# Formal tools for separating syntactically correct and incorrect structures *

**Martin Plátek, Vladislav Kuboň, Tomáš Holan**

platek@kki.ms.mff.cuni.cz, vk@ufal.ms.mff.cuni.cz, holan@ksvi.ms.mff.cuni.cz

Faculty of Mathematics and Physics

Charles University, Prague, Czech Republic

**Abstract**

In this paper we introduce a class of formal grammars with special measures capable to describe typical syntactic inconsistencies in free word order languages. By means of these measures it is possible to characterize more precisely the problems connected with the task of building a robust parser or a grammar checker of Czech.

## 1 RFODG

This paper is actually an abstract of [3]. The main topic of [3] is the introduction of a formalism called *RFODG (Robust Free-Order Dependency Grammar)*. This formalism was developed as a tool for the description of syntactically ill-formed sentences of a language with high degree of word-order freedom. The *RFODG* serves for the description of surface syntax; it provides the base of the parsing with subsequent evaluation of *syntactic inconsistencies* (violation of a syntactic rule) and localization of *errors* (message).

Every symbol of RFODG belongs to *terminals* or other symbols (*nonterminals*), to *deletable* or *nondeletable* symbols and to *positive* or *negative* symbols. The terminals of RFODG are the lexical categories of the morpholexical analysis, next pairs of sets serve for the classification and localization of syntactic inconsistencies.

[3] contains the definition of a *DR-tree* by *G*. The *DR-tree* should map the essential part of history of deleting dependent symbols, and rewriting dominant symbols, performed by the rules applied while (bottom to up) analysis.

*DR-tree* is similar to a standard derivation tree of CFG ff but a node (constituent) of a *DR-tree* may cover a discontinuous subsequence of the input sentence, the terminals can be used to create any type of nodes, not only the leaves and each node has a fixed horizontal position, which is shared by exactly one leaf. This property of *DR-trees* is used to localize syntactic inconsistencies in analyzed sentences.

We say that a sentence w is *positively parsed*, if there exists *DR-tree* for $w$ containing positive symbols only. Sentence $w$ is *robustly parsed*, if it is parsed, but not positively parsed.

The properties of *RFODG* allow to define three complexity measures, namely *local* and *global number of gaps*, the *size of gaps* and the *degree of robustnes*.

First three measures are defined on *DR-trees* by means of a notion of *coverage*. The coverage of a node $T$ may be intuitively described as a set of horizontal indices (the index of a terminal node is in fact its position in the sentence counted from left to right) of terminal nodes, which are derived from the node $T$. If the *DR-tree* is projective, the *local number of gaps*, *global number of gaps* and of course also the *sizes of gaps* are all equal to zero. Nonprojective *DR-tree* contains at least one gap.

The degree of robustness equals the number of negative symbols in a *DR-tree*.

The processing of the main phase of the system, so called grammar-checking analysis, is carried in three phases. The first phase checks whether the set of projective positively parsed trees is empty. If it is, then the second phase using and extended grammar containing also error anticipating rules and rules with relaxed constraints tests if exists a positively parsed nonprojective tree or a negatively parsed projective tree (with some limitations formulated via mentioned measures). The third phase checks the existence of a negatively parsed

*This work is supported by the Grant Agency of the Czech Republic, Grant-No. 201/96/0195 and by the RSS/HESP grant No.85/1995.

nonprojective trees. If any of the three phases finds a nonempty set of trees, the grammar checking ends and the evaluation module is called.

## 2 Evaluation of parses

The negative symbols can be classified into a number of groups (see [3]), the most interesting of which are the negative nonterminals signalizing an agreement inconsistency or some other type of inconsistency which can be corrected by some morphological changes of the word forms in the analyzed sentence. We denote such nonterminals as *mf-symbols*.

It is clear that there is a path leading up to the node which is assigned a pertaining *mf-symbol* from all nodes which contribute to the inconsistency signaled in that node. In order to be able to locate the source of this inconsistency we divide the set of rules into two subsets: the rules which transfer (these rules are called *mf-sensitive* or do not transfer the morphemic information *mf-insensitive*.

The report ([3]) introduces an exact definition of a notion of an *mf-component*. For the purpose of this paper it is possible to describe this notion informally as a smallest subtree of a *DR-tree*, which contains a particular *mf-symbol* and from which leads an *mf-insensitive* edge ( such an edge was created by the application of an *mf-insensitive* rule).

The evaluation phase is part of the system following the grammar-checking analysis. If the grammar-checking analysis ends in the first phase, the evaluation module is not invoked because the string being analyzed is considered correct.

If the grammar-checking analysis ends after the second phase, the evaluation receives the sets of trees $TR2$. In this case the first task of the evaluation module is to check whether the non-projective positive trees may be also considered as an expression of an error in agreement in the projective readings of the analyzed string $w$. From the set $TR2$ the evaluation selects the subset of those trees which do not contain other negative symbols than mf-symbols. This set is denoted as *TR-mf*. The dependency trees corresponding to *TR-mf* with marked *mf-components* will be enumerated (by the contraction of *DR-trees*). These trees contain possible agreement errors.

In case *TR-mf* is empty, the evaluation will not return any warning which means that the string being analyzed is considered to be correct (and nonprojective).

If the grammar-checking analysis ends in its third phase, it issues the set of trees $TR3$ for the evaluation. If the set $TR3$ is not empty, the evaluation returns dependency trees containing negative nodes in order depending on number of negative nodes and number of mf-components.

## 3 Conclusion

In [3] we have summarized the current stage of our research concerning grammar-checking of Czech language. Its main result is the development of the formalism capable of localization and classification of syntactic inconsistencies in languages with a high degree of word order freedom. It also provides a base for future exact research in the field of grammar checking and robust parsing and opens a number of questions requiring further investigations.

## References


[1] A.V. Gladkij: Formalnyje gramatiki i jazyki, Iz.: NAUKA, Moskva, 1973

[2] D.T.Huynh: Commutative Grammars: The complexity of Uniform word Problems, Information and Control 57, 1983, pp. 21-39

[3] T.Holan,V.Kubon, M.Platek: Formal tools for separating syntactically correct and incorrect structures. Technical report MFF UK, in press, Charles University Prague