Determining Type of TIL Construction with Verb Valency Analyser

Pavel Smrž and Aleš Horák

Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic**
{smrz,hales}@fi.muni.cz

Abstract. In our paper we discuss an approach to semiautomatic corpus processing aimed at analysing verb valencies in Czech and consecutive determining the type of TIL (Transparent Intensional Logic) construction that belongs to the verb. Obtaining the type of the construction is a corner-stone of the logical semantic analysis of sentences. TIL is a highly suitable tool for representing the semantic structure of utterance as it is presented later in the paper. Our approach is based on the technique of partial syntactic analysis using a special kind of LALR grammar processing tool.

1 Introduction

Several approaches to semantic analysis have appeared during last decades. 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 [**?**]. Traditionally the first order predicate logic was used for the semantic description of language. As Montague [**?**] 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 try to overcome this weakness. However, as Tichy showed in his book [**?**], the Montague Semantics can run into severe problems when analysing certain kind of sentences, which are commonly used in natural language. That is why TIL was designed to represent semantic structure of the language by constructions.

TIL, or Transparent Intensional Logic, similarly as Montague Semantics, follows Frege's principle of compositionality, i.e. "The meaning of a sentence is a function of the meanings of its constituents" [**?**]. The basic idea of TIL lies in the presupposition that every well-defined language has a definite intensional base which can be explicated by an "epistemic" framework. Tichy 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

** The research is sponsored by the Czech Ministry of Education under the grant VS97028

object that represents the entity described by the analyzed expression is referenced 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.

The aim of TIL semantic analysis is to find an algorithm for associating language expression with equivalent construction. There is a three-leg way from the language expression to the (real world) object it identifies. The first step from the expression to the construction is a subject of semantic analysis. The connection between a construction and the constructed TIL object (the second part) is always fact-independent and it is directed by the mechanism of typed lambda calculus and thus it is well defined. The last leg of the journey is (mostly) dependent on the knowledge of the facts that hold in (and form) the actual world at the actual time.

In computational linguistics researchers try to device analytical tools that can process large amounts of corpus data without the need of human supervision. Automatic analysis based on TIL needs to find a translation algorithm that takes as its input a natural language sentence and outputs the corresponding TIL construction. The corner-stone of sentence meaning analysis is the semantics of the verb group with its arguments. Analysis of the verb groups are often based on Fillmore's semantic cases [**?**], verb frames and verb valencies.

Fillmore's semantic cases and verb frames are not suitable enough for Czech language which displays quite complicated case system (7 cases in both numbers). In Czech grammatical tradition, which prefers rather dependency oriented approach to syntax, valencies are widely used. If we decided to use Fillmore's semantic cases, we would have to somehow solve the conflicts between "deep" semantic cases and "real" grammatical cases existing in Czech. Our valency notation makes it possible to work with all 7 cases (nominative, genitive, dative, accusative, vocative, locative and instrumental) directly (to show an example). If there is a further need for semantic specification of the cases, it can be done by means of the appropriate semantic features and selectional restrictions.

2 Verb Valencies

In the following text we use the concepts of *valency expression* and *valency pattern* or valency. Valency expression is a schematic notation of a noun or adverb group or a clause, that expresses the requested obligatory attributes of the group or clause. Valency pattern for a given verb is formed by a set of valency expressions that express a scheme of a semantically correct part of sentence which contains the verb and appropriate noun or adverb groups or clauses. For example, the verb `vyvozovat` (infer) has two different valency patterns:

```
vyvozovat něco z něčeho    infer something from something
vyvozovat z něčeho , že    infer from something that
```

The format used for valency representation must be designed so that complies with the following requirements:

1. it describes all the syntactic information of the relationship between verbs and its arguments
2. it is easy to parse with computer tools
3. at the same time it must be effectively decodable by a human

The format we present meets the above points. The format describes the valency expression schema using the attribute-value pairs. The basic attributes and their values are enlisted in table **??**.

Table 1. The basic attributes of used valency notation

attribute **h** type (semantic features)	attribute **c** case (grammatical features)	attribute **s** clause (syntactic features)	attribute **r** preposition (syntactic features)
P, person	1, nominative	I, infinitive	*particular preposition in curly braces*
T, thing	2, genitive	C, conj. `až`	
Q, quality	3, dative	D, conj. `že`	
R, reflexive	4, accusative	F, conj. `zda`	
M, amount	5, vocative	P, conj. `at'`	
L, location	6, locative	R, rel. clause	
A, direction from	7, instrumental	U, conj. `aby`	
F, direction to		Z, conj. `jak`	
D, gen. direction			
W, time			

The transcription of valency patterns for the above mentioned verb `vyvozovat` then looks like this:

```
vyvozovat <v>hTc4-hTc2r{z},hTc2r{z}-sD
```

One can make an objection to the readability of the format. Actually linguists working with valencies may use the "verbose" format which corresponds to the linguistic tradition of valency notation in Czech. Of course, both the formats are equivalent to the feature structure representations usually assumed in recent grammatical theories.

3 Building a Valency List

Linguistics has been using the concept of verb valency for a long time, but, without the advantage of computer tools, the work with valencies is a very

lengthy and inevitably incomplete process, the results of which are of informative value only. At present new ways of getting and exploiting a valency list of a language seem to appear.

1. The first technique of building a list of verb valencies is the "manual" technique, when a researcher writes down valencies according to his or her linguistic knowledge or intuition. This technique, even if it may look archaic and inefficient way in computer processing, seems to be a needful one. Until complete and errorless tools for automatic processing of valencies are developed, the "manual" technique is convenient for making corrections and additions to the list or for building the core of the list.
2. The next technique, that is good to begin with when creating a valency list, consists in taking up a list of valencies that can be found in the form of a dictionary (see [**?**], [**?**]) after converting it into the electronic form. Although this technique is a good starting point, some typical difficulties arise during its realization, like a lack of the electronic version of the printed dictionary or inconsistent and out-of-date contents of such "manually" created list.
3. The third technique is based on exploring a language via its representative — text corpus (see [**?**,**?**]). If the corpus is large enough and satisfactorily exemplifying the language (which are the assumptions of a well built corpus), then this corpus technique is the most accurate one of all the stated techniques of building a valency list. It is highly probable that we can find all (used) variants of a given verb in corpus, and it is certain that all valency patterns which are obtained from corpus, are up-to-date, they are being used. An important feature of this technique is the possibility to obtain complete results, that do not contain processing errors, in a rather short time (when compared to the "manual" techniques). An initial disadvantage of the corpus technique is the need of tools working with raw natural language texts and capable of getting the verb valency patterns out of the text only with knowledge of grammatical attributes of the words that can be found in a tagged corpus. If we do not have tools for syntactic analysis or its output available, then the necessary tools must be relatively sophisticated programs, especially in case of variform Slavonic languages (Czech).

4 The Technique of Partial Syntactic Analysis

The partial syntactic analysis is conducted by the GC system. This system works with an LALR(1) grammar that allows the shift-reduce conflict to appear in any state. Such conflict is solved by successive processing of both branches of analysis.

The input to GC is essentially context-free grammar in machine-readable Backus-Naur Form (BNF) [**?**]. The description of contextual actions connected to each rule of the grammar contains higher grammatical functions that perform additional tests. The grammar is entered in this form:

```
noun-with-proper-names-group  ->  NOUN
```

```
          propagate_all($1)
noun-with-proper-names-group  ->  proper-name-group
          propagate_all($1)
noun-with-proper-names-group  ->  NOUN proper-name-group
          agree_case_number_gender_and_propagate($1,$2)
```

The GC system reads an input sequence of tokens (words tagged with a morphological analyser) and processes it according to the grammatical rules. If the input is correct, the system outputs a derivative tree of the given natural language sentence.

As we mentioned above some pre-defined grammatical tests and procedures can be 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-genitive-group  ->  noun-group noun-group
          test_genitive($2)
          propagate_all($1)
```

- agreement test of case in prepositional construction

```
prepositional-group  ->  PREPOSITION noun-group
          agree_case_and_propagate($1,$2)
          add_prep_ngroup($1)
```

- agreement test of number and gender for relative pronouns

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

- test of agreement between subject and predicate
- test of the verb valencies

```
clause  ->  subj-part verb-part
          agree_subj_pred($1,$2)
          test_valency_of($2)
```

The contextual actions `propagate_all` and `*_and_propagate` propagate all relevant grammatical information from the nonterminals on the right hand side to the one on the left side of the rule.

During the analysis the GC system builds a list of noun groups and adverbial groups (procedures `add_ngroup`, `add_prep_ngroup` and `add_adverb_group`) and a list of verb forms (`add_verb`). The relevant grammatical features of noun and adverbial groups are extracted and translated into valency patterns of found verbs. Eventually the valencies may be confronted with valencies from the existing list [**?**].

5 Assigning TIL Type According to Valencies Found

We use the valency list obtained by means of the GC system when we want to find the logical construction that corresponds to the verb meaning.

Having the valency list we want to find a distribution of all verbs into classes of equivalence. As equivalent we regard those verbs whose valency lists are similar. The algorithm of finding the similar valency lists for verbs first modifies the original valency list. The modifications are as follows:

1. In the valency list the valency expressions that are formed by a noun group with preposition (`hPr{}` or `hTr{}`) are (where it is possible) replaced by one of the expression `hL` (location), `hF` (direction from), `hA` (direction to), `hD` (way description) or `hW` (time).
   This mechanism is very important since we work with "raw" data from syntactic analysis as described in the previous paragraph. Thus the information about location, direction or time is often expressed in the form of a noun group with preposition which has to be translated into the corresponding valency.
2. The valency expressions of location and time are deleted from the valency patterns. The reason for this is that these expressions often represent adjuncts that display circumstantial meaning.
3. The valency lists for verbs modified in the previous steps are then sorted and duplicate valency expressions are left out. Resulting valency lists are compared eventually.

In such a way it is possible to define a decomposition of the set of verbs into classes of equivalence. The verbs in each class then share the same type of logical construction.

The Transparent Intensional Logic works with a hierarchy of types with the following four basic types: ι (individuals), o (truth values), τ (real numbers or time moments) and ω (possible worlds). Other types are then created as functions from one type to another one or as types of higher rank, that can run over constructions. Some important types are $\iota_{\tau\omega}$ (individual role), $(o\iota)_{\tau\omega}$ (a class of individuals or a property) or $(o\alpha\beta)_{\tau\omega}$ (an intensional relation between objects of types α and β).

If we want to translate a sentence into a construction, we first need to know the type of constructions that correspond to particular words in the sentence. Among them the construction representing a verb usually forms the basic part of the resulting construction and constructions of other words form its arguments. To determine the type of the verb construction seems to be more difficult than it is perhaps with a noun.

The classification of verbs described above divides verbs into groups with the same type of construction. Moreover, it is possible to formulate rules for deducing the type directly from the valency list for a verb. We derive the type from the valency list of a verb class in the following way — first we construct a set of all valency expressions that appear in the valency list for a verb, so called

multi-valency. The multi-valency is a schema of all possible expressions that can be tied with the verb, the verb "arguments". It also shows the number and kind of each argument. We assume that the verb expresses a relation between (at most) these arguments. In the sentence where some of these expressions are not present, the corresponding arguments are filled with null values. This approach allows to fill in a value of an argument that is missing in the sentence but is known from the preceding text and thus it semantically belongs to the verb.

The expressions are translated to verb arguments in the following ways:

1. `hQ` (property) is regarded as a property of individuals, $(o\iota)_{\tau\omega}$-objects.
2. `hM` (amount) expresses a number of some individuals, it is an extensional (not dependent on the actual world or time) relation between a number and an individual or individuals, a $(o\tau\iota)$-object (logical object of type $(o\tau\iota)$).
3. `hP` (person) and `hT` (thing) can express an individual role or a class of individuals, thus it has type $\iota_{\tau\omega}$ or $(o\iota)_{\tau\omega}$. Only during the analysis of a particular sentence it can be determined which one of these types should be used and in some cases it cannot be determined at all since the respective expression can be ambiguous.
4. `hA` (where to), `hF` (where from), `hD` (which way) and `hR` (reflexive pronoun) usually serve as modificators of the verb meaning. Therefore they do not change the type of the verb construction, they are functions that show the logical object expressing the modified meaning of a verb.
5. all `sX` expressions refer to another construction, thus they are of a higher rank type $*_n$.

For example, if we process the valency list of the verb `mít` (have) with the algorithm, we obtain a multi-valency `hA-hF-hPTc4-hPTc4r{za}-hPTc7r{s}-sI`, which yields the following construction[1]:

$$\lambda w/\omega.\lambda t/\tau.\lambda kdo/I.\lambda koho\_co/I.\lambda za\_koho\_co/I.\lambda s\_kym\_cim/I.\lambda inf/*_n\,.$$
$$[^0kam/((o*_n IIII)(o*_n IIII)_{\tau\omega})_{wt}$$
$$[^0odkud/((o*_n IIII)(o*_n IIII)_{\tau\omega})_{wt}$$
$$^0mit/(o*_n IIII_{\tau\omega})_{wt}]],$$

where $I = \iota_{\tau\omega}$ or $(o\iota)_{\tau\omega}$.

The construction can be schematically written as

```
modifier_where_to(modifier_where_from(
  have(
    sb_nomin,sb_st_accus,as sb_st_accus,with sb_st_instr,inf
  )
))
```

The constructions obtained by means of verb valencies represent the way how to extract the attributes of the verb meaning from the syntactic structure of the sentence.

[1] The object and variable names in the construction translated to English:
$\lambda w.\lambda t.\lambda sb\_nomin.\lambda sb\_st\_accus.\lambda as\_sb\_st\_accus.\lambda with\_sb\_st\_instr.\lambda inf.$
$[^0where\_to_{wt}\,[^0where\_from_{wt}\,^0have_{wt}]]$

6 Conclusions

The most important results lie in the implementation of the algorithm of partial syntactic analysis of Czech language that can automatically discover verb valencies in corpus data. We have also introduced an algorithm for determining the type of TIL construction associated with the verb meaning according to the list of its valency patterns. This procedure plays a key role in the system of TIL semantic analysis.

References

1. Pulman, S. G.,Language Analysis and Understanding, in *Survey of the State of the Art in Human Language Technology*, R. A. Cole, editor, pp 122–129, URL: `http://www.cse.ogi.edu/CSLU/HLTsurvey/`
2. Montague, R., The Proper Treatment of Quantification in Ordinary English, in *Approaches to Natural Language*, Hintikka, J., editor, pp 221–242, Reidel, 1973
3. Tichý, P.: *The Foundations of Frege's Logic*, de Gruyter, Berlin, New York, 1988
4. Frege G., Über sinn und bedeutung (On Sense and Reference), in Geach and Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, Oxford, translation 1960
5. Fillmore, C., "The Case for Case," *Universals in Linguistic Theory*, New York, 1968, pp. 1-88
6. Filipec, J., et al., *Slovník spisovné češtiny* (The Dictionary of Literary Czech), Academia, Prague, 1994
7. *Slovník spisovného jazyka českého* (The Dictionary of Literary Czech Language), Academia, Prague, 1989
8. Pala, K., Rychlý, P., Smrž, P., DESAM — approaches to disambiguation. Technical Report FIMU-RS-97-09, Faculty of Informatics, Masaryk University, Brno, 1997.
9. Pala, K., Rychlý, P., Smrž, P., "DESAM — Annotated Corpus for Czech," *Lecture Notes in Computer Science 1338*, SOFSEM'97, pp. 523–530
10. Aho, A. V., Sethi, R., Ullman, J. D., *Compilers — Principles, Techniques, and Tools*, Addison-Wesley, 1986.
11. Pala, K., Ševeček, P., "Valence českých sloves" (Valencies of Czech Verbs), *Proceedings of Works of Philosophical Faculty at the University of Brno*, Brno, 1997, pp. 41–54