[64] even in this work

[65] 'the American president' — 'It was an unexpected meeting of two American presidents.'

[66] 'the highest man in the world' — 'All the highest men in the world came to the competition.'

Noun	**Analysis**	**Description**
pes, člověk *	$x \ldots \iota$: $\textbf{pes}_{wt}x$, $\mathsf{pes}/(o\iota)_{\tau\omega}$	an individual from the class of individuals — such x for which $\textbf{pes}_{wt}x$ holds
prezident *	$\mathsf{prezident}/\iota_{\tau\omega}$	an individual role — see comment at the beginning of the Section 5.2
volitelnost *	$\mathsf{volitelnost}/$ $(o\iota_{\tau\omega})_{\tau\omega}$	a property of an individual role
výška, hmotnost *	$\mathsf{výška}/(\tau\iota)_{\tau\omega}$	a quantity
výrok, tvrzení *	$p \ldots *_\pi$: $\textbf{výrok}_{wt}p$, $\mathsf{výrok}/(o*_\pi)_{\tau\omega}$	a construction of a proposition from the class of constructions of a proposition
válka, smích, zvonění *	$\mathsf{válka}/(o(o\pi))_\omega$	a class of episodes — an activity that directly corresponds to a verb
leden, podzim *	$\mathsf{leden}/(o(o\tau))$	classes of time moments — time intervals specified by month or season.

---

*'dog', 'human'; 'president'; 'eligibility'; 'height', 'weight'; 'statement', 'assertion'; 'war', 'laughter', 'ringing'; 'January', 'autumn'

Table 5.5: Examples of a noun analysis.

In the light of these examples, we can never tell in the automatic processing (without the background knowledge of the speaker about that particular assertion) whether the noun phrase denotes an individual role (notwithstanding the *de dicto* and *de re* distinction of an individual role analysis[67]) or a property of individuals.

A solution to this problem that we offer lies in the specification of *individual role conditions* in the lexicon for certain lexical items (e.g. 'prezident/stát', 'markrabě/region', 'náš ...', 'ten ...', 'nej...' the superlative of an adjective[68]). Only under such condition(s), and if not ruled out by other circumstances like plural, the noun phrase can be analysed (also) as an individual role.

---

[67] see [Tichy88, section 41]

[68] 'president/state', 'margrave/region', 'our ...', 'the ...', 'the most ...'

5.2.1 Adjective Modifier

The first operation that a basic noun phrase can undergo so as to form a compound is the adjective modification. During this process, an adjective group is *applied* to the noun phrase and the resulting compound usually denotes a *subset* of the content of the original concept.

Hence, in case when the original concept represents a property (a class) of objects, the adjective modifier is of the same type and, together with original, defines the corresponding subset:

‘jablko’[69] .... $x \dots \iota$: $\textbf{jablko}_{wt}x$
‘červené jablko’[69]... $x \dots \iota$: $\textbf{jablko}_{wt}x \wedge \textbf{červený}_{wt}x$

(jablko,červený/$(o\iota)_{\tau\omega}$)

Such analysis of an adjective modifier is possible only in case the adjective modifier denotes a property of the extensional object (an individual in many cases). Otherwise, the adjective operates over the whole intension of the original noun phrase. We analyse the adjective as an function of type $((o\xi)(o\xi)_{\tau\omega})_{\tau\omega}$ that takes an $(o\xi)_{\tau\omega}$-object to a new object of the same type but with the modified content:

‘slon’[70]....... $x \dots \iota$: $\textbf{slon}_{wt}x$
‘malý slon’[70] .. $x \dots \iota$: $[\textbf{malý}_{wt}\textbf{slon}]\, x$

(malý/$((o\iota)(o\iota)_{\tau\omega})_{\tau\omega}$; slon/$(o\iota)_{\tau\omega}$)

In the special case when the resulting compound noun phrase fulfils an individual role condition, the whole compound noun phrase object is analysed with the help of the singularizer operator to point out the fact that the result must be either a particular individual (in a world and a time) or nothing if the role is currently unoccupied. The appropriate construction then looks like

‘prezident’[71]........... $x \dots \iota$: $\textbf{prezident}_{wt}x$
‘americký prezident’[71] ... $x \dots \iota_{\tau\omega}$: $x = \lambda w \lambda t (\imath y)[\textbf{prezident}_{wt}y \wedge \textbf{americký}_{wt}y]$

(americký/$(o\iota)_{\tau\omega}$; prezident/$(o\iota)_{\tau\omega}$)

Here, we can use the *type checking* mechanism for rejecting such analysis in case that some other modifier (upwards in the derivation tree) or another circumstance (e.g. plural) disagrees with it. In any of such cases, an object of

---

[69] ‘an apple’, ‘a red apple’
[70] ‘an elephant’, ‘a small elephant’
[71] ‘a president’, ‘the American president’

type different from the individual role would be expected on the place of the compound and the type $\iota_{\tau\omega}$ would be suppressed for such expression (with the possibility of being recalled later in the analysis — e.g. in expression 'ten americký prezident'[72]).

Some compounds have been assigned a special meaning (often within a particular terminology) that is quite different from the meaning of its constituents. In such noun phrases, the adjective modifier is not a subclass, but rather a function that constructs a new class according to the original one (an $(\xi\xi)_{\tau\omega}$-object):

'trpaslík'[73] . . . . $x \ldots \iota$: $\textbf{trpaslík}_{wt}x$
'bílý trpaslík (hvězda)'[73] . . . $x \ldots \iota$: $[\textbf{bílý}_{wt}\textbf{trpaslík}_{wt}]\, x$
(bílý$/((o\iota)(o\iota))_{\tau\omega}$; trpaslík$/(o\iota)_{\tau\omega}$)
'pes'[74] . . . . $x \ldots \iota$: $\textbf{pes}_{wt}x$
'dřevěný pes'[74]. . . $x \ldots \iota$: $[\textbf{dřevěný}_{wt}\textbf{pes}_{wt}]\, x$

If the original noun phrase (before the modification) does not denote a class of objects, the adjective modifier itself is analysed as a *property* of (the extension of) the corresponding object, i.e. it is of type $(o\xi)_{\tau\omega}$.

'výška'[75]. . . . . $\textbf{výška} \ldots (\tau\iota)_{\tau\omega}$
'malá výška'[75] . . . $x \ldots (\tau\iota)_{\tau\omega}$: $x = \textbf{výška} \wedge [\textbf{malý}_{wt}\, x_{wt}]$
(malý$/(o(\tau\iota))_{\tau\omega}$; výška$/(\tau\iota)_{\tau\omega}$)

An adjective group may be further modified with a modal adverb ('zdravě', 'zářivě'[76]) or quantificational adverb ('velmi', 'trochu'[77]). Here, the adverb again narrows down the class of the denoted objects. Therefore, we can analyse an adverbial modifier as a function that takes its argument of type $(o\xi)$, which is the type of (a member of the extension of) the adjective group object (adjective group always denotes a class of objects), to the subclass of those ξ-objects that comport to the quality (or degree) expressed by the adverb. The type of the object denoted by the adverbial modifier is thus $((o\xi)(o\xi))_{\tau\omega}$.

---

[72] 'that American president'
[73] 'a dwarf', 'a white dwarf (star)'
[74] 'a dog', 'a wooden dog'
[75] 'height', 'a small height'
[76] 'in a healthy manner', 'brightly'
[77] 'very', 'little'

‘zářivě červené jablko’[78]. . . $x \dots \iota$: $\mathbf{jablko}_{wt} x \wedge [\mathbf{zářivě}_{wt}\mathbf{červený}_{wt}]\, x$
‘velmi malá výška’[78] . . . . . $x \dots (\tau\iota)_{\tau\omega}$: $x = \mathbf{výška} \wedge$
$\wedge\, [\mathbf{velmi}_{wt}\mathbf{malý}_{wt}]\, x_{wt}$

(červený$/(o\iota)_{\tau\omega}$; zářivě$/((o\iota)(o\iota))_{\tau\omega}$; malý$/(o(\tau\iota))_{\tau\omega}$; velmi$/((o(\tau\iota))(o(\tau\iota)))_{\tau\omega}$)

5.2.2 Prepositional Noun Phrase

Prepositions or prepositional compounds (in the form of an idiomatic noun phrase such as ‘vzhledem k’[79]) take a noun phrase to form a new compound in order to

1. form a *verbal object* together with a verb (and possibly with other arguments of that verb). In such case, the preposition is explicitly stated in the verb valency list and can (however, often ambiguously) serve as the determinative constituent for the identification of the noun phrase at the place of the verb argument. The meaning of the noun phrase usually does not change, the preposition plays only a syntactic role here.

   EXAMPLE The verb ‘myslet’ can take e.g. the verb frame of ‘někdo myslí na někoho,’[80] with one possible instance of it in the sentence ‘Petr myslí na Marii.’[81] In that sentence the prepositional noun phrase ‘na Marii’ (of Mary) denotes a ι-object, the same as the noun ‘Marie’ (Mary) itself, and the function of the preposition ‘na’ is purely syntactic (it is often taken as a part of the verb itself). □

2. change its meaning to the meaning of an *adverbial phrase.* The analysis of such noun phrase then follows the rules described in the Section 5.1.5 about the adverbial modification of verbal objects.

In the point 1, we leave the analysed noun phrase as is and let it try to find its place within the particular verb frame.

What remains to be specified, is the transition from a noun phrase to the adverbial phrase mentioned in the point 2. Often, we cannot determine whether a prepositional noun phrase forms an adverbial phrase or a verb argument without evaluating other (sometimes quite complicated) conditions.

---

[78] ‘a brightly red apple’, ‘very small height’
[79] ‘according to’
[80] ‘to think’, ‘somebody thinks of somebody’
[81] ‘Peter thinks of Mary.’

That is why, we leave the process ambiguous for now, which means that either any of the two alternatives is rejected during the further processing, or the whole sentence analysis comes out ambiguous.[82]

The classification of the prepositional noun phrases into the classes according to the four kinds of adverbial phrases can be done (in most cases) in concordance with the following table.[83]

Preposition	**NP in**[84]	**AdvP**
‘od’$_{\text{away from}}$, ‘do’$_{\text{to}}$, ‘z’$_{\text{from}}$, ‘ze’, ‘u’$_{\text{at}}$, ‘vedle’$_{\text{next to}}$, ‘kolem’$_{\text{around}}$, ‘okolo’, ‘podél’$_{\text{along}}$, ‘poblíž’$_{\text{near to}}$, ‘nedaleko’$_{\text{not far from}}$, ‘uvnitř’$_{\text{inside}}$, ‘vevnitř’, ‘zevnitř’$_{\text{from inside}}$, ‘zpoza’$_{\text{from behind}}$, ‘zpod’$_{\text{from under}}$, ‘uprostřed’$_{\text{in the middle of}}$, ‘doprostřed’$_{\text{to the middle of}}$, ‘napříč’$_{\text{through}}$	genitive	loc.
‘od’$_{\text{since}}$, ‘do’$_{\text{until}}$, ‘uprostřed’$_{\text{in the middle of}}$, ‘doprostřed’$_{\text{to the middle of}}$, ‘u příležitosti’$_{\text{at the time of}}$, ‘během’$_{\text{during}}$, ‘koncem’$_{\text{at the end of}}$, ‘začátkem’$_{\text{at the beginning of}}$, ‘počátkem’	genitive	temp.
‘bez’$_{\text{without}}$, ‘beze’, ‘podle’$_{\text{according to}}$, ‘dle’, ‘včetně’$_{\text{including}}$, ‘kromě’$_{\text{besides}}$, ‘krom’, ‘vyjma’$_{\text{except}}$, ‘stran’$_{\text{about}}$, ‘pomocí’$_{\text{by the help of}}$, ‘prostřednictvím’$_{\text{by the medium of}}$, ‘(na)místo’$_{\text{instead of}}$,	genitive	mod.
‘následkem’$_{\text{in sequel to}}$, ‘v důsledku’, ‘v rámci’$_{\text{intra-}}$, ‘na rozdíl od’$_{\text{unlike}}$, ‘z hlediska’$_{\text{from the viewpoint of}}$	genitive	mod.
‘ohledně’$_{\text{regarding}}$, ‘navzdory’$_{\text{in spite of}}$	genitive	caus.
‘k’$_{\text{to}}$, ‘ke’, ‘ku’, ‘proti’$_{\text{against}}$, ‘naproti’, ‘oproti’	dative	loc.
‘k’$_{\text{to}}$, ‘ke’, ‘kvůli’$_{\text{because of}}$, ‘díky’$_{\text{thanks to}}$, ‘dík’, ‘vzhledem k’$_{\text{according to}}$	dative	caus.
‘přes’$_{\text{across}}$, ‘mimo’$_{\text{outside}}$, ‘skrz’$_{\text{through}}$, ‘naskrz’, ‘ob’$_{\text{every other}}$, ‘na’$_{\text{up to}}$, ‘před’$_{\text{to the front of}}$, ‘za’$_{\text{behind}}$, ‘nad’$_{\text{over}}$, ‘pod’$_{\text{under}}$, ‘mezi’$_{\text{between}}$	accus.	loc.
‘s odvoláním na’$_{\text{with reference to}}$, ‘v’$_{\text{in}}$, ‘ve’, ‘mimo’$_{\text{besides}}$	accus.	mod.
‘pro’$_{\text{for}}$, ‘přes’$_{\text{despite}}$, ‘s ohledem na’$_{\text{considering}}$, ‘bez ohledu na’$_{\text{disregarding}}$, ‘nehledě na’	accus.	caus.
‘při’$_{\text{at}}$, ‘v’$_{\text{in}}$, ‘ve’, ‘na’$_{\text{on}}$	locative	loc.
‘v’$_{\text{in}}$, ‘ve’	locative	mod.

---

[82] In which case, the sentence is often ambiguous even for a human reader.

[83] underlined prepositions can form more than one kind of APs and double underlined prepositions are ambiguous (also) in the grammatical case of the noun phrase they modify.

Preposition	**NP in**[84]	**AdvP**
‘při’$_{\text{during}}$, ‘po’$_{\text{after}}$	locative	temp.
‘při’$_{\text{with}}$, ‘v závislosti na’$_{\text{in relation with}}$	locative	caus.
‘před’$_{\text{in front of}}$, ‘za’$_{\text{behind}}$, ‘nad’$_{\text{over}}$, ‘pod’$_{\text{under}}$, ‘mezi’$_{\text{between}}$, ‘napříč’$_{\text{through}}$	instr.	loc.
‘před’$_{\text{before}}$, ‘mezi’$_{\text{between}}$	instr.	temp.
‘s’$_{\text{with}}$, ‘se’, ‘spolu s’$_{\text{together with}}$	instr.	mod.
‘v souvislosti s’$_{\text{in connection with}}$, ‘ve srovnání s’$_{\text{in comparison with}}$, ‘v porovnání s’	instr.	caus.

As a former of an adverbial phrase, the preposition is analysed as an (intensional) function that translates between (the extensions of) the corresponding types. For instance, the objects denoted by prepositions ‘před’ and ‘po’ (before, after) are constructible by $\lambda t[\mathbf{Bef}/\mathbf{Aft}\, t\, S] \ldots (o\tau)$, where Bef/Aft are $(o\tau(o\tau))$-objects[85] and S is the time period denoted by the noun phrase modified with the preposition. As we can see, the temporal adverbial phrase often denotes extensions since, in such case, they express a specific time interval independently of the selected world.

In the example of a locational adverbial phrase ‘v Praze’ (in Prague), the preposition ‘v’ denotes a function that takes an individual (Prague, the town) to the property of other objects (verbal objects or objects denoted by another noun phrase) being in Prague. Schematically, the function denoted by ‘v’ (in) is of the type $(\eta\xi)_{\tau\omega}$, where η is the type of the resulting property (e.g. $((o(o(o\pi)(o\pi)))\omega)$ in case of verbal objects) and ξ is the type of (a member or the extension of) the noun phrase modified by the preposition ‘v’ (e.g. ι for Prague, the town). In order to illuminate this step, let us have a look at the analysis in context:

‘Praha’[86] .... $\mathbf{Praha} \ldots \iota$
‘byt v Praze’[86] ..... $x \ldots \iota$: $\mathbf{byt}_{wt}x \wedge [\mathbf{v}^1{}_{wt}\mathbf{Praha}]_{wt}x$
‘bydlet v Praze’[86]... $v \ldots (o(o\pi)(o\pi))$: $v \subset \mathbf{bydlet}_w \wedge [\mathbf{v}^2{}_{wt}\mathbf{Praha}]_w v$

$$(\mathsf{v}^1/((o\iota)_{\tau\omega}\iota)_{\tau\omega}; \mathsf{v}^2/((o(o(o\pi)(o\pi)))_{\omega}\iota)_{\tau\omega})$$

A different approach needs to be followed for the causal adverbial phrases, where the preposition denotes a relation between the sentence proposition and another proposition that corresponds to the noun phrase modified by

---

[84] Each preposition modifies a noun phrase that must be in one specific grammatical case out of seven cases — nominative (not used in the table), genitive, dative, accusative, vocative (no preposition connects with this case), locative or instrumental.

[85] see the Section 5.1.3.2.

[86] ‘Prague’, ‘an apartment in Prague’, ‘to live in Prague’

the preposition. Such adverbial phrase is then analysed according to the procedure described in the part on causal adverbial phrase in the Section 5.1.5.

5.2.3 Genitive Construction

Two adjacent noun phrases can be joined together to form a new noun phrase by several reasons. The most common causes of such juncture are

1. an *apposition* of two non-prepositional noun phrases ascribe the corresponding properties to one and the same object and, therefore, we can analyse them as a conjunction of the two denotations. For instance, 'chudák matka'[87] may be analysed as $x \ldots \iota : \textbf{chudák}_{wt}x \wedge \textbf{matka}_{wt}x$.[88] Often, the apposition is used as a specification of an individual role — in such case one of the two noun phrases is usually a (compound of) proper name(s). Here, the resulting noun phrase can be analysed as expressing a proposition that is independent on the rest of the sentence: 'Alois Jirásek, rodák z Hronova, byl velký spisovatel'[89] has the same meaning as two propositions 'Alois Jirásek byl rodák z Hronova' and ' Alois Jirásek byl velký spisovatel.'[90]
2. the *prepositional attachment* — the second noun phrase specifies in more details some properties of the first noun phrase, the second NP starts with a preposition (e.g. 'muž s brašnou' or 'záloha na mzdu'[91])
3. the *genitive construction* expresses that the first noun phrase in some way *belongs* to the second noun phrase. In this case, the second noun phrase must be in genitive. E.g. 'dno řeky', 'stroj času' or 'výška postavy.'[92]

Hence, we know how to cope with point 1 and 2 (the analysis of the prepositional attachment has already been explained in the Section 5.2.2).

In the case of genitive construction, we can see that the meaning of the second noun phrase is either

a) an *argument* of the first noun phrase. Then the type checking mechanism must agree with it — e.g.

---

[87] 'poor mother' with 'poor' as a noun
[88] 'poor' (noun) and 'mother'
[89] 'Alois Jirásek, the native of Hronov, was a great writer.'
[90] 'Alois Jirásek was a native of Hronov', 'Alois Jirásek was a great writer.'
[91] 'a man with a bag', 'advance for payroll'
[92] 'a river bed', 'a time machine' and 'somebody's height'

‘výška’[93] .... **výška** ... $(\tau\iota)_{\tau\omega}$
‘postava’[93] .... $y \ldots \iota$: $\textbf{postava}_{wt}y$
‘výška postavy’[93] ... $x \ldots \tau$: $x = [\textbf{výška}_{wt}y] \land \textbf{postava}_{wt}y$

or

b) a *property* of the object denoted by the first noun phrase, though there is no expression that is connected with that property. For this purpose, we represent the property with relation $\mathsf{Of}/(o\eta\xi)_{\tau\omega}$, where η is the type of (a member of the extension of) the object denoted by the first noun phrase and ξ is the same for the second noun phrase. Thus, the compound noun phrase ‘dno řeky’ can be analysed as

‘dno řeky’[94] ... $x \ldots \iota$: $(\exists y)[\textbf{dno}_{wt}x \land \textbf{řeka}_{wt}y \land \textbf{Of}_{wt}x\,y]$

$(\mathsf{Of}/(o\iota\iota)_{\tau\omega})$

The capability of noun phrases to tie another noun phrase is (for noun phrases with arguments like ‘výška’) strong enough so that the decision, which of these two approaches should be chosen, can usually be safely superintended entirely by the type checking mechanism.

5.2.4 Pronoun and Proper Name

Inside a sentence, personal, possessive and demonstrative pronouns (and also some locational[95] adverbs like ‘tady’ or ‘nedaleko (odtud)’[96]) usually represent extra-sentential information expressed by the *indexical features* of the utterance. An exception to this is the case where pronouns[97] represent *anaphoric* (or cataphoric) relations between the sentence constituents like in

Jana náhle uviděla kousek před sebou velikého hada.[98]

---

[93] ‘height’, ‘person’, ‘height of a person’

[94] ‘a river bed’

[95] but not temporal adverbs even if their usage may look similar to that of the corresponding locational adverbs — ‘tady’ (here) depends on the utterance situation, but ‘teď’ (now) is the identity function over time moments. A more detailed substantiation of this fact can be found in [Tichy80a, pp. 367, note 14].

[96] ‘here’, ‘not far (from here)’

[97] all of the above mentioned kinds, i.e. personal, possessive and demonstrative, plus also the relative pronouns which refer to the head of the relative clause.

[98] ‘Jane suddenly saw a big snake right before her(self).’

The analysis of the anaphoric relations may be solved by connecting the pronoun with the sentence subject (in case of a reflexive pronoun such as 'sebe' or 'svůj'[99]) or with the (usually just preceding) verb object that must agree with the (personal or possessive) pronoun in number and gender:

Jana uviděla Petra a před ním velikého hada.[100]

The relation between the sentence subject and the reflexive pronoun may be either anaphoric or cataphoric (the pronoun may precede or follow the subject), while the linkage between a verb object and the corresponding personal or possessive pronoun must follow only the specified succession of constituents.

Apparently, even less amenable to the usual logical analysis is the meaning of sentences with indexicals which comprise expressions like 'já', 'ten muž vedle mě' and 'tady,'[101] but also (may be little surprisingly) proper names, especially in case of personal names.[102] Some proper names that refer to a really *unique* object (like 'hora Mount Everest' or 'Luna'[103]) can be safely analysed as ι-objects, but most of the common proper names (even New York refers to more than one town) are ambiguous and their meaning depends on the particular *communicative situation* the same as does the meaning of the indexicals.

An elegant solution to the analysis of a sentence with indexicals or proper names can be found in [Materna98, Section 7.1] in the conception of pragmatic meaning of *pragmatically anchored expressions*. According to Materna, a pragmatically anchored expression is every NL expression that contains indexicals, demonstratives or proper names. The (non-pragmatic) *meaning* of such expression is then the corresponding *open* construction with

---

[99] '(her/him/it)self', 'my/your/her/his/its/our/their'

[100] 'Jane saw Peter and a big snake before him.'

[101] 'I', 'the man next to me' and 'here'

[102] After an introspection into what the sentences with personal proper names like Peter or Václav Havel mean, we can see that the proposition (i.e. $o_{\tau\omega}$-object) denoted by such sentence may change from speaker to speaker (and even for one speaker from situation to situation) according to the fact which actual person the speaker intends to refer to. If we say

Václav dobře artikuluje (Václav articulates well).

then the relation between this sentence and its truth-value does not depend only on the chosen world and time, but also on the fact, whether the name Václav refers, e.g., to the Czech president Václav Havel or to Václav Klaus, currently the leader of the Czech Parliament. In such sense, the analysis of the sentence is analogous with the analysis of 'Ty dobře artikuluješ' (You articulate well).

[103] 'the Mount Everest', 'the Moon'

the problematic constituents analysed as free variables of the appropriate types. The *pragmatic meaning* of the expression in a particular *situation* is then the *closed* construction (or its concept) obtained by assigning (replacing) the particular objects to all the free variables.

In concordance with this straightforward (and thus, according to Occam's razor the most acceptable) conception, the analysis of the sentence 'Já vidím Janu'[104] will then look like

$$\lambda w \lambda t \Big[\textsf{\textbf{Does}}_{wt}\, J\acute{a} \big[\textsf{\textbf{Imp}}_{w} [\textsf{\textbf{vidět}}\ Jana]_{w}\big]\Big]$$ [105]

$$(J\acute{a}, Jana \ldots \iota)$$

As a matter of course, such analysis has to be *anchored*, i.e. closed according to the utterance situation, before it can be used in the inference mechanism since until then we cannot say anything about the truth-content of the sentence.

Note that for the anchoring of the expression the *names* of the variables are important (this fact makes them somewhat special among the variables) and that two meanings of pragmatically anchored expressions that differ only in the indexicals (i.e. in the names of the variables) cannot be counted as equivalent, since in one and the same situation they usually have different pragmatic meanings.[106]

5.2.4.1 Interrogative, Indefinite and Negative Pronoun

The indexical or anaphoric function is usually not connected with pronouns of one of the three remaining kinds — interrogative, indefinite or negative. The meaning of such pronoun is projected in a construction that influences the content of the analysis of the whole sentence.

An interrogative pronoun (like 'jaký', 'který' or 'kdo'[107]) always preface an interrogative sentence, a so called Wh-question. Their analysis is thus described in the Section 5.6 about the questions and imperatives. The same words can often represent a relative pronoun, which connects (in an anaphoric-like relation) a relative clause with its head noun phrase (see the Section 5.3.1).

---

[104] 'I can see Jane.'

[105] 'I', 'to see', 'Jane'

[106] A consequence of this fact is that in the analysis the indexicals cannot be represented as variables with names assigned in alphabetical order, as Materna states in [Materna98, pp. 119, Definition 45], but they must be given such names which definitely identify the appropriate indexical or proper name.

[107] 'what kind', 'which' and 'who'

Indefinite and negative pronouns are analysed by the help of the existential quantifier. With the indefinite pronoun, we express that there exists an entity ($\exists x$), which is the protagonist of the sentence or to which a certain property is ascribed. On the other hand, a negative pronoun in company with the negation of the whole clause (its finite form verb) expresses the exact opposite, i.e. 'it is not true that ($\exists x$) ...'. Therefore the negative pronoun is analysed as the denial ($\neg$) of the same sentence in positive and with a corresponding indefinite pronoun on the place of the negative pronoun ('Nikdo nepřišel' → 'Není pravda, že někdo přišel'[108])

5.2.5 Numeral

The numerals always refer (is some way) to a (definite or indefinite) amount or number (of some objects). We capture the particular form of the referring to a number with functions that relate a number with the corresponding object. The numerals are thus analysed according to the *method* they use for linking a number with an object. The most frequently appearing cases of such methods are

1. the *numbers* themselves. Examples of such numerals are 'jedna', 'sto' or 'dvě stě dvanáct'[109]. These numerals are in all means equivalent to their digital notation (1, 100, 212). Their interpretation depends heavily on the context, but in most cases we can analyse them as the corresponding τ-objects ($^0\mathsf{1}$, $^0\mathsf{100}$, $^0\mathsf{212}$), which in a noun phrase prefaced by the numeral (such as 'pět korun' or 'dvě školy'[110]) denotes the cardinality of a class of the objects that are denoted by the noun phrase. Note that this analysis should not be confused with the analysis of the expression 'počet (čeho)' (the number of some things), which is an intensional quantity (a $(\tau\xi)_{\tau\omega}$-object).

2. the *names of numbers* — 'jednička', 'trojka' or 'desítka.'[111] The behaviour of these numerals resembles in most cases individuals, since they mean (in such case) 'the object labelled with the number N'. Therefore, we can analyse them as $x \ldots \xi$: $[\mathbf{NumLabel\_of}_{wt}\, x\, {}^0\mathsf{N}]$ (the numerical label of x is N, $\mathsf{NumLabel}/(o\xi\tau)_{\tau\omega}$), where ξ is the type of the numbered object and $^0\mathsf{N}$ is the particular number. Otherwise, we

---

[108] 'Nobody [did not] came' → 'It is not true that somebody came.' — Czech uses double negation 'nobody' and 'did not' in one sentence.

[109] 'one', 'one hundred', 'two hundred and twelve'

[110] 'five crowns', 'two schools'

[111] '(the name of) one', '(the name of) three', '(the name of) ten'

can analyse the number names in the same way as numbers, i.e. as the corresponding τ-objects. Such analysis is needed for sentences of the form ‘Dvojka a trojka dají dohromady pětku.’[112]

3. *ordinal* numerals specify the property of an object to be the N^{th} in some sequence of objects (e.g. ‘první’, ‘druhý’, ‘stý’, ‘padesátý druhý’[113]). In compliance with their syntactico-morphological function, we can analyse them by means of a linkage that relates the number with (a member of the extension of) the ordered object. Let $\mathsf{Ord}/(o\tau\xi)_{\tau\omega}$ denote the linkage. The analysis of an ordinal numeral then can look like

   ‘první’[114] . . . . $x \ldots (o\iota)_{\tau\omega}$: $\lambda w\lambda t\lambda x[\mathbf{Ord}_{wt}{}^{0}1\, x]$
   ‘první plavec’[114] . . . $x \ldots \iota$: $\mathbf{plavec}_{wt}x \wedge [\mathbf{Ord}_{wt}{}^{0}1\, x]$

   There are cases where a noun phrase with a numeral denotes an individual role (e.g. ‘první muž na Měsíci’[115]). However, the conditions under which a noun phrase denotes an individual role are too broad to be specified as the individual role conditions. Thus, the only generally applicable rule seems to be to (ambiguously) offer the analysis as an individual role to the type checking mechanism and let it choose (if determinable) the right eventuality in the broader context.

4. numerals *of kind* that express the multiple character of an object. Examples are ‘dvojí’, ‘několikerý’, ‘patero’, ‘desatero.’[116] The words with the ending ‘-ero’ are in most cases used as an (archaic) version of common numbers and can thus be analysed as numerals in point 1. Otherwise, the numerals of kind mean that a certain object has a property that it comes in N different kinds. We can analyse this property with a relation $\mathsf{NKinds}/(o\tau\xi)_{\tau\omega}$:

   ‘život’[117] . . . . $\mathbf{život} \ldots (o(o\pi))_{\omega}$
   ‘dvojí život’[117] . . . $\lambda w\lambda z[z = \mathbf{život}_{w} \wedge \mathbf{NKinds}_{wt}{}^{0}2\, z] \ldots (o(o\pi))_{\omega}$

---

[112] ‘The (name of) two and the (name of) three make together the (name of) five.’
[113] ‘first’, ‘second’, ‘hundredth’, ‘fifty second’
[114] ‘first’, ‘first swimmer’
[115] ‘the first man on the Moon’
[116] ‘double’, ‘(of) several (kinds)’, ‘(of) five’, ‘(of) ten’
[117] ‘life’, ‘alternative lives’ (in singular in Czech)

5. *frequency adjectives* say how many times is something multiplied (e.g. 'dvojnásobný', 'desetinásobný'[118]). We analyse such property with the relation $\mathsf{Mult}/(o\tau\xi)_{\tau\omega}$, so that $[\mathbf{Mult}_{wt}{}^0\mathsf{N}\,x]$ means 'x is N-fold'.

6. expressions 'několikrát', 'pětkrát' or 'dvacetkrát'[119], which express 'how many times/how often is something done', are sometimes counted among numerals. In concordance with their analysis, we take them as *frequency adverbs* that are discussed in the Section 5.1.3.2 about the verb tenses.

7. *fractions* like 'polovina', 'pětina', 'dvě setiny'[120] are just special kinds of numerals in point 1. The main difference between them lies in the fact that fractions represent not only a number, but also a certain procedure (that corresponds to the mathematical expression). Hence, we analyse them as τ-constructions that construct the corresponding number with the help of the operation '/' of the type $(\tau\tau\tau)$. The analysis of a fractional expression then looks like $[{}^0 1/{}^0 2]$, $[{}^0 1/{}^0 5]$ or $[{}^0 2/{}^0 100]$.

8. numerals that denote N-*tuples* specify a class of objects with a definite number of elements. Examples of such numerals can be 'dvojice', 'trojice' or 'pětice.'[121] We thus analyse them as properties of classes of ξ-objects $((o(o\xi))_{\tau\omega})$. The corresponding class has the property, if it fulfils the extra condition that the cardinality of the class is a certain number. Such an analysis can look like

   'pětice'[122] $\ldots\ldots z \ldots (o\iota)$: $[\mathbf{Card}\, z] = {}^0 5$
   'pětice psů'[122] $\ldots z \ldots (o\iota)$: $\big[z \subset \mathbf{pes}_{wt} \wedge [\mathbf{Card}\, z] = {}^0 5\big]$

Almost all of the above different kinds of numerals may be specified with an *indefinite* numeral, such as 'několik', 'několikrát' or 'mnohokrát.'[123] Such numerals are, as most of the indefinite (or vague) expressions, the possible source of many problems in the logical analysis and especially in the successive inference. A discussion on the subject of vagueness can be found e.g. in [Materna98, Section 7.2.3], however, even there the author does not offer an intuitive and generally acceptable solution to the analysis of vague

---

[118] 'twofold', 'tenfold'

[119] 'several times', 'five times' and 'twenty times'

[120] 'a half', 'a fifth', 'two hundredths'

[121] 'doublet', 'triplet', 'quintuplet (five-tuple)'

[122] 'quintuplet' (five-tuple), 'five-tuple of dogs'

[123] 'several', 'several times', 'many times'

expressions. If we try to go deeper into the exact specification of *vagueness*, we soon come across the fact that almost all NL expressions are vague and indefinite in so far that even a property such as being a table can have very foggy borders where we cannot with certainty say whether a particular object of some fantastic shape still *is* a table or already cannot be counted as such.

A potential solution, which, however, we do not embrace in the NTA (at least at the moment), could lie in extending the content of the type o of truth-values into the real interval $<0;1>$ and replace the equality operator $=_o$ (and other function working with truth-values, of course) with several *fuzzy* operators in order to express the relative *probability* of being True or False.

Nevertheless, in TIL without such far-reaching changes, we can employ one of two approximative approaches: 1) we can define the appropriate τ-properties ($(o\tau)_{\tau\omega}$-objects) which undertake the specification of how many is 'several' or 'many'; or 2) the probably most exact, but unfortunately not self-reliant for the inference, is the analysis of the indefinite numerals as pragmatically anchored expressions (see the Section 5.2.4), i.e. as $(o\tau)$-variables that specify the particular acceptable range of a numeral according to the communicative situation.

5.2.5.1 Mathematical Expression

Some compound noun phrases may be regarded as special kinds of numerals — the mathematical expressions. By this term, we understand such noun phrases that denote a numerical non-intensional object and often the phrase and its constituents directly correspond to the appropriate construction and subconstructions of the numerical object.

As a mathematical expression, we can count also the number itself, which is often (especially in case of big non-unitary numbers like 'dvanáct tisíc čtyři sta šedesát devět'[124]) a compound of several words. However, their analysis should be already the part of the lexical (pre-syntactical) analysis, since it would be quite untenable to represent the number ^{0}12469 as a construction that corresponds to the functions of the appropriate parts of its verbal representation.

Apart from numbers, the mathematical expressions consists of operators like 'plus', 'mínus', 'krát', 'sinus', 'na druhou'[125] and many others. All of these operators can be analysed as the corresponding (extensional) mathematical

---

[124] 'twelve thousands four hundred sixty nine'

[125] 'plus', 'minus', 'times', 'sinus', 'squared'

functions (in many cases $(\tau\tau\tau)$-objects) with the resulting construction of the whole expression being analysed as a $*_\tau$-object which can be related to an individual or other objects in the verbal objects of *attitude* to a (mathematical) expression such as 'násobit (výraz jiným výrazem)' or 'počítat (výraz).'[126]

5.2.6 Quantificational Phrase

If a noun phrase denotes a class of objects in place where the verb expects only one of those objects, the noun phrase usually represents a *quantificational expression* whose meaning is governing the meaning of the whole sentence. Let us have a look at the sentence

$$\text{Petr sleduje na obloze deset labutí.}^{127} \tag{5.9}$$

According to [Tichy94b], the expression 'deset labutí' (ten swans) is an (intensional) quantifier of the whole sentence and its analysis may be schematically symbolized as

$$\text{Deset labutí}\,\lambda x[\,\text{Petr sleduje na obloze } x\,]^{128} \tag{5.9a}$$

which then corresponds to a construction following the appropriate schema of $\lambda w\lambda t\big[\textsf{\textbf{deset\_labutí}}_{wt}\lambda x[\ldots x\ldots]\big]$, where $\textsf{deset\_labutí}$ is an $(o(o\iota))_{\tau\omega}$-object and means 'those classes of individuals watched by Peter overhead which are swans and have exactly ten members.'

However, alike to the need of plural versions of the relation Does between a subject and a verbal object, in some cases, the action is necessary related to the whole group of objects represented with a quantificational expression and if we itemize the action to each of the group members, the meaning of the whole sentence changes. As an example, let as take the fairy-tale sentence

$$\text{Honza zabil přesně sedm much jednou ranou.}^{129} \tag{5.10}$$

We are able to find two possible logical analysis of the quantifier 'přesně sedm much' (exactly seven flies), where one follows the Tichý's directions and the other takes the quantificational phrase as one argument of the verb — the group of objects as a whole:

$$\text{Přesně sedm much}\,\lambda x[\,\text{Honza zabil } x \text{ jednou ranou}\,]^{130} \tag{5.10a}$$

$$[\,\text{Honza zabil } s \text{ jednou ranou}\,] \wedge s = [\text{přesně sedm much}]^{131} \tag{5.10b}$$

---

[126] 'multiply (an expression with another expression)' or 'count (an expression)'

[127] 'Peter watches ten swans overhead.'

[128] Ten swansλx[Peter watches x overhead]

[129] 'Jack has killed exactly seven flies at a single blow.'

We can see that in the reading (5.10a) the resulting proposition can in no means express that all the seven flies were killed *at a single blow* as is stated in the reading (5.10b) and as naturally corresponds to the meaning of the sentence (5.10).

Moreover, when we apply the analysis schema of (5.10b) to a sentence like (5.9), we obtain a correct analysis of the sentence:

$$[\,\text{Petr sleduje na obloze } s\,] \wedge s = [\text{deset labutí}]^{132} \tag{5.9b}$$

This analysis seems to follow the meaning of (5.9) in a more exact way than (5.9a), since if Peter watches ten swans, then saying that it is exactly the same as when he watches every one of them is, at least, unlikely (imagine that Peter watches one thousand of swans — in that case it would be impossible for him to watch every single swan among them).

We thus prefer the *plural* analysis (that one in (5.9b) and (5.10b)) of quantificational phrases, even if it means that we need to provide also a 'plural version' of the verb arguments in a verb type specification. The encouragement of such analysis lies first in its agreement with the intuitive understanding of the meaning of a sentence with a quantificational phrase, and second in the analogy of the analysis with the subject-predicate Does and Do relations designed by Tichý.

5.3 Sentence Building

The logical analysis described in the previous sections involved only those sentence constituents that do not exceed the boundaries of a single clause. Now, we are going to show how we can combine the subconstructions corresponding to particular clauses in order to capture the appropriate subordinate and coordinate relations between the clauses in the resulting construction of the whole sentence.

5.3.1 Subordinate Clauses

A clause that is in a subordinate position to a principal clause may represent almost any of the principal clause constituents. So the kinds of subordinate clauses correspond one to one to the kinds of the clause constituents.

---

[130] Exactly seven fliesλx[Jack has killed x at a single blow]

[131] [Jack has killed s at a single blow] $\wedge\, s =$ [exactly seven flies]

[132] [Peter watches s overhead] $\wedge\, s =$ [ten swans]

An important feature of a subordinate clause is the way (usually the conjunction) how this clause connects to the principal clause. We use the clause conjunction as a leading element for determining the kind of the clause (and thus the form of its analysis).

In the following table, we give a classification of subordinate conjunctions according to the possible kind of object denoted by the clause they preface. Some conjunctions can consists of more than one word — if such phrase contains a comma (‘,’), then we mark its position in the phrase with a dash (‘-’, an exception to this is the verb ending ‘-li’, which is attached right at the end of the word without any intercepting comma). Underlined conjunctions are those that can preface more than one kind of clauses.

Subordinate Conjunction[133]	**Clause**
‘kdo’$_{\text{who}}$, ‘co’$_{\text{what}}$, ‘jaký’$_{\text{what kind}}$, ‘který’$_{\text{which}}$, ‘jenž’$_{\text{who}}$, ‘jako’$_{\text{as}}$	adj.
‘že’$_{\text{that}}$, ‘aby’$_{\text{in order to}}$, ‘ať’$_{\text{I wish}}$, ‘jak’$_{\text{how}}$, ‘kolik’$_{\text{how many}}$	cons.
‘kde’$_{\text{where}}$, ‘kam’$_{\text{where to}}$, ‘odkud’$_{\text{where from}}$, ‘kudy’$_{\text{which way}}$, ‘kamkoli’$_{\text{wherever}}$, ‘odtud’$_{\text{from here}}$	loc.
‘když’$_{\text{when}}$, ‘až’$_{\text{till}}$, ‘jak’$_{\text{as soon as}}$, ‘hned jak(mile)’, ‘jakmile’, ‘sotva(že)’, ‘ještě než’$_{\text{before}}$, ‘než’$_{\text{until}}$, ‘zatím(co|-co)’$_{\text{while}}$, ‘dokud’$_{\text{until}}$, ‘kdykoli’$_{\text{whenever}}$	temp.
‘jakožto’$_{\text{as}}$, ‘tak-že’$_{\text{so-that}}$, ‘tak-aby’$_{\text{so-to}}$, ‘tak-jak’$_{\text{so-as}}$, ‘bez toho-že’$_{\text{without the fact that}}$, ‘bez toho-aby’$_{\text{without it to}}$, ‘div-že’$_{\text{a wonder that}}$, ‘jinak než’$_{\text{other than}}$, ‘jako by’$_{\text{as would}}$	mod.
‘protože’$_{\text{because}}$, ‘proto’$_{\text{therefore}}$, ‘tudíž’, ‘pročež’, ‘poněvadž’$_{\text{since}}$, ‘jelikož’, ‘tedy’$_{\text{thus}}$, ‘takže’$_{\text{insomuch that}}$, ‘díky tomu-že’$_{\text{thanks to that}}$	caus.
‘aby’$_{\text{in order to}}$, ‘proto-že’$_{\text{for -ing}}$, ‘proto-aby’$_{\text{so as to}}$, ‘za účelem toho-aby’$_{\text{for the purpose of}}$, ‘k tomu-aby’$_{\text{(make sb.) to do}}$, ‘kvůli tomu-že’$_{\text{because of}}$, ‘se zřetelem na to-že’$_{\text{in consideration of}}$	purp.
‘jestliže’$_{\text{if}}$, ‘jestli’, ‘zda’, ‘-li’, ‘když’$_{\text{when}}$, ‘pokud’$_{\text{as far as}}$, ‘leda(že)’$_{\text{unless}}$, ‘kdyby’$_{\text{if it were}}$, ‘pokud by’$_{\text{as far as it were}}$, ‘za podmínky-že’$_{\text{under condition that}}$	cond.
‘ač(koli(v))’$_{\text{although}}$, ‘třeba(s)(že)’$_{\text{though}}$, ‘přestože’$_{\text{even if}}$, ‘i když’, ‘ani když’$_{\text{not even if}}$, ‘aniž’, ‘i kdyby’$_{\text{even if it were}}$, ‘až na to-že’$_{\text{except for}}$, ‘(jen) taktak-že’$_{\text{almost (not)}}$	conc.
‘pokud jde o’$_{\text{regarding}}$, ‘co se týká(týče)’$_{\text{as for}}$	synt.

---

[133] Except conjunctions, we list here some other words, that may connect a subordinate clause to its principal clause, namely relative pronouns such as ‘jaký’, ‘který’ or ‘jenž’

The analysis of the corresponding clauses or sentences comprising such clauses can then proceed according to the following guidelines:

1. *adjectival (adj.)* — a clause expressing a property in a way similar to an adjective group. Hence, their analysis also follows the rules that were described in the Section 5.2.1 (or in a broader sense in the whole Section 5.2 which corresponds to the cases where a clause represents a subject or a verb argument and not just an attribute of another object). The main difference between a common adjective group and what we call an adjectival clause, is that the subject of the adjectival clause (usually expressed with the conjunction in the form of a relative pronoun) is denoted by a variable and serves as a connection to the principal clause. As an example let us have a look at the sentence ‘Kdo si hraje, nezlobí.’[134] Its analysis may look like

   $$\lambda w\lambda t\neg(\exists x)\big[\textbf{Does}_{wt}x[\textbf{Imp}_w\textbf{hrát\_si}_w] \wedge \textbf{Does}_{wt}x[\textbf{Imp}_w\textbf{zlobit}_w]\big]\;\text{[135]}$$

   The clause ‘Kdo si hraje’ (who is playing) is first analysed as a property constructible by $\lambda w\lambda t\lambda x[\textbf{Does}_{wt}x[\textbf{Imp}_w\textbf{hrát\_si}_w]]$ which is then applied to the variables w, t and x in the principal clause and by reduction converted to the form above.

2. *constructional (cons.)* clauses are usually analysed as trivializations of the corresponding constructions. In the principal clause, a constructional clause represents an object of a higher-order type (the most frequent are $*_\pi$- and $*_\tau$-objects).

   EXAMPLE

   An example can be the sentence ‘Můžu si spočítat, že dva plus dva jsou čtyři.’[136]

   $$\lambda w\lambda t\Big[\textbf{Can}_{wt}J\acute{a}\Big[\textbf{Perf}_w\big[\textbf{spočítat\_si}\;{}^0[{}^02+{}^02={}^04]\big]_w\Big]\Big]\;\text{[137]}$$

   The constructional clause ‘že dva plus dva jsou čtyři’ (that two plus two makes four) is analysed as a trivialization of the corresponding mathematical expression. □

(‘what kind’, ‘which’, ‘who’) or adverbs like ‘kde’, ‘odkud’ or ‘kdy’ (‘where’, ‘where from’, ‘when’).

[134] ‘Who is playing, is not fractious’

[135] ‘to play’, ‘to be fractious’

[136] ‘I can count that two plus two makes four.’

[137] ‘I’, ‘to count’

3.–5. *locational (loc).*, *temporal (temp.)* and *modal (mod.)* clauses represent adverbial phrases of the corresponding kind. In the analysis of such clauses, the object that we ascribe to the conjunction is a function which takes a proposition[138] to the corresponding object needed for the analysis of the deputized adverbial phrase.

EXAMPLE

For instance, the analysis of the sentence with a locational clause 'Petr se vrací, odkud Karel přišel'[139] can in a shortened form look like

$$\lambda w\lambda t\Big[\mathbf{Does}_{wt}\mathbf{Petr}\Big[\mathbf{Imp}_{w}v \wedge v = \mathbf{vracet\_se}_{w} \wedge \wedge \big[\mathbf{odkud}_{wt}\lambda w_1\lambda t_1[\ldots\mathbf{Karel}\ldots\textbf{přijít}\ldots]\big]_{w}v\Big]\Big]$$ [140]

where odkud (from where) is of type $((o(o(o\pi)(o\pi)))_{\omega}\pi)_{\tau\omega}$. □

A slightly different approach is chosen in the analysis of an adverbial temporal clause where we do not ascribe the generating of a time class to the conjunction, but rather use the clause's construction directly as a generator of a collection of the time moments where the clause's extension is True.

EXAMPLE

Hence, the analysis of the sentence 'Karel přišel, když byl Petr nemocný'[141] can look like

$$\lambda w\lambda t\Big[\mathbf{P}_{t}\Big[\mathbf{Onc}_{w}\lambda w_1\lambda t_1\Big[\mathbf{Does}_{w_1t_1}\mathbf{Karel}\big[\mathbf{Perf}_{w_1}\textbf{přijít}_{w_1}\big]\Big]\Big] \lambda t_0\Big[\mathbf{P}_{t}\Big[\mathbf{Onc}_{w}\lambda w_2\lambda t_2\big[\textbf{nemocný}_{w_2t_2}\mathbf{Petr}\big]\Big]t_0\Big]\Big]$$ [142]

---

[138] A question is, whether the functions and relations corresponding to an attachment of a subordinate clause should not work over $*_{\pi}$-objects represented by the trivializations of the constructions that correspond to the respective clauses. The main difference between these two approaches lies in that if the function works directly with π-objects, it cannot distinguish two equivalent constructions (i.e. constructions that construct one and the same proposition). However, from examples of the adverbial phrases (including clauses), we can suppose that the denotation of an adverbial phrase, i.e. the specification of the place, cause, condition, purpose, etc., does not depend on the exact construction and is equally represented by the corresponding proposition.

[139] 'Peter goes back, from where Charles came.'

[140] 'Peter', 'to go back', 'from where', 'Charles', 'to come'

[141] 'Charles came, when Peter was ill.'

The temporal clause ‘když byl Petr nemocný’ (when Peter was ill) is analysed as $\lambda w\lambda t\lambda t_0\left[\mathbf{P}_t\left[\mathbf{Onc}_w\lambda w_2\lambda t_2\left[\textbf{nemocný}_{w_2t_2}\mathbf{Petr}\right]\right]t_0\right]$ which is then used in the principal clause as a generator of the reference time span, i.e. a characteristic function of a class of time moments ($\lambda t_0[\ldots]$). □

6.–9. all the other kinds of clauses (except the last one) are a sort of *causal adverbial* clauses. In the logical analysis they represent a certain relation between the propositions (see also the Note 138) denoted by the principal and causal clauses. The *causal (caus.)* (in the narrower sense) clauses express a reason for or a substantiation of the assertion in the principal clause. A *final (purp.)* clause says that the principal clause is done for the purpose specified in the final clause. Analogously to the causal clauses the *conditional (cond.)* clauses determine the condition (real or unreal) under which the principal clause’s assertion holds. And, at last, the *concessive (conc.)* clauses state a circumstance which contravenes to the principal clause.

In analogy with the analysis of the causal adverbial phrases, we let the clause’s conjunction to denote the appropriate relation between the two propositions.

EXAMPLE

As an example, let us demonstrate the analysis of a (causal) conditional clause in the sentence ‘Petr na výlet nepůjde, ledaže se ještě dnes uzdraví.’[143]

$$\lambda w\lambda t\big[\textbf{ledaže}_{wt}\lambda w_1\lambda t_1[\ldots\mathbf{Petr}\ldots\textbf{jít}\ldots] \\ \lambda w_2\lambda t_2[\ldots\mathbf{Petr}\ldots\textbf{uzdravit\_se}\ldots]\big]$$ [144]

The conditional conjunction ‘ledaže’ (unless) is here analysed by means of the object ledaže$/(o\pi\pi)_{\tau\omega}$. □

The conjunction of some causal clauses could be also translated into their logical equivalents (with the help of $\Rightarrow$, $\neg$, $\wedge$ or $\vee$). Since this process may bring some inaccuracies into the analysis, we prefer in the NTA, at least at the moment, to keep the “causal” functions in one to one relation to the actual NL expressions that denote them. However, nothing hinders us from describing the implications between propositions in all respects by means of rules of the inference mechanism.

---

[142] ‘Charles’, ‘to come’, ‘ill’, ‘Peter’

[143] ‘Peter will not go for a trip unless he gets over today.’

[144] ‘unless’, ‘Peter’, ‘to go’, ‘Peter’, ‘to get over’

10. several conjunctions seem to have only a *syntactical (synt.)* function in a sentence and as thus they do not influence the logical analysis of the sentence. E.g. the sentence 'Co se týče vašeho syna, tak je to chytrý hoch' can be taken as logically equivalent to 'Váš syn je chytrý hoch.'[145]

These 10 points should cover most of the cases of the logical analysis of subordinate relations between clauses in a sentence.

5.3.2 Coordinate Clauses

The coordinate relationship between equipollent clauses is in its meaning very similar to that of the causal subordinate clauses. The coordinate conjunctions can be as well analysed as linkages between two or more propositions which correspond to the particular (principal) clauses.

In the following table, we divide the coordinate conjunctions into several classes according to the (coarse-grained) method they use for connecting the clauses. The words in the table represent either conjunctions that are written after or without a comma between two clauses, or conjunctions that appear in both the connected clauses — in such case, we signify a long dash ('—') between them.

Coordinate Conjunction	**Relation**
'a'$_{\text{and}}$, 'také'$_{\text{also}}$, 'i', 'pak'$_{\text{then}}$, 'potom', 'ani—ani'$_{\text{neither—nor}}$, 'jednak—(a) jednak'$_{\text{on the one hand—on the other hand}}$, 'dílem—dílem'$_{\text{partly—partly}}$, 'přičemž', 'ani'$_{\text{not even}}$, 'ba'$_{\text{nay}}$, 'ba i'$_{\text{even}}$, 'ba dokonce', 'dokonce', 'nejen—ale (i)'$_{\text{not only—but (also)}}$, 'nejen—nýbrž', 'nadto'$_{\text{moreover}}$	conj.
'jakož i'$_{\text{as also}}$, 'natož'$_{\text{let alone}}$, 'však taky'$_{\text{but also}}$	conj.
'ale'$_{\text{but}}$, 'nýbrž', 'zato', 'ale zato', '(a)však'$_{\text{however}}$, 'nicméně', 'sice—(a)však', 'sice—nicméně', 'jen(om)že'$_{\text{though}}$, 'ovšem'$_{\text{indeed}}$, 'ale naopak'$_{\text{on the contrary}}$, 'a přece'$_{\text{and still}}$, 'a přesto', 'ani ne—jako spíše'$_{\text{not either—but}}$, 'sice—ale'$_{\text{otherwise}}$, 'ne tak—ale'$_{\text{not—but}}$, 'ani ne tak—jako spíše'$_{\text{not—rather}}$, 'jakkoli'$_{\text{although}}$, 'i když'$_{\text{even if}}$	anti.
'(a)nebo'$_{\text{or}}$, 'neboli', 'či', 'čili', 'aneb', 'eventuálně'$_{\text{possibly}}$, 'popřípadě', 'buď—(a) nebo'$_{\text{either—or}}$, 'ať už—nebo'$_{\text{whether—or}}$, 'ať—ať'	disj.

---

[145] 'As for your son, he is a clever boy' and 'Your son is a clever boy.'

Coordinate Conjunction	**Relation**
‘nebot’’$_{\text{for}}$, ‘vždyt’’$_{\text{why}}$, ‘totiž’$_{\text{namely}}$, ‘a proto’$_{\text{hence}}$, ‘a tedy’$_{\text{and thus}}$, ‘a tak’$_{\text{and so}}$, ‘a pak’$_{\text{and then}}$, ‘a to’, ‘v důsledku toho’$_{\text{consequently}}$, ‘když—tak’$_{\text{when—then}}$ ‘jak—tak’	eff.

The classification in the table is meant rather as leading instructions for the inference rules design, since for the logical analysis, we prefer (analogously with the causal clauses) to capture the coordination with fine-grained intensional propositional operators that one to one correspond to the conjunctions used in the sentence.

The approximative meaning of the conjunctions according to the conjunction groups stated in the table are summarized as:

1. the *conjunctive (conj.)* words serve to connect two or more clauses within a balanced relationship which could be in most cases analysed as propositions connected with the “logical and” operator. Clauses separated by mere comma are also in the conjunctive relationship and we analyse them as if they were joined by the ‘a’ (and) conjunction.

2. clauses separated with an *adversative (anti.)* conjunction assert such propositions where the content of the second one contradicts to the first one. An example of such sentence may be ‘Slunce pěkně vycházelo, ale obloha se brzy zatáhla.’[146]

3. likewise the conjunctive clauses the *disjunctive (disj.)* relationship between clauses may be analysed by means of a logical operator, in this case the “(exclusive) or”. Disjunctive clauses express assertion that debar each other — either one is true, or the other is true. As an instance of such sentence, we may take ‘Buď budeš pěkně zticha, nebo půjdeš spát.’[147]

4. the last group of coordinate relations is represented by the conjunctions *of effect (eff.)* or *reason for* the assertion in one of the clauses. In this case, the most convenient logical operator could be the “implication”, which, however, can be used only in the direction from an assertion to its effects and even there the implication may be too approximative. For examples, let as state the sentences ‘Kouření způsobuje rakovinu,

---

[146] ‘The sun was nicely rising, but the sky clouded over soon.’

[147] ‘Either you will be quiet, or you will go to bed.’

a proto na kuřiva stát uvalil vysokou daň.' (effect) and 'Letadlo nemohlo vzlétnout, protože byla hustá mlha'[148] (reason).

EXAMPLE

An example of analysis of coordinate conjunctions shall be demonstrated with the sentence 'Jaro končí, den se prodlužuje a začíná léto.'[149] Schematically, the analysis of the three-clause sentence can look like

$$\begin{aligned}\lambda w\lambda t\big[\mathbf{a}_{wt}\lambda w_1\lambda t_1[\ldots\textbf{jaro}\ldots\textbf{končit}\ldots]\\ \lambda w_2\lambda t_2[\ldots\textbf{den}\ldots\textbf{prodlužovat\_se}\ldots]\\ \lambda w_3\lambda t_3[\ldots\textbf{léto}\ldots\textbf{začínat}\ldots]\big]\end{aligned}$$ [150]

Here, the conjunction 'a' (and) is of type $(o\pi\pi\pi)_{\tau\omega}$, it expresses the conjunctive relation among three propositions. □

The instructions in this sections, should thus cover most of the common ways of combining clauses in the process of sentence building and graphically describe the techniques of their logical analysis in TIL.

5.4 Folding of Constituents

Practically all the sentence constituents may be folded in lists of constituents in NL. Examples of such lists can be adverbial phrases 'rozlobeně až zuřivě'[151] or adjectival group 'mladý, krásný a chytrý.'[152] As we can see from the first example, the constituents need not be connected just by commas and the word 'a' (and). Actually, the connection may be mediated with most of the coordinate conjunctions and even with some subordinate conjunctions (in which case, the constituent is a short for the appropriate subordinate clause).

Thus one possible approach to the analysis of lists of constituents could follow the same principles as in the case of clause conjunctions and let the constituents be joined by the corresponding conjunction operators.

However, we suppose that here is the right place to simplify the things a little, and thus, we try to analyse the junction by means of logical operators where it is possible. That means, in the analysis, that we take

---

[148] 'Smoking gives rise to cancer and that is why state has imposed a high tax on smokables.' and 'The airplane could not take the air because of a heavy mist.'

[149] 'The spring is ending, the day draws out and the summer is coming.'

[150] 'and', 'spring', 'to end', 'day', 'to draw out', 'summer', 'to start'

[151] 'with anger, even furiously'

[152] 'young, pretty and smart'

the conjunctive (together with adversative) and disjunctive conjunctions as the corresponding logical operators and the particles such as ‘až’, ‘dokonce’ or ‘i’[153] are ignored between subclausal constituents. In case of the other conjunctions (subordinate and coordinate expressing effect or reason), we, nevertheless, have to analyse them with the conjunction operator and for this sake, reconstruct the whole clause that is behind the folded constituent.

EXAMPLE

The noun phrase ‘mladý, krásný a chytrý člověk’[154] can be analysed as conjunction of the respective properties:

$$x \ldots \iota\colon\ \textbf{člověk}_{wt}x \wedge \textbf{mladý}_{wt}x \wedge \textbf{krásný}_{wt}x \wedge \textbf{chytrý}_{wt}x$$

Similarly, the sentence ‘Petr se tváří ne překvapeně, ale rozlobeně’[155] can be analysed as

$$\lambda w\lambda t\Big[\textbf{Does}_{wt}\textbf{Petr}\big[\textbf{Imp}_{w}v \wedge v = \textbf{tvářit\_se}_{w} \wedge \\ \wedge\, [\neg\textbf{překvapeně}_{w}]v \wedge \textbf{rozlobeně}_{w}v\big]\Big]^{156}$$

On the other hand, the sentence ‘Petr je chytrý, i když tak mladý’[157] must be analysed in the same way as if the adjectival phrase ‘chytrý, i když tak mladý’ (smart, though so young) were translated into a subordinate clause: ‘Petr je chytrý, i když je tak mladý.’[158]

□

5.5 Special Compound

Among special compounds, we include the constituents that express extensions like date or time. Such compounds can be identified during any of the phases of the analysis (lexical, syntactical or logical) and denoted as one compound. With such definition, we can count among special compounds also numbers and (quoted) strings of characters.

Their logical analysis then lies in labelling them with (i.e. stating them as names of) objects that serve as the mediators between such compounds and the extensions in demand.

---

[153] ‘even’

[154] ‘young, pretty and smart human’

[155] ‘Peter does seems not to be surprised, but angry.’

[156] ‘Peter’, ‘to seem to be’, ‘surprised’, ‘angry’

[157] ‘Peter is smart, though so young.’

[158] ‘Peter is smart, though he is so young.’

Following this, e.g. numbers are analysed as their own trivializations ($^{0}\mathsf{2}$ or $^{0}\mathsf{361}$) and date and time values are represented by the corresponding time classes and time moments constructible with constructions of the form $\mathbf{S}^{1.1.2001}$ or $\mathbf{S}^{18:30}$.

5.6 Questions and Imperatives

In the text so far, we have always been involved in the analysis of indicative sentences. What does the material stand for in the case of interrogative and imperative sentences?

We shall see from the following examples that all we have written about the analysis of (parts of) the indicative sentence, can be without changes used in the analysis of questions and imperatives.

First, let us present a few examples of interrogative and imperative sentences with their indicative equivalents:

(5.11a) Je Petr vyšší než Karel? (Is Peter taller than Charles?)
(5.11b) Petr je vyšší než Karel. (Peter is taller than Charles.)
(5.12a) Která hora je nevyšší na světě? (Which mountain is the highest in the world?)
(5.12b) Mount Everest je nevyšší hora na světě. (Mount Everest is the highest mountain in the world.)
(5.13a) Proč je Marie smutná? (Why is Mary sad?)
(5.13b) Marie je smutná, protože je Petr nemocný. (Mary is sad, because Peter is ill.)
(5.14a) Petře, uvař oběd! (Peter, make lunch!)
(5.14b) Petr uvaří oběd. (Peter will make lunch.)

What is the difference in meaning of (5.11a) and (5.11b)? Their analysis lies in both cases in ascribing the relation vyšší_než (taller than) to the two individuals, viz. Peter and Charles. The difference is just in the approach of the speaker, where in (5.11b) he asserts that the relation takes the value of True in the current world and time, while in (5.11a) the speaker does not declare anything about his knowledge of the propositions truth-value, but he rather wants the hearer to pronounce what the current truth-value of the fact is.

We can symbolize this attitude of the speaker by the notation of a *match*[159] of the form $x : C$ which we can take as a relation between an

---

[159] for more details see the Chapter 6.

object or variable x and a construction C in the sense that both sides construct (or are) one and the same object. Thus if $C^{5.11}$ is the π-construction that corresponds to the analysis of both (5.11a) and (5.11b), then the difference in the attitude to the question and to the answer is symbolized as

(5.11a') $x \ldots o : {C^{5.11}}_{wt}$
(5.11b') $\mathsf{True} : {C^{5.11}}_{wt}$

Here in (5.11a'), we express that we (the speaker) search for the value (to be stored in the variable x of type o) of $C^{5.11}$ at w and t. Since the answer consists of either "Yes" or "No", we call sentences like (5.11a) the *yes/no*-questions.

The sentences (5.12a) and (5.12b) do not talk about a mere truth-value — they talk about an object of another type, in this case, about an individual. The proper analysis of the question (5.12a) is a construction of the *property*[160] 'nejvyšší hora na světě' (highest mountain in the world). Schematically, the attitudes to the question and to the answer may be depicted as

(5.12a') $s \ldots o\iota : {C^{5.12}}_{wt}$
(5.12b') $\{\mathsf{Mount\_Everest}\} : {C^{5.12}}_{wt}$

where $C^{5.12}$ is the construction of the above mentioned property (i.e. an $(o\iota)_{\tau\omega}$-object) and the sought answer is the (singleton) class of all the objects that have the property $C^{5.12}$ at the current world and time. Such questions seek for "what" is the extension of some intension, we call them *wh*-questions.

The toughest nut from the viewpoint of the logical inference is represented by questions similar to (5.13a). When we need to cope with such a question, the only inducement which can lead us to the answer is that we seek for a proposition that can be stated either as a reason for the fact contained in the question or that results in the fact (the fact is a consequence of the sought proposition). Let us denote as Expl the function of type $(\pi\pi)_{\tau\omega}$ that takes every proposition p to its explanation $\mathbf{Expl}_{wt}p$, i.e. either p is a reason for $\mathbf{Expl}_{wt}p$ or $\mathbf{Expl}_{wt}p$ is a consequence of p (at w, t). Then again, we can symbolize the question (5.13a) and the answer (5.13b) with

(5.13a') $p \ldots \pi : \mathbf{Expl}_{wt}C^{5.13}$
(5.13b') $\lambda w \lambda t[\mathbf{nemocn\acute{y}}_{wt}\mathbf{Petr}]^{161} : \mathbf{Expl}_{wt}C^{5.13}$

---

[160] for now, we let alone the possibility of the analysis as an individual role.
[161] 'ill','Peter'

where $C^{5.13}$ is a construction of the fact that 'Marie je smutná' (Mary is sad).

In the case of imperative sentences, the analysis again differs from the analysis of a corresponding indicative sentence only in the attitude of the speaker to the particular assertion (and to the hearer, of course). Here, we do not have such a symbolic device as the match for questions, since imperatives usually do not serve as data for the inference mechanism. Their meaning consists in that by a command like (5.14a), that has as its analysis the corresponding indicative sentence (5.14b) (construction $C^{5.14}$), the speaker *wants* the hearer to do some activity which would eventually lead to the *fulfilment* of the command, i.e. that would cause the proposition constructed by $C^{5.14}$ to be True (either some time in the future, or at the time included in the imperative).

5.7 Putting It All Together

In all the previous sections of this chapter, we have specified (and in many cases also graphically demonstrated) the way how to analyse most of constituents of all the various kinds. We thus already know, how the logical analysis of a verb group with its arguments and adverbial modifiers should look like, how to analyse all kinds of noun phrases including adjectival groups, pronouns or numerals, or how we can build a construction of a sentence consisting of several coordinate or subordinate clauses.

What remains to be specified, is the basic guide-post that suggests the best order in which all the partial analyses of phrases, clauses and sentence should proceed.

We suppose that the input state of the logical analysis is formed by an already disambiguated (uniquely identified) derivation tree.[162] Hence, the logical analysis may run either after the end of the syntactical analysis of the input sentence, or as well in parallel with it, in which case the necessary procedures perform as certain contextual actions[163] which work over the possible combinations of the (locally) analysed constituents. The asset of such parallel approach is in its capability to prune analyses which are type-inconsistent, e.g. if the verb expects an individual as its argument, the type checking mechanism would not allow a proposition to take this place. However, the cases where such pruning may reduce the extent of the syntactic analysis are quite rare (remember, this pruning applies only on sentences which are correct in their syntax but inconsistent in the types of

---

[162] such as the examples of the first stage of the analysis in the Appendix A.

[163] see the Section 3.1.2.

their constituents) or they can be substituted with the verb frame analysis only. The drawback of the parallel analysis lies also in the time and space spent on the overabundant logical analysis of those subtrees that are not part of the resulting derivation tree (i.e., in the parallel analysis, we cannot cast away any subtree that is successful "so far," even if it may be ruled out within the successive analysis).

Anyway, whether the logical analysis runs in parallel with the syntactic analysis or not, the process should always proceed in certain (time) successive steps, which can be summarized as follows:

1. the logical analysis starts to build the construction of the whole sentence from *inside*, i.e., in concordance with the Frege's Functionality Principle, the meaning of the compound is constructed as the meaning of its constituents. Therefore the first step must necessarily run in the lowest part of the derivation tree — the analysis of the input lexical items. In this step, we have not much choice other than to look up the proper analysis (analyses) of the lexical items in the *lexicon*.[164] Hence, as a gain of this part, we receive the type of each lexical item as well as a schema of its working with other (dependent) constituents (e.g. a conjunction is accompanied with the schema of the relevant clause (propositions) as its arguments). Such a lexical item that expects some arguments to be meaningful is called a *functional* lexical item.

2. the analysis then moves up the derivation tree, rule by rule. Each rule is supplemented with a similar schema as the functional lexical items, a schema that tells how the constituents, that correspond to the nonterminals (or preterminals) on the right hand side, combine together to form a construction of the left side nonterminal. The result of the application of the schema is then subject to the type checking mechanism which safeguards that the constituents typologically agree with the others in the resulting construction, i.e. that all arguments of a composition have the types needed by the corresponding function.

   In this way, we form the constructions of constituents such are noun phrases or adverbial phrases up to the level of a clause.

3. in a rule of the form '`clause` → `...`', the process becomes a little more complex than to be described in one step only. In such rule, we

---

[164] At best, the lexicon can supply some wild-card values based on the grammatical category of the lexical item, but in such case we risk the possibility of incorrect type assignment (e.g. the word '`výška`' (height) cannot be analysed as an individual).

have identified the kind of the verb group[165] and in groups of so called intersegments we have the candidates for the verb arguments and free adjuncts in the form of noun phrases, prepositional noun phrases, other clauses or adverbial phrases. In several successive steps, we now need to form the construction corresponding to this particular clause:

(a) first, we try to identify the *subject* of the clause. In most cases, we can seek for a noun phrase[166] in nominative, however, we also should cope with subjects in the form of an infinitive (see the Section 5.1.7) or a subjective clause or translate such forms as a genitive subject ('nebylo tam <u>nikoho</u>' → 'nebyl tam <u>nikdo</u>'[167]) or a group subject 'at somebody's place' ('<u>U Nováků</u> dělají . . . ' → '<u>Novákovi</u> dělají . . . '[168]). So far, this part of the algorithm serves as an approximation of the processing of sentences with general agent — a thorough elaboration is currently beyond the scope of this work.

If the subject cannot be determined, we suppose that it is inexplicit and supply a indefinite subject of the type of individual or a class of individuals according to the number (singular or plural) of the verb.

(b) after that, we look up the *finite form verb* in the lexicon where we obtain all acceptable verb frames of this verb with the corresponding analyses (that includes the types of the verb arguments, as well).

(c) what follows is just a more tedious case of the procedure in point 2. In order to reduce the multiplicative extent of the number of participants to be checked during this process, we run one round of pruning yet before we start to build the construction — we check all the intersegments against the available verb frames and first, score out those that with certainty cannot take part in the verb frame, and secondly, check all the possibilities (based only on the stated grammatical categories) of their fitting in place in the verb frame (e.g. we do not allow two independent verb objects in accusative). After this, we obtain the possible

---

[165] i.e. whether it is an attributive or an episodic verb, active or passive voice and past, present or future tense.

[166] including a single adjectival group, optionally followed by an indeclinable word such as particle, adverb or interjection

[167] 'There was <u>nobody</u> there.'

[168] '<u>At Novák's place</u> they do . . . ' → '<u>Nováks</u> do . . . '

    verb arguments that are then type checked according to the requirements of the verb.

    (d) if we have linked in a relative clause or a clause with an inexplicit subject, we try to supplement it with the subject of the principal clause (i.e. if its verb and the subject agree in number and gender). Otherwise, we find the clause's subject as inexplicit.

   In this way, we obtain the construction of a clause.

4. eventually, we process the clauses' constructions according to their conjunction as we have described in the Section 5.3 about the sentence building process.

Thus, following these steps (and the guidelines provided in the previous sections), we can accomplish the logical analysis of the whole natural language sentence.

In the following chapter, we briefly disclose the eventualities of further processing and exploitation of the acquired analysis mainly in the form of knowledge representation and automatic reasoning or logical inference.

Chapter 6

Knowledge Representation and Reasoning

Currently, knowledge representation and reasoning (KRR) systems usually aim at applications where only limited expressive power is necessary, but rapid responses to questions are essential.[1] With this approach it is feasible to describe a selected field of interest with a possibly huge amount of facts and reuse this *knowledge* in an expert system tool.

Another approach, with a very small number of real implementations, is based on the fact that most of the human knowledge is encoded in the form of natural (non-artificial) language. Thus a straightforward way to handle such information is to build a system capable of analysing the sentences directly with a machine parseable output and a connection to an automatic inference machine.

A system that comprises of the Normal Translation Algorithm (NTA) and the TIL Inference Machine (TIM) is a tool for *Communication and Artificial Reasoning with* TIM, shortly CAT. It follows the approach of full natural language analysis with successive feeding of the acquired logical analysis result into the TIL inference machine. Even if the system as a whole is a long-time project, NTA and TIM put basis to two most significant parts of CAT. A schema of the way the particular parts of the CAT system cooperate is presented in the Figure 6.1.

In the next sections, we present a brief preview of the TIL inference mechanism with the sketch of the knowledge base design and the basic schema

---

[1] see e.g. [McGuin98].

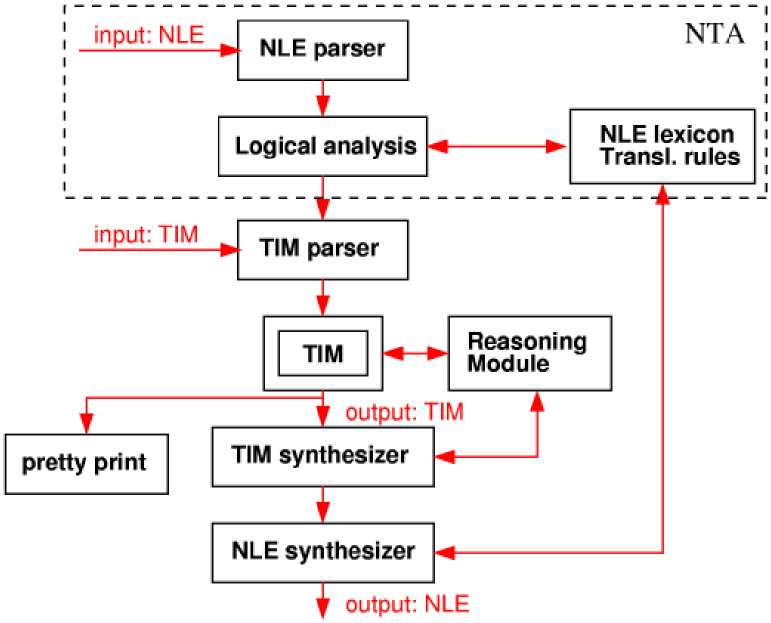


Figure 6.1: Schematic outline of the components of the CAT system.

of inference rules employed in the system. For thorough definitions of those notions, we send the reader to the work of Leo Hadacz.[2]

6.1 Knowledge Base of the TIL Inference Machine

We can look at the knowledge base (KB) design as consisting of *the ontology* and the description of the way *inference* is performed. The ontology is used as a specification of all concepts, that the inferring machine (TIM) knows about, only those concepts that commit to the ontology can ever be thought of as being processed by the machine. In our case the ontology is well defined by means of TIL. All objects that are stored in the knowledge base are derived directly from input constructions that reflect the meaning of facts that the user had implanted to TIM's memory. Every object has an equivalent type in the epistemic framework of TIL.

The TIM system knowledge base design is a form of a semantic network consisting of declarative memory for storing facts and procedural memory

---

[2] see [Hadacz2001].

for storing inference rules (a similar approach is used, e.g., in the Soar system [Soar93]). Facts in the declarative memory of Tim are strictly time-dependent, which means that each fact is bounded to the time moment, at which the fact was stored.

The declarative memory of Tim is realized by three databases called Tim Database of Constructions (TDBC), Tim Database of Types (TDBT) and Tim database of Values (TDBV). Those database are logically interconnected in a form of a directed acyclic graph with several edge types. The knowledge base can then be viewed of as a semantic network, since the edges represent the structural semantics of subconstructions following Frege's Functionality Principle in TIL.

`EXAMPLE`

The best way how to briefly present the KB structure is offered in the next example representing the facts that arise from the input sentence

$$\text{Petr přinesl včera své matce květiny.}^{3} \tag{6.1}$$

The equivalent construction in the Normal Translation Form (the output of NTA) can look like

$$\begin{aligned}\lambda w \lambda t \Bigg[\textbf{\textsf{P}}_t \Bigg[\textbf{\textsf{Onc}}_w \lambda w_1 \lambda t_1 (\exists x)(\exists z) \Bigg[&\textbf{\textsf{matka}}_{w_1 t_1} x \wedge \\ &\wedge \Big[\textbf{\textsf{Of}}_{w_1 t_1} x\, \textbf{\textsf{Petr}}\Big] \wedge \Big[z \subset \textbf{\textsf{květina}}_{w_1 t_1}\Big] \wedge \\ &\wedge \Big[\textbf{\textsf{Does}}_{w_1 t_1} \textbf{\textsf{Petr}} \big[\textbf{\textsf{Perf}}_{w_1} [\textbf{\textsf{přinést}}\, x\, z]_{w_1}\big]\Big]\Bigg]\Bigg] [\textbf{\textsf{Yd}}\, t]\Bigg]^{4}\end{aligned} \tag{6.1'}$$

$$(x \dots \iota; z \dots (o\iota); \textsf{přinést}/(o(o\pi)(o\pi))_\omega \iota(o\iota); \textsf{Yd}/\textsf{yesterday}/((o\tau)\tau))$$

When storing the facts in KB the existentionally quantified variables are replaced with newly allocated constants of the appropriate type (by the process of skolemization) or linked to already allocated objects.

The part of KB that represents our construction consists of 15 simple type objects (objects without subconstructions), 2 variables (w_1 and t_1) and 13 structured terms ($K_1, \dots, K_{13}$). The whole semantic network that represents this sentence is depicted in the Figure 6.2. The objects I_1 and C_1 are the constants that replaced the skolemized

---

[3] 'Peter has brought flowers to his mother yesterday.'

[4] 'mother', 'Peter', 'flower', 'to bring'

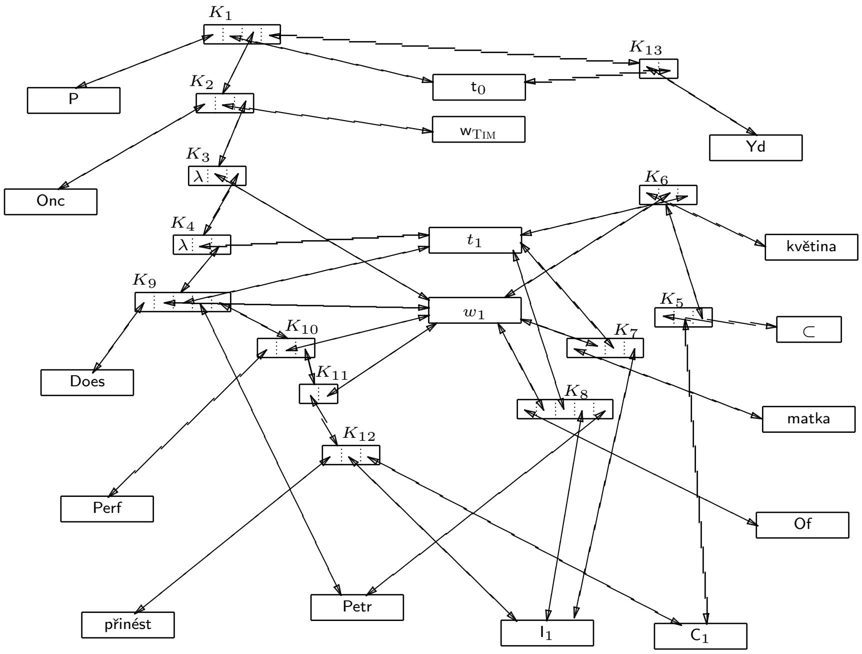


Figure 6.2: The part of knowledge base that comprises the construction (6.1').

> variables x and z. Two other constant objects t_0 and $\mathsf{w}_{\mathrm{TIM}}$ refer to the moment of the utterance of (6.1) and the reference world of TIM. Those constants were put into (6.1') on the place of the abstracted variables w and t. □

Such format of the knowledge base properly reflects the *reuse* property of constructions and subconstructions. This property is necessary for large knowledge databases, since they share the space of identical constructions, and on the other hand, it helps to identify all the constructions working over one topic (expressed by a subconstruction) in a very straightforward way.

6.2 Inference in TIL

The inference process lies in searching for an answer to a starting question with the help of previously inserted facts. As we have already suggested in the Section 5.6 about the analysis of various kinds of questions, the tool for expressing the difference in the attitude to an assertion and to a question is a so called *match*,[5] which is a pair of two components. The first component in the pair is called an atom — it is an object (i.e. a member of TIL type, a constant), a variable or a mark $\perp$ with the meaning of undefined value. The other component of a match is a construction. Intuitively, the atom in a match tracks the entity, which the construction constructs. We denote a match between an atom a and a construction C with $a : C$.

In a particular instance of the inference process, we assume that the propositions corresponding to the facts and the question are true in a (fixed) reference world w and at the time of utterance t. Thus a collection of the input facts and the starting question are expressed by a sequence of matches:

$$\begin{array}{l} \mathsf{T} : \mathsf{P}_{1\mathsf{wt}}, \ldots, \mathsf{T} : \mathsf{P}_{n\mathsf{wt}} \\ ? : Q_{\mathsf{wt}} \end{array}$$

For the representation of intensional objects, whose value (empirically) depends on a selected world and time, we use their instantiated tabular approximations with respect to Tim's knowledge. The original logical analysis of a property as 'a class of object that have that property in a certain world and time' is thus narrowed to the 'class of object about which Tim knows that they have the property in the questioned time.' Note that this is not truly equivalent to just referring them to the reference world w_{TIM}, since we usually *know* that there are more individuals with a certain property than those specified in the knowledge base. Otherwise, it would be true that any other object about which Tim does not know whether it has the property or not, would have to be counted as *not* having that property, which would inevitably lead to incorrect inferences.

In order to express the relativity of Tim's knowledge of the extensions of empirical objects, we have slightly modified the term match — in the automatic inference it is replaced with a *prevaluation* of the construction on the right hand side. A prevaluation a of a construction C is, similarly as a match, denoted as $a :: C$, but the meaning changes to 'as far as Tim knows, C constructs (the same as) a.' The input facts and the question that starts the inference are then expressed as

$$\begin{array}{l} \mathsf{v}_1 :: C_1, \ldots, \mathsf{v}_n :: C_n \\ ? :: C_q \end{array}$$

The task of the inference machine is then to find the answer $\mathsf{v} :: C_q$ by successive applications of inference rules on selected subsets of the collection of input facts and the question.

---

[5] see [Tichy82].

Note that the constructions are not relative to $\mathsf{w}_{\mathrm{TIM}}$ and the moment of the utterance, since TIM must (nearly always) infer the answer from facts that are related to a different time (and in some cases even to a different possible world). The inference rules must take the time flow into consideration and infer results from facts of various tenses, reference time spans and frequency adverbs.

The reduction rules, that form the second part of TIM's KB, the procedural memory, evolved from the inference system proposed by Tichý[6] that is similar to Gentzen's Sequent Calculus, in which the inference behaves like operations over sequents. In TIM, the reduction rules take the form of

$$\frac{P_1\ ;\qquad \ldots \qquad ; P_n}{C_1 \rightarrow C_2}\,(\mathrm{rR})$$

The essential constituents of a reduction rule are so called *metaconstructions*, i.e. constructional schemata that contain places (metavariables) that are to be replaced by actual constructions at the time of (a test of) the application of the rule.

The symbols C_1 and C_2 are the *input* and *output metaconstructions*, $P_1, \ldots, P_n$ represent *premises of the rule* — a premise is a certain form of a prevaluation of a (meta)construction or of a specific operation over (meta)constructions. Before a reduction rule can be applied, all the premises must be satisfied and the input (meta)construction must be unifiable with the actual input construction. Moreover, the limitations of the inference mechanism specify that only a closed construction can be a subject to reduction. An example of a reduction rule may be the rule for numeric addition

$$\frac{\mathsf{a} :: A\ ; \quad \mathsf{b} :: B\ ; \quad \mathrm{add}(\mathsf{a},\mathsf{b}) = \mathsf{c}}{[A\,{}^0{+}B] \rightarrow \mathbf{c}}\,(+\mathrm{R})$$

In this rule, we specify a way how to reduce the composition of the plus operation with two numerical constructions. If construction A has the prevaluation a and B has the prevaluation b then the construction (schema) $[A\,{}^0{+}B]$ can be reduced to the trivialization of the sum of a and b. An instance of the rule with its application can then look like the reduction $[{}^0\mathsf{3}\,{}^0{+}[{}^0\mathsf{2}\,{}^0{+}{}^0\mathsf{4}]] \ \xrightarrow{+\mathrm{R}}\ [{}^0\mathsf{3}\,{}^0{+}{}^0\mathsf{6}] \ \xrightarrow{+\mathrm{R}}\ {}^0\mathsf{9}$.

The inference in TIM is thus realized with sequences of reductions of constructions. Probably the most frequent is the well known reduction of β-equivalent constructions, but TIM has to comprise a number of various reduction rules for numerous kinds of actions.

---

[6] see [Tichy88, Tichy82]

At the time of the preparation of this thesis, TIM works mostly as a verification machine, i. e. an expert needs to input the sequence of reduction steps in order to find the prevaluation of a construction. The next steps of TIM's implementation thus include capabilities of automated reasoning, which will be based on the possibility of grouping the questions into a relatively small number of classes that share a common strategy of the answer induction, which then serves as a template for all questions in the same class.

Chapter 7

Implementation and Results

In this chapter we present some basic data that demonstrate the capabilities of the system as well as its efficiency. Firstly, we compare the results that we obtained in testing the parser for various parsing strategies. Then, we show the features of the generated grammar G2 and the expanded grammar G3 and discuss their analysis running times on sentences from the PDTB corpus. Then, we present several statistical data acquired from running the system with the DESAM corpus also with the estimate of the coverage and precision.

7.1 Parsing Strategies

During the development of the underlying parser, we have implemented and tested several parsing strategies and their variants:

- LALR parser — abandoned due to the need of serial processing of the exponential number of derivation trees
- top-down chart parser
- bottom-up chart parser
- head-driven chart parser
- generalized LR parser

The chart parsing technique and the GLR parser with its graph-structured stack have been designed for processing highly ambiguous grammar and thus they are suitable for grammars of natural languages.

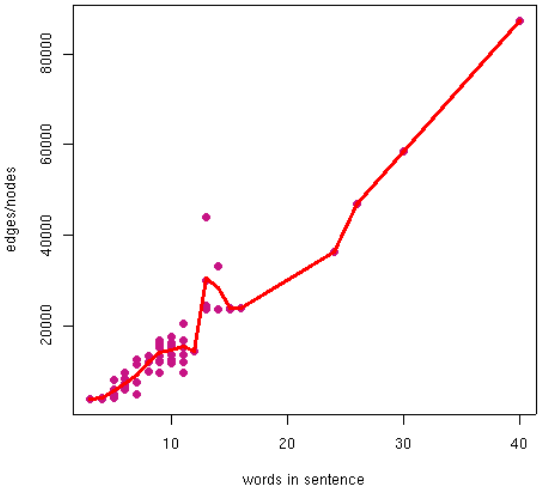


Figure 7.1: Map of words/edges for top-down chart parser

The basic efficiency of the parser can be expressed as the number of structures it creates when seeking for correct representations of the input syntax. In case of chart parsers, these structures are the chart edges, for GLR, the structures correspond to nodes in resulting packed shared forest. But note, that this measuring represents only “efficiency up to a constant”, i.e. two different parsers with the same number of structures can have their running times linearly depending on each other.

Various variants of chart parsing technique differ in the number of the overabundant edges, which are the main cause of a slowdown of the analysis. For the top-down analysis these edges are formed by the nonterminals on the way from the root nonterminal, which are found incompatible with the appropriate surface elements. For the bottom-up parser the extra edges come into existence as tree combinations over preterminals that cannot further compose a correct subtree. The head-driven variant tries to reduce the top-down excess edges while following the idea, that when we start with an important preterminal in the rule (its head), e.g. a noun in a noun phrase rule, the probability of later unsuccessful rejection of the rule decreases.

We have run the parsers with input formed of sentences of various length (from 3 to 40 words) and varying ambiguity rate and counted the resulting

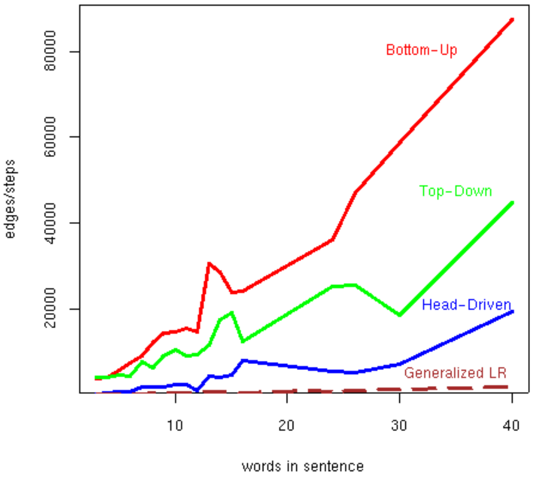


Figure 7.2: Comparison of words/parser structures for 4 parsing strategies

number of parser structures (chart edges/forest nodes). The obtained map (see the Figure 7.1) was then represented by an interpolated curve, which may be used as an estimate of the efficiency of the method. Comparison of all four methods is then displayed in the graph in the Figure 7.2. Note, that the line representing the generalized LR parser is counting different structures (forest nodes), and that the actual time must be multiplied with a constant. In correspondence with the presented results, we have chosen the head-driven chart parser as our "chief parser". Even though the GLR parser may achieve better running times, its application is handicapped by the need of time consuming process of building the transition table, which lasts for tens of minutes or hours, and loading the precomputed table before the actual analysis, which also takes (in our current implementation) more time than in the case of chart parsers. These features designate the implemented GLR parser as more suitable for processing bulk of input sentences in one run than starting the analyser for every single sentence and mark it as inapplicable in the time of designing the layout of the grammar due to the need of rebuilding the parser's transition table.

G1 meta-grammar – # rules	326
G2 generated grammar – # rules	2919
shift/reduce conflicts	48833
reduce/reduce conflicts	5067
G3 expanded grammar – # rules	10207

Table 7.1: Numbers of grammar rules in all three grammar forms

7.2 Rules Expansion

In our system, we work with a grammar of the Czech language, which is being developed in parallel with the parsing mechanism. The grammar in the three forms, as exemplified above, has the numbers of rules as stated in the Table 7.1

As a measure of the ambiguity rate of G2, we display the number of shift/reduce and reduce/reduce conflicts as counted with a standard LR parser generator. These data, together with the number of rules in the grammar, provide basic characteristics of the complexity of analysis.

The comparison of parsing times when using the grammars G3 and G2 is summarized in the Table 7.2. We present the time taken for parsing a selected subset of testing sentences — only sentences with more than 40 words were chosen.

The results show that in some cases, which are not so rare in highly inflectional languages, the expanded grammar achieves even lower running time than the original grammar. This effect significantly depends on the ambiguity rate of the input text. A question remains, how to exactly characterize the relation between ambiguity in the grammar and in the input.

The fully expanded grammar G3 is only moderately larger than the G2 grammar (about three times the size). The reason lies in the fact that the full expansion takes place mainly in the part of the grammar that describes noun phrases. This part forms only a small amount of the total number of G2 rules. Considering this, it is not surprising that the parse times are not much worse or even better. It also benefits from early pruning by transforming the unification constraints into the CFG. The agreement tests between subject and predicate should possibly also be expanded. Nevertheless, we have not put it to practice, since the position of subject is free, it cannot be described with CF rules without imposing a huge amount of ambiguity to every input sentence.

Sent #	**# of words**	**G2 # edges**	**G2 time**	**G3 # edges**	**G3 time**	**time G3/G2**
0006	42	30313	0.52	96834	0.76	146 %
0490	52	39945	0.96	93245	0.70	73 %
0588	53	36419	0.44	107813	0.80	182 %
0650	42	30976	0.76	73778	0.54	71 %
0724	58	63058	0.62	159613	1.14	184 %
0760	40	21197	0.49	60360	0.48	98 %
0843	63	122839	1.26	385399	2.72	216 %
1235	44	29291	2.37	216061	2.10	89 %
1633	45	26149	0.36	58371	0.46	128 %
1747	48	41955	0.52	123119	0.88	169 %
1782	52	20264	0.25	44825	0.36	144 %
2284	43	55835	1.22	133509	1.05	86 %
2338	40	30411	0.71	73203	0.52	73 %
2387	43	31589	0.40	77394	0.56	140 %
2609	46	23429	0.26	66969	0.52	200 %
2624	42	37809	0.37	99695	0.73	197 %
2781	49	93851	1.01	244110	1.73	171 %

Table 7.2: Running times for G2 and G3

7.3 Coverage and Precision

Features of systems of text analysis are usually summarized in a few numbers — *coverage* of the input sentences, *precision* of the produced analysis or various frequency variables, such as *recall* or *relative coverage*.

Our system has been designed as very robust to most kinds of Czech input sentences, allowing even such anomalies like a sentence consisting only of the subject or object constituent. That is why, we achieve very high estimates of the percentage of coverage of common input texts. However, the prompt usability of the results tends to be delayed due to the number of possible analysis, of which only a few appropriately reflect the *semantics* of the input sentence. We are currently working on supporting mechanisms, that enable the user to limit the resulting number of derivation trees according to various contextual variables, such as the valency frame of the verb, enlisted lexico-semantic constraints (see the Section 3.3.2) or the probability of the analysed attachment according to preterminals occurring in the input.

	# of sent.	percentage
successful at level 0, corpus	5150	51.5 %
successful at level 99, corpus	3986	39.9 %
successful at level 0, text	304	3.0 %
successful at level 99, text	211	2.1 %
unsuccessful	349	3.5 %
overall successful	9651	**96.5 %**
sum	10000	100.0 %

Table 7.3: System coverage measured on 10000 sentences

average time for sentence	0.17 s
minimum —||—	<0.01 s
maximum —||—	32.47 s
median of —||—	**0.09 s**
average number of words in sentence	15.4
minimum —||—	1
maximum —||—	73
median of —||—	**14**
average number of trees	$890 \cdot 10^{12}$
minimum —||—	1
maximum —||—	$5.7 \cdot 10^{18}$
median of —||—	**56**
average number of edges	6519.7
minimum —||—	81
maximum —||—	186329
median of —||—	**4181**

Table 7.4: Statistical data describing the analysis of 10000 sentences of corpus text

average number of words in sentence	13.89
minimum —‖—	4
maximum —‖—	33
median of —‖—	**13**
average number of trees	1339.2
minimum —‖—	1
maximum —‖—	32800
median of —‖—	**32**

	# of sent.	perc. %
hit precision of sentences of 1-10 words	32	100.0 %
hit precision of sentences of 11-20 words	37	80.4 %
hit precision of sentences of more than 20 words	8	57.1 %
overall hit precision	77	**83.7 %**
number of sentences with mistakes in input	8	8.0 %
number of sentences	100	100.0 %

Table 7.5: Statistical data describing the analysis of 100 sentences and their hit precision

We have measured the coverage and the precision on data from the DESAM_S corpus — a subset of the 1 million DESAM corpus with grammatical tags for words (see [PaRySm97]). The main feature of the corpus when building DESAM_S was the correct sentence marking, which predestines it to be a suitable test set. DESAM_S contains 315352 positions and consists of 18604 sentences.

In our measurements, we have run successive combinations of the following tests:

- level 0 analysis × highest level (99) analysis
- analysis of tagged text from corpus × analysis of plain text

The order of analyses was — level 0 analysis of corpus text, level 99 analysis of corpus text, level 0 analysis of plain text and level 99 analysis of plain text. Each of the tests was run only in case when the previous test was unsuccessful. The results (representing the coverage) of running the tests on 10000 sentences are displayed in the Table 7.3.

The analysed data are in more details described in the Table 7.4, where we present several statistical data (average, maximum, minimum and median) about the running time, number of words in sentences, number of resulting derivation trees and number of chart edges.

In order to be able to offer the other important measure of a parsing system, the precision, we need to specify what can be counted as a successful and correct analysis. We define it as the analysis, which correctly passes the parsing process (matches rules and actions) and among the output trees includes the (at least one) tree that reflects properly all the suprasyntactic relations (like PP attachment) in the particular input. This should not be interchanged with the usual meaning of precision, where we suppose that the system identifies directly the most probable analysis. To distinguish these two precisions, we call the percentage describing the portion of our correct analyses the *hit precision*.

Identifying the semantically appropriate derivation tree lies in a tedious process of preparing a treebank of sentences together with their analyses. Unfortunately, we cannot directly use the Prague Dependency Tree-Bank corpus, since structures in PDTB are dependency trees, which are not compatible with our derivation trees. That is why, we have created a small set of trees for 100 sentences and thus obtained a rough estimate of the system precision. The results are displayed in the Table 7.5. For the purpose of the exploitation of the PDTB dependency trees, we prepare a set of actions that allow to limit the analyses according to the information obtained from the dependency relations and in such way to speed up the treebank building process.

7.4 The Logical Analysis

Within this work, we have implemented the Normal Translation Algorithm described in the Chapter 5 for a selected subset of Czech sentences. The subset is specified on one hand with the coverage (and precision) of the syntactic parser and on the other hand on the coverage of the lexicon with the analysis of particular lexical items.

As we have already noted in the Section 5.7, which describes the overall process of the logical analysis of one sentence, the two parts of analysis, syntactic and logical, may be run either in parallel or as successive procedures. Since the advantages of the second approach seem to prevail,[1] we have decided to implement the logical analysis as a process that runs specifically

---

[1] see the argumentation in the above mentioned Section 5.7.

over one of the output trees (i.e. not over the whole chart) denoted by its rank in the probabilistic ordering of the trees.

For building the constructions of the constituents, we need to specify the description of the logical analysis on several levels:

- each *syntactic rule* must be supplemented with a schema which specifies (all) the possible ways of how could the constructions of the subconstituents be combined together in the rule. The acceptability of the results is always checked by the type checking mechanism.
- all the *lexical items* need to have their logical analysis described in the lexicon. In this way, we enlist the possible analyses of particular words or collocations (e.g. the preposition 'vzhledem k' (according to) is analysed as one lexical item).
- besides the primary lexicon, we keep two separate lexicons, the first of them being the lexicon of *verbs*. Together with every possible valency of a verb, we state the appropriate logical analyses of the verb arguments. The data in this lexicon are used during the processing of a clause, where the constructions of the verb arguments and the verb together compose the construction of the verbal object.
- the last lexicon contains idiomatic and terminological *collocations*, in which the meaning of their subconstituents is different from the usual meaning built as the corresponding composition. Examples of such collocations with their analysis are presented in the Section 5.2.1 on the adjective modifiers in a noun phrase.

  In this lexicon, the data refer to a particular syntactic rule with a specific instantiation of its constituents on the right hand side (the instantiation may be also specified as classes of lexical items taken either from the Wordnet hierarchy or enumerated in the lexicon).

Building these lexicons and rule's analyses is a long and tedious process which is, moreover, very often incomplete in the first run and needs to be amended in case a missing analysis is discovered. Due to these difficulties, we have currently filled the lexicon with only a few hundreds of testing items, which allows a very limited subset of Czech sentences to obtain a TIL construction of the sentence, although their syntactic analysis is correctly handled by the system. An important fact is that the limits of the coverage of the logical analysis are not due to the selected algorithm but due to the sparseness of the underlying (lexicon) data. So this problem can be solved either by manual “meaning” tagging of the words in the lexicons or with a (semi)automatic process that assigns the possible analysis according to an analogy between similar words or phrases with the control of a human expert. Nevertheless, a massive improvement of the automatic lexicon building techniques is still a matter of our research.

Chapter 8

Conclusions and Future Directions

The Normal Translation Algorithm is based on the underlying syntactic analyser of natural language sentences. Since, our aim was to specify the NTA for the Czech language where there is no publically available syntax parser so far, we inevitably needed to provide our own implementation. Nowadays, we have been working on an efficient analyser for more than three years. The used formalism is constituted on a special meta-grammar with a CFG backbone and a set of contextual actions and tests that assure the grammatical agreement of sentence constituents and drive the generation of the dependency output.

The meta-grammar contains powerful constructs that enable to reduce the number of rules which need to be maintained by human expert (linguist). Currently we have about 300 rules that are automatically expanded to more than 10000 rules and we are adding new constructs that will lead to further decrease of the number of the basic rules.

We have shown that shifting all possible feature agreement computations to the CFG backbone is suitable for free word order languages and it does not need to cause a serious increase in parsing time. We discuss three consecutively produced forms of our grammar and give a comparison of different parser running times on highly ambiguous input.

The CFG backbone enables us to use fast cardinal analyser that is based on a head-corner chart parser with probabilistic selection of new edges. In order to face up to the high number of obtained derivation trees, we define a sort ordering of the output trees that is specified by probabilities computed from appropriate edges in the chart structure. The statistics that correspond to a PCFG are involved in the process of sorting out the edges from agenda in the order that leads directly to N most probable analyses. Further work lies in the search of such statistical features that (in the form of the figure of merits of a constituent) provide the best correspondence between an expert-made syntactic analysis of a sentence and the analysis obtained as the result of our system.

In the process of parsers evaluation, we lacked the possibility to compare the parsing efficiency on a number of testing grammars. These grammars cannot be automatically generated, since they should reflect the situation in real-world parsing systems. As we have suggested at COLING'2000, future cooperation in NL parsing could therefore lead to the creation of a commonly shared bank of testing grammars with precisely specified ambiguity measures.

Even if the system is able to identify errors and mismatches in input, the system is not directly suitable for disambiguation of plain text due to the robustness of the grammar. The grammar design concentrates on finding all the possible combinations of constituents. Currently, the version that specifies the probability of each output analysis can be used for disambiguation of the text in a form of recommended attachments with grammatical agreement resolved.

The central acquisition of this work lies in the specification of the Normal Translation Algorithm for most of the phenomena of a natural language represented by Czech. In this specification, we have come out from the work of Pavel Tichý and his followers. However, we have not only compiled the previous works into a synoptical text, many parts of the NTA are new in this work and have never been explicated in this way in the literature. An example of a contribution to the theory of TIL is the new definition of concept in the Chapter 4.

The implementation of the NTA that is accompanying this work is still focused more on experiments with the algorithm explicated in the Chapter 5 than to an automatic sentence meaning analyser, mainly due to the need of large lexicons for the real-world data. However, despite of this fact, the first stage of the implemented NTA is available as a powerful syntactic analyser of Czech with a high percentage of coverage of common texts.

Nowadays, the implementation of the conjoint project of the TIL Inference Machine already in its early stage forms the basis of complex KRR system, which is capable of direct application to natural language expressions. With TIL as the underlying "semantic language," Tim promises to properly reflect all intricate phenomena of written human-to-human communication, such as intensionality or belief sentences.

Future implementation directions of the Tim system are aimed at natural automating of the verification process with the auxiliary facts induction without any expert interference. In that stage, we expect Tim and its covering project CAT to become a general information retrieval and reasoning tool with a large number of possible applications in computational semantics and other related fields.

Appendix A

Examples of Parsing System Output

In this appendix, we present several examples of the derivation trees that were obtained as the output of the parsing system. The sentences were obtained from the DESAM_S corpus of disambiguated Czech texts:

Ukázalo se, že vhodným způsobem naservírovaný dobrý nápad je lepší než jakákoliv půjčka.

(It was revealed that a good idea served in the appropriate way is better than any loan.)

Kvůli častějšímu broušení hlav vzrostly i náklady na provoz zařízení.

(On account of frequent grinding of the heads, the operating expenses increased.)

Postižených cestovek bylo víc, dvě z nich udělaly bankrot, vyprávěla nám Ludmila Janočková, majitelka agentury.

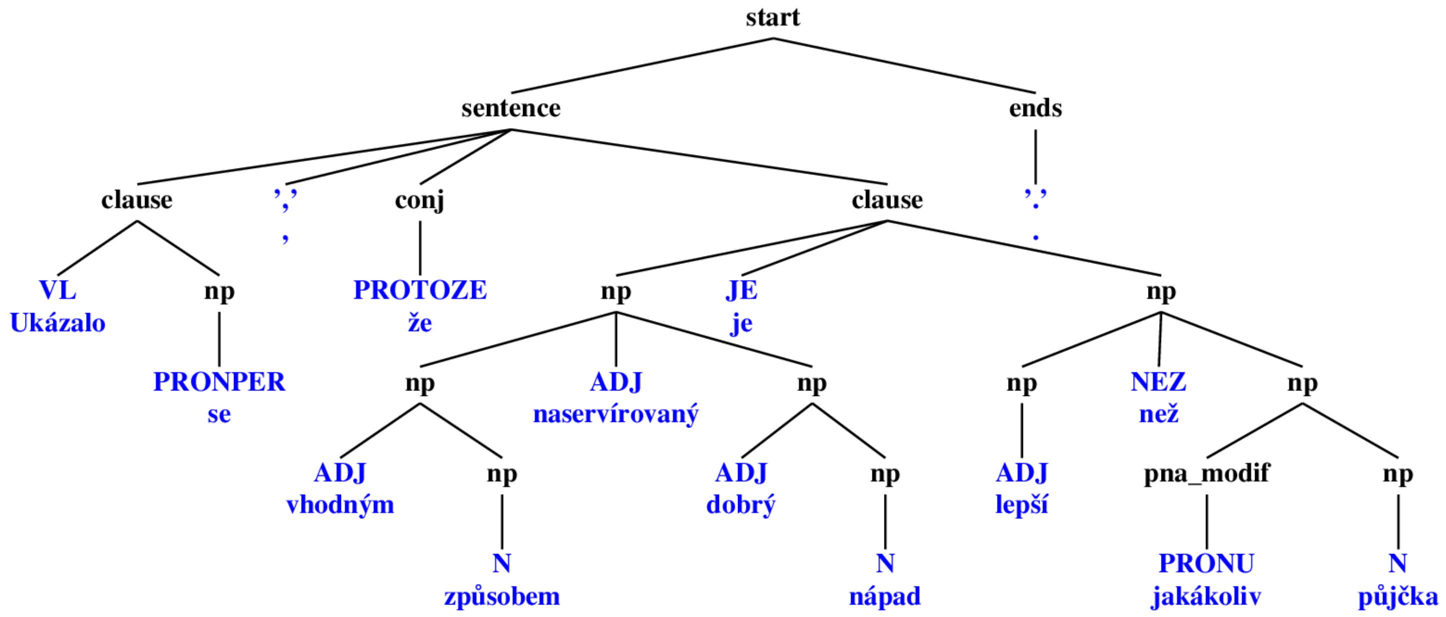
(There were more of the involved agencies, two of them have bankrupted, Ludmila Janočková, a head of the agency, said to us.)

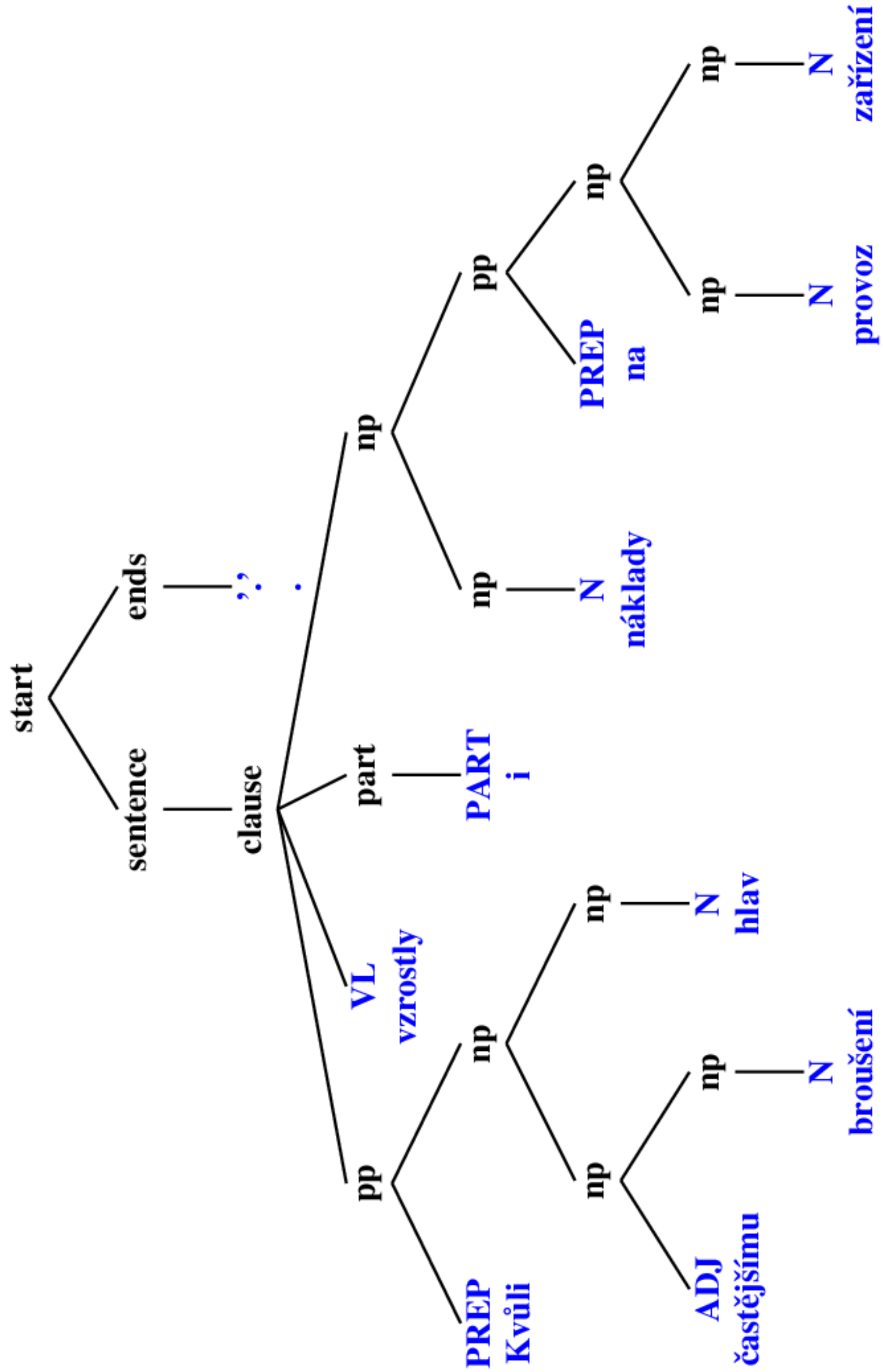
Pokud nechtěli nebo nemohli odklad platby povolit, použila k úhradám zálohy vybrané na připravené zájezdy.

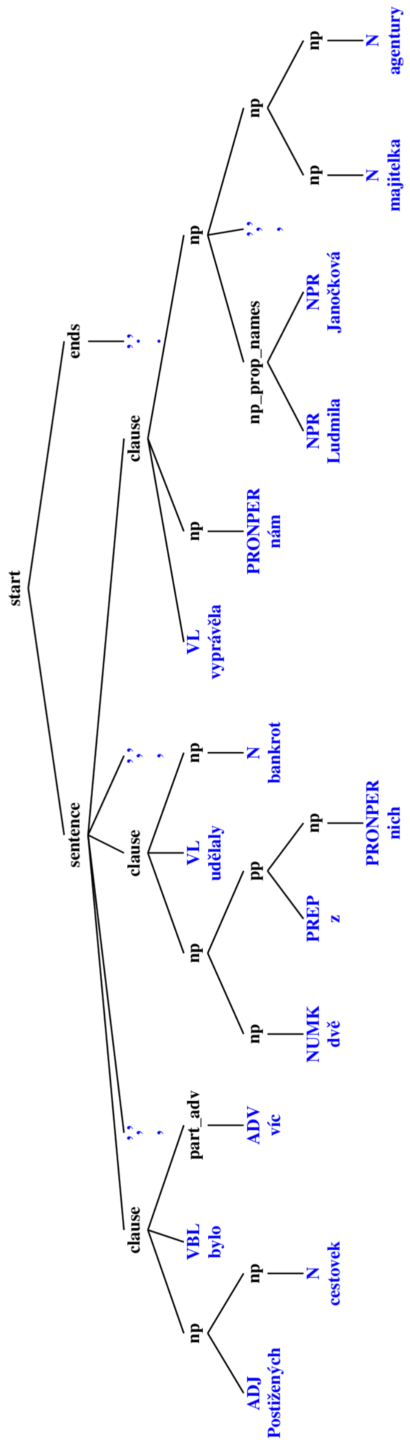
(If they did not want or were not able to allow the postponement of the payment, she used the deposits collected for the prepared tours to pay the expenses.)

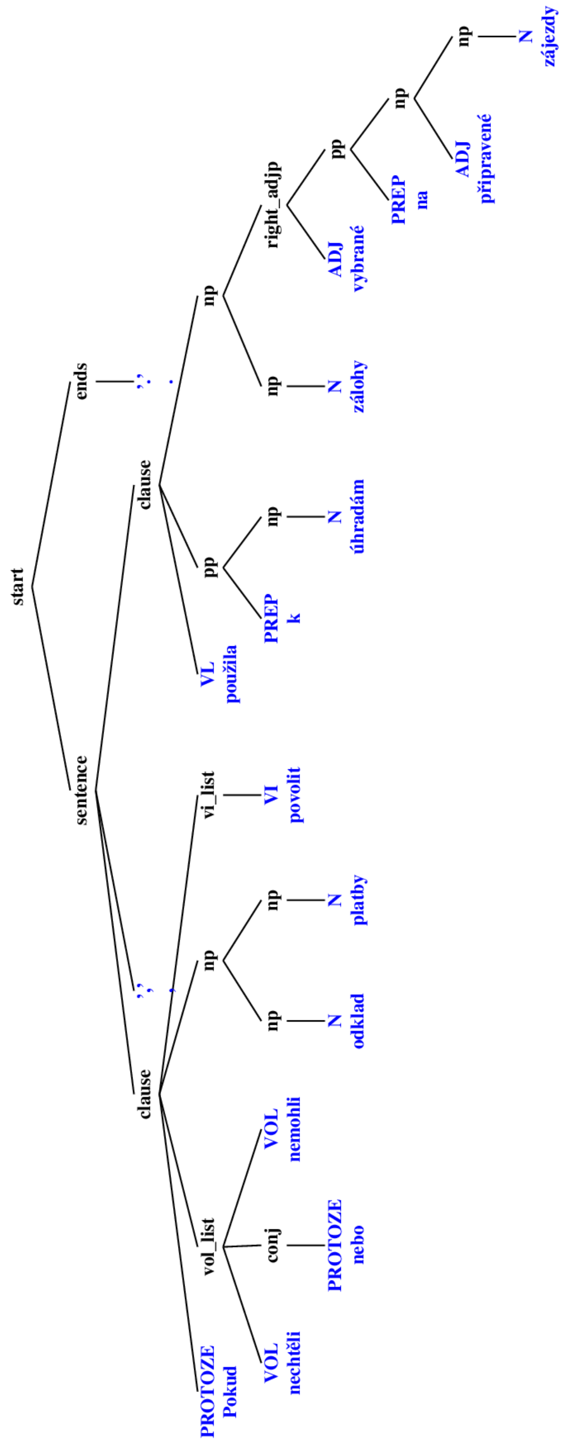
Byli jsme tři, kteří jsme získali malou továrničku na výrobu nábytku.

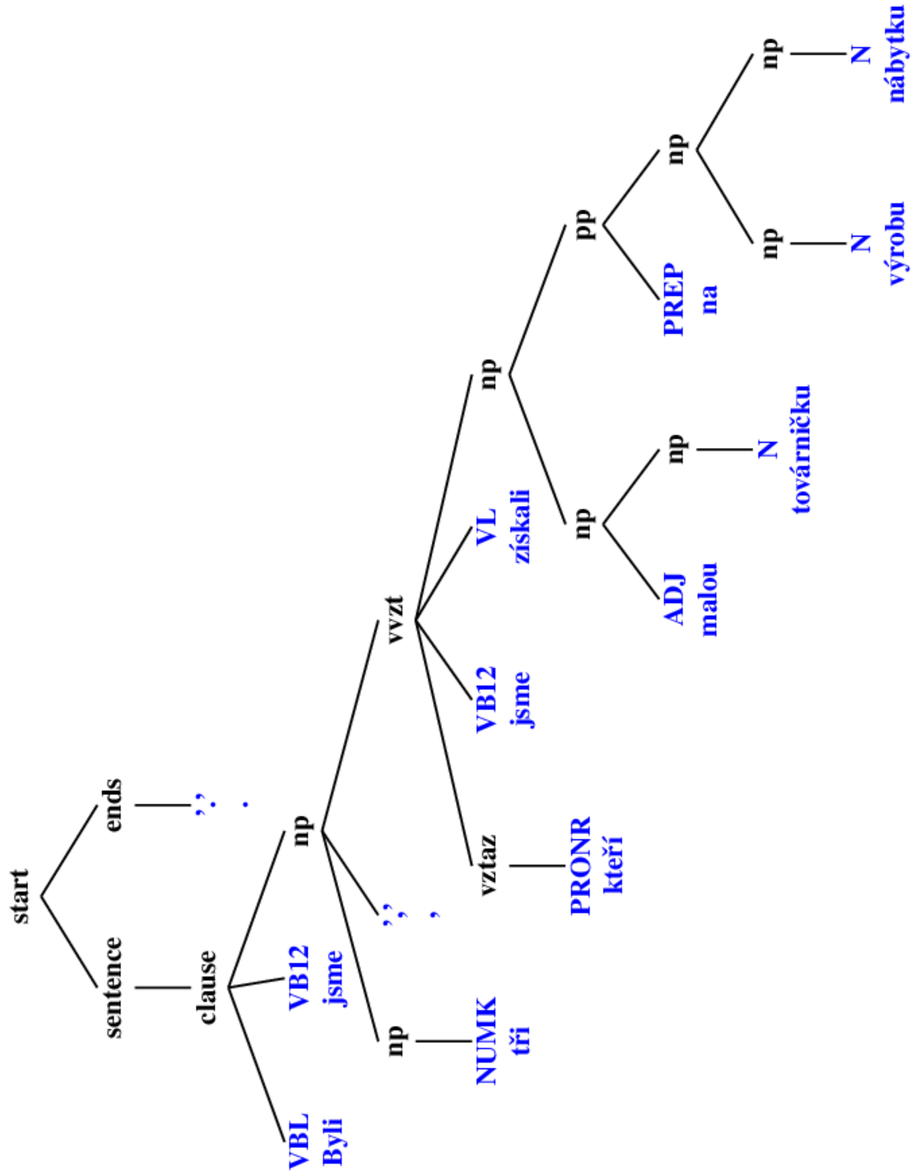
(We were three who acquired a small factory for the production of furniture.)


start
sentence
ends
clause
’,’
,
conj
clause
’.’
.
VL
Ukázalo
np
PROTOZE
že
np
JE
je
np
PRONPER
se
np
ADJ
naservírovaný
np
np
NEZ
než
np
ADJ
vhodným
np
ADJ
dobrý
np
ADJ
lepší
pna_modif
np
N
způsobem
N
nápad
PRONU
jakákoliv
N
půjčka


start
sentence
ends
clause
;
.
pp
PREP
Kvůli
np
np
ADJ
častějšímu
np
N
broušení
np
N
hlav
VL
vzrostly
part
PART
i
np
np
N
náklady
pp
PREP
na
np
np
N
provoz
np
N
zařízení


start
sentence
ends
'.'
.
clause
','
,
clause
','
,
clause
np
ADJ
Postižených
np
N
cestovek
VBL
bylo
part_adv
ADV
víc
np
np
NUMK
dvě
pp
PREP
z
np
PRONPER
nich
VL
udělaly
np
N
bankrot
VL
vyprávěla
np
PRONPER
nám
np
np_prop_names
NPR
Ludmila
NPR
Janočková
','
,
np
np
N
majitelka
np
N
agentury


start
sentence
clause
PROTOZE
Pokud
vol_list
VOL
nechtěli
conj
PROTOZE
nebo
VOL
nemohli
np
np
N
odklad
np
N
platby
vi_list
VI
povolit
,
clause
VL
použila
pp
PREP
k
np
N
úhradám
np
np
N
zálohy
right_adjp
ADJ
vybrané
pp
PREP
na
np
ADJ
připravené
np
N
zájezdy
ends
.


start
sentence
ends
'.'
.
clause
VBL
Byli
VB12
jsme
np
','
,
np
NUMK
tři
vvzt
vztaz
PRONR
kteří
VB12
jsme
VL
získali
np
np
ADJ
malou
np
N
továrničku
pp
PREP
na
np
np
N
výrobu
np
N
nábytku

Bibliography

[Allen94] J. Allen: *Natural Language Understanding.* Benjamin/-Cummings Publishing Company, Redwood City, 2^{nd} edition, 1994. 16

[BachHar79] K. Bach and R. Harnish: *Linguistic Communication and Speech Acts.* MIT Press, 1979. 5

[Brill95] E. Brill: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21, 4, pages 543–565, 1995. 2

[BtYacc98] C. Dodd and V. Maslov: *BtYacc – BackTracking Yacc, version 2.1*, 1998. `http://www.siber.com/btyacc/`. 27

[Chrz84a] T. Chrz: *Reprezentace znalostí při komunikaci člověk–stroj (The Knowledge Representation in the Human–Machine Communication).* PhD thesis, VÚMS, Prague, 1984. 16

[Chrz84b] T. Chrz: A System for Knowledge Representation Based on Intensional Logic. *Computers and Artificial Intelligence*, , 3, 4, pages 193–209, 305–317, 1984. 16

[EWN98] *EuroWordNet: Building a Multilingual Database with WordNets for Several European Languages.* 1998. `http://www.let.uva.nl/~ewn/`, CD-ROM. 34

[Frege1892] G. Frege: Über Sinn und Bedeutung (On Sense and Reference). In Geach and Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, Oxford, 1960. translation of the original from 1892. 4, 36

[GroeStok91] J. Groenendijk and M. Stokhof: Dynamic Predicate Logic. *Linguistics and Philosophy*, 14, pages 39–100, 1991. 13

[Hadacz2001] L. Hadacz: *Inference v transparentní intenzionální logice (Inference in Transparent Intensional Logic).* PhD thesis, FI MU, Brno, 2001. manuscript. 17, 54, 122

[Haj98] J. Hajič: Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In *Issues of Valency and Meaning*, pages 106–132, Prague, 1998. Karolinum. 30

[HajSgSk94] E. Hajičová, P. Sgall, and H. Skoumalová: An Automatic Procedure for Topic-Focus Identification. *Computational Linguistics*, 21, pages 81–94, 1994. 3

[HNS91] R. Heemels, A. Nijholt, and K. Sikkel, editors: *Tomita´s Algorithm: Extensions and Applications : Proceedings of the First Twente Workshop on Language Technology*, Enschede, 1991. Universiteit Twente. 28

[Horak98] A. Horák: Verb Valency and Semantic Classification of Verbs. In *Proceedings of TSD'98*, pages 61–66, Brno, 1998. Masaryk University. 34, 88

[Kadlec2001] V. Kadlec: Syntaktická analýza přirozeného jazyka (Syntactic Analysis of Natural Language). Master's thesis, Masaryk University, Brno, 2001. 28

[Kamp81] H. Kamp: A Theory of Truth and Semantic Representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language*, Amsterdam, 1981. Mathematisch Centrum. 14

[KapBres82] R. M. Kaplan and J. Bresnan: LexicalFunctional Grammar: A Formal System for Grammatical Representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281, Cambridge, Massachusetts, 1982. MIT Press. 3, 9

[Kaplan2000] R. M. Kaplan: Why not Cubic? In *Proceedings of the COLING'2000, Efficiency in Large Scale Parsing Systems Workshop*, pages 3–4, Luxembourgh, 2000. invited talk. 30

[Kay89] M. Kay: Head Driven Parsing. In *Proceedings of Workshop on Parsing Technologies*, Pittsburg, 1989. 27

[Koukol88] J. Koukolíková: *Temporální transparentní intenzionální logika v analýze přirozeného jazyka (Temporal Transparent Intensional Logic in the Analysis of Natural Language).* PhD thesis, ČVUT, Praha, 1988. 62, 66, 75

[Kubon99] V. Kuboň: A Robust Parser for Czech. Technical Report TR-1999-06, UFAL, Charles University, Prague, 1999. 6

[LavRos2000] A. Lavie and C. P. Rosé: Optimal Ambiguity Packing in Context-Free Parsers with Interleaved Unification. In *Proceedings of IWPT'2000*, Trento, Italy, 2000. 25

[LugStub98] G. F. Luger and W. A. Stubblefield: *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Addison Wesley Longman, Reading, MA, 3rd edition, 1998. 824 p. 4, 16

[Materna95] P. Materna: *Svět pojmů a logika (The World of Concepts, and Logic).* FILOSOFIA–ΦΙΛΟΣΟΦΙΑ, Praha, 1995. 36, 55, 56

[Materna98] P. Materna: *Concepts and Objects*, volume 63 of *Acta Philosophica Fennica.* The Philosophical Society of Finland, Helsinki, 1998. 36, 39, 42, 55, 56, 57, 99, 100, 103

[MaxKap91] J. T. Maxwell III and R. M. Kaplan: The Interface between Phrasal and Functional Constraints. In M. Rosner, C. J. Rupp, and R. Johnson, editors, *Proceedings of the Workshop on Constraint Propagation, Linguistic Description, and Computation*, pages 105–120. Instituto Dalle Molle IDSIA, Lugano, 1991. Also in Computational Linguistics, Vol. 19, No. 4, 571–590, 1994. 24, 25

[McGuin98] D. L. McGuinness and P. F. Patel-Schneider: Usability Issues in Knowledge Representation Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, Wisconsin, 1998. 121

[Montague73] R. Montague. The Proper Treatment of Quantification in Ordinary English: In J. Hintikka, editor, *Approaches to Natural Language*, pages 221–242. Reidel, 1973. 3

[Montague74] R. Montague: *Formal Philosophy.* Yale University Press, New Haven, 1974. 3, 36, 42

[Moore2000] R. C. Moore: Improved Left-Corner Chart Parsing for Large Context-Free Grammars. In *Proceedings of the 6th IWPT*, pages 171–182, Trento, Italy, 2000. 28

[Neidle94] C. Neidle. Lexical-Functional Grammar (LFG): In R. E. Asher, editor, *Encyclopedia of Language and Linguistics*, volume 3, pages 2147–2153. Pergamon Press, Oxford, 1994. 9, 25

[Noo97] G. van Noord: An Efficient Implementation of the Head-Corner Parser. *Computational Linguistics*, 23, 3, 1997. 27

[PaRySm97] K. Pala, P. Rychlý, and P. Smrž: DESAM — Annotated Corpus for Czech. In *Proceedings of SOFSEM'97*, pages 523–530. Springer-Verlag, 1997. Lecture Notes in Computer Science 1338. 134

[PaSe97] K. Pala and P. Ševeček: Valence českých sloves (Valencies of Czech Verbs). In *Proceedings of Works of Philosophical Faculty at the University of Brno*, pages 41–54, Brno, 1997. Masaryk University. 34

[Platek96] M. Plátek, T. Holan, B. Kuboň, and J. Hryz: Grammar Development & Pivot Implementation. final research report, Prague, 1996. JRP PECO 2824 Language Technologies for Slavic Languages. 6

[PollSag94] G. Pollard and I. Sag: *Head-Driven Phrase Structure Grammar.* University of Chicago Press, Chicago, 1994. 3, 8, 25

[Pulman96] S. G. Pulman. Language Analysis and Understanding: In R. A. Cole, editor, *Survey of the State of the Art in Human Language Technology*, pages 122–129. 1996. `http://www.cse.ogi.edu/CSLU/HLTsurvey/`. 3

[Russel1903] B. Russell: *The Principles of Mathematics.* Cambridge University Press, London, 1903. 36

[Sarkar96] A. Sarkar: Incremental Parser Generation for Tree Adjoining Grammar. In *Proceedings of the 34th Meeting of the ACL, Student Session*, Santa Cruz, 1996. 2

[Schabes88] Y. Schabes, A. Abeille, and A. K. Joshi: Parsing Strategies with 'Lexicalized' Grammars: Application to Tree Adjoining Grammars. In *Proceedings of the 12th COLING*, pages 578–583, Budapest, Hungary, 1988. 3, 10

[Sedlacek99] R. Sedláček: Morfologický analyzátor češtiny (Morphological Analyser of Czech). Master's thesis, Masaryk University, Brno, 1999. 2

[Sevecek96] P. Ševeček. LEMMA — Lemmatizer for Czech, 1996. C program. 2

[Sgall67] P. Sgall: *Generativní popis jazyka a česká deklinace (Generative Description of the Language and the Czech Declension).* Academia, Prague, Czech Republic, 1967. 88

[SgHaPa86] P. Sgall, E. Hajičová, and J. Panevová: *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects.* Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands, 1986. 88

[SikkAkk96] K. Sikkel and R. op den Akker: Predictive Head-Corner Chart Parsing. In H. Bunt and M. Tomita, editors, *Recent Advances in Parsing Technology*, Dordrecht, 1996. Kluwer Academic Publishers. 12

[SmrHor99] P. Smrž and A. Horák: Implementation of Efficient and Portable Parser for Czech. In *Proceedings of TSD'99*, pages 105–108, Berlin, 1999. Springer-Verlag. Lecture Notes in Artificial Intelligence 1692. 3

[Soar93] P. Rosenbloom, J. Laird, and A. Newell: *The Soar Papers: Readings on Integrated Intelligence.* MIT Press, Cambridge, MA, 1993. 123

[Tichy80a] P. Tichý: The Logic of Temporal Discourse. *Linguistics and Philosophy*, 3, pages 343–369, 1980. 36, 62, 65, 98

[Tichy80b] P. Tichý: The Semantics of Episodic Verbs. *Theoretical Linguistics*, 7, pages 264–296, 1980. 36, 62, 65, 68, 72

[Tichy82] P. Tichý: Foundations of Partial Type Theory. *Reports on Mathematical Logic*, 14, pages 52–72, 1982. 125, 126

[Tichy88] P. Tichý: *The Foundations of Frege's Logic.* de Gruyter, Berlin, New York, 1988. 3, 36, 42, 44, 53, 91, 126

[Tichy94a] P. Tichý: The Analysis of Natural Language. *From the Logical Point of View*, III, 2, pages 42–80, 1994. 36, 37, 62

[Tichy94b] P. Tichý: Cracking the Natural Language Code. *From the Logical Point of View*, III, 2, pages 6–19, 1994. 36, 41, 62, 105

[Tomita86] M. Tomita: *Efficient Parsing for Natural Languages: A Fast Algorithm for Practical Systems.* Kluwer Academic Publishers, Boston, MA, 1986. 11, 28

[VolkSch98] M. Volk and G. Schneider: *Comparing a Statistical and a Rule-Based Tagger for German.* 1998. `http://xxx.lanl.gov:80/ps/cs/9811016`. 2

[WordNet90] G. Miller: Five Papers on WordNet. *International Journal of Lexicography*, 3, 4, 1990. Special Issue. 34

List of Definitions

Summary

1 Algoritmus normální translace v transparentní intenzionální logice pro češtinu

Práce popisuje implementaci efektivního syntaktického analyzátoru českých vět, který je založen na pravděpodobnostní analýze typu head-corner chart parsing (tabulkový analyzátor s řídícím prvkem) s přidanými kontextovými akcemi a testy pro zajištění gramatické shody a generování výstupu ve tvaru stromu závislostí. Analyzátor pracuje s gramatikou českého jazyka s vysokým procentem pokrytí správných vět z korpusu. V teoretických kapitolách uvádíme stručný přehled nejčastěji používaných gramatických formalismů a technik syntaktické analýzy.

Analýza syntaxe nám slouží jako rozhraní v přirozeném jazyce pro Algoritmus normální translace (NTA), jenž poskytuje prostředek pro analýzu vět pomocí konstrukcí transparentní intenzionální logiky (TIL), které reprezentují význam vět. TIL je systém temporální logiky vyššího řádu s hierarchií typů založený na pojmu konstrukce jako nositele významu. V rámci práce podrobně popisujeme NTA a implementujeme jej pro vybranou podmnožinu českých vět.

Výstup NTA může být následně předán inferenčnímu stroji (TIM), který představuje vyvozovací mechanismus pro dedukci nových faktů ze zadaných vstupních konstrukcí. V práci popisujeme základní návrh struktury báze znalostí, jak je uložena v TIMu, a naznačujeme směr techniky inference ze znalostí uložených v této bázi.

2 The Normal Translation Algorithm in Transparent Intensional Logic for Czech

In the work we present an implementation of an efficient syntactic analyser of Czech sentences. The parser is based on a probabilistic head-corner chart parser with supplemented contextual actions and tests for grammatical agreement and dependency output generation. The analyser works with a grammar of Czech with a high percentage of coverage on correct corpus data. In the theoretical chapters we provide a brief survey of the most frequently used grammar formalisms and parsing techniques.

The syntax analysis offers a natural language interface to the Normal Translation Algorithm (NTA) that provides means for analysing sentences with constructions of transparent intensional logic (TIL), which then serve for representing the sentence meaning. TIL is a system of higher order temporal logic with a hierarchy of types built arround the central notion of construction as the meaning sustainer. We present a thorough description of the NTA and implement the algorithm for a specified subset of Czech sentences.

The output of NTA can be successively advanced to the TIL inference machine (TIM), which represents the reasoning mechanism for deduction of new facts out of the input constructions. In the work, we sketch the grounding of the inner form of the TIM knowledge base and indicate the way of the technique of inference from the knowledge that is stored in such base.