20. Theorietag der GI-Fachgruppe AFS “Automaten und Formale Sprachen”

Baunatal bei Kassel

29.9.–1.10.2010

Friedrich Otto, Norbert Hundeshagen, Marcel Vollweiler (Hrsg.)

Vorwort

Die Fachgruppe AFS (früher Fachgruppe 0.1.5) der Gesellschaft für Informatik veranstaltet seit 1991 einmal im Jahr ein Treffen der Fachgruppe im Rahmen eines Theorietags, der traditionell eineinhalb Tage dauert, und auf dem auch die jährliche Fachgruppensitzung durchgeführt wird. Das erste solche Treffen fand 1991 in Magdeburg statt. Die weiteren Theorietage wurden in Kiel (1992), in Dagstuhl (1993), in Herrsching bei München (1994 und 2003), auf Schloß Rauischholzhausen (1995), in Cunnersdorf in der Sächsischen Schweiz (1996), in Barnstorf (1997), in Riveris bei Trier (1998), in Schauenburg-Elmshagen bei Kassel (1999), in Wien (2000 und 2006), in Wendgräben (2001), in der Lutherstadt Wittenberg (2002 und 2009), in Caputh bei Potsdam (2004), in Lauterbad bei Freudenstadt (2005), in Leipzig (2007) und in Wettenberg-Launsbach bei Giessen (2008) ausgerichtet. Seit dem Jahr 1996 wird dem eigentlichen Theorietag noch ein eintägiger Workshop zu speziellen Themen der theoretischen Informatik vorangestellt.

In diesem Jahr wird der Theorietag vom Fachgebiet "Theoretische Informatik" im Fachbereich Elektrotechnik/Informatik der Universität Kassel organisiert. Er findet statt vom 29.9. bis 1.10.2010 in Baunatal bei Kassel. Der Workshop am 29.9. steht in diesem Jahr unter dem allgemeinen Thema "Ausgewählte Themen der Theoretischen Informatik". Als Vortragende konnten

- Carsten Damm aus Göttingen,
- Markus Holzer aus Giessen,
- Peter Leupold aus Kassel,
- Martin Plátek aus Prag und
- Heribert Vollmer aus Hannover

gewonnen werden. Das Programm des eigentlichen Theorietags am 30.9. und 1.10. besteht aus 20 Vorträgen sowie der Sitzung der Fachgruppe AFS. In diesem Band finden sich die Zusammenfassungen aller Vorträge sowohl des Workshops als auch des Theorietags. Desweiteren enthält er das Programm und die Liste aller Teilnehmer.

Wir danken der Gesellschaft für Informatik für die finanzielle Unterstützung dieses Theorietags. Desweiteren danken wir Frau Djawadi ganz herzlich für ihre Hilfe bei der Organisation. Wir wünschen allen Teilnehmern einen interessanten und erfolgreichen Theorietag sowie einen angenehmen Aufenthalt in Baunatal.

Kassel, den 20.9.2010

Friedrich Otto
Norbert Hundeshagen
Marcel Vollweiler

Inhaltsverzeichnis

On the Prague Group of Mathematical and Algebraic Linguistics and Its Formal Tools*

Martin Plátek, Markéta Lopatková
Charles University in Prague,
Czech Republic
martin.platek@mff.cuni.cz

Introduction

Formal modeling of syntactic structure of a natural language, its syntactic analysis as well as synthesis, has an important impact on an insight into the characteristic features of the language and into the needs of its explicit description.

In this talk we focus on the basic tasks and methods developed within the *Functional Generative Description* of Czech (FGD), beginnings of which – connected with the name of Petr Sgall – date back to the 1960s. Both the 'classical' model based on pushdown automata (Section 1) as well as the current model adopting the framework of restarting automata (Section 2) are discussed.

1 The 'classical' model of Functional Generative Description

Functional Generative Description, as proposed by Petr Sgall in [12] and further developed by the *Group of algebraic linguistics* at Charles University in Prague, can be characterized as a *stratificational* and *dependency based* descriptive system for the Czech language.[1] The language description is split into layers, each layer providing a complete description of a (disambiguated) sentence and having its own vocabulary and syntax. Further, it adopts dependency formalism – syntactic information (both at the underlying and surface layers) is captured in a form of dependency trees: words are represented as nodes of

*The paper reports on the research supported by the grants of GAČR No. P202/10/1333 and 405/08/0681. It is carried under the project of MŠMT ČR No. MSM0021620838.

[1] This section is based on [6], which describes the experiments with testing the theoretical adequacy of FGD. The text can be found at the DBLP Bibliography Server.

the respective trees, each node being a complex unit with the lexical, morphological and syntactic features; relations among words are represented by oriented edges.

FGD was designed for *generating correct Czech sentences.* The 'classical' model consists of two components, a *generative component* and a *transducing component.*

The generative component

The generative component generates *tectogrammatical representations* (TR(s) in the sequel), i.e., "underlying representations on the level of linguistic meaning representing a specific patterning of extra-linguistic, ontological content, the set of generated strings surpassing only moderately the set of context-free languages", see [6, 11]. This component was originally based on a context free grammar combined with elements of dependency approach; later it was reformulated using exclusively pushdown store automata [12, 11].

This 'classical' model of FGD does not consider coordination and appositional constructions as these constructions go significantly beyond the straightforward concept of purely dependency-based approaches and require a more general formal model. A possible extended model (still based on pushdown store automaton) was introduced in [7].

The model imposes a significant constraint on a *projectivity* of a dependency tree, which allows for *linearized* representation of a dependency tree. Although this constraint conforms to the theoretical assumption on projectivity of TRs in FGD, it does not allow an adequate description of frequent non-projective surface constructions in Czech [1].

TRs describe all the linguistically relevant semantic information. These representations are disambiguated and identical for all synonymous surface variants. Thus the transducing component captures the relations of *synonymy* and *ambiguity* – it is able to generate all synonymous variants of a sentence from their common TR; on the other hand, an ambiguous sentence has several different TRs (one for each meaning).

The transducing component

The transducing component, which serves for translating the tectogrammatical representations of sentences to the lower layers of the language system, has the form of a sequence of pushdown store automata; the translation is split into steps that more or less correspond to the layers of language system (underlying and surface syntax, morphemics and phonemics/graphemics).

In [6], the model is described as follows: "The mathematical apparatus used for the transduction components of FGD is a sequence of pushdown store automata, transducing the TR into the surface representations (dependency trees) and the latter into morphemic ones (strings); then follows a finite automaton transducing the representation into the graphemic output form. [...]"

"Each transduction of the representation of the structure of the sentence to the adjacent level needs a pair of automata. The conditions constraining the transduction to the next level can be characterized as follows:

(a) In a given step only a single dependency syntagm (the governing word and its modification) is processed [...].

(b) A single pass through the sentence (in the text-to-rule order) is sufficient for every transducer.

(c) The process of transduction is based on the governing unit being handled by every pushdown automaton earlier that its modifications (dependent words). [...]"

The theoretical adequacy of the generative system and its practical usefulness were tested in an experimental implementation – a set of TRs of Czech sentences (mostly grammatically correct, though their meaningfulness could be doubted) were gained as a result of the procedure of random generation at the computer EC 1040, see esp. [6].

In 1980s a new system of translation schemes was designed, which made the interpretation in both directions possible; i.e., it worked as both a *generative* and an *analytical system* [8]. This model was based purely on dependency constructions.

2 The current framework for modeling FGD

Here we present our effort to formalize a basic linguistic method, an *analysis by reduction*, a method based on a stepwise simplification of an analyzed sentence. This method makes it possible to define formal dependency relations between particular sentence members – the (in)dependencies are obtained by correct reductions of Czech sentences – as well as to describe properly the complex word-order variants of a language with a high degree of free word order, including non-projective surface constructions. If we allow for rewriting analysis by reduction is also able to partially capture coordination and appositional constructions [4].

Analysis by reduction provided a crucial motivation for a new formal model of FGD based on the novel concept of *restarting automata* [2, 5]. Restarting automaton is a nondeterministic machine with a finite-state control unit, a finite characteristic vocabulary and a head which can read and process the symbols (words) of the sentence on a flexible working tape, marked by special symbols (the end-markers). This type of automaton starts its computation over an input sentence in the initial state with its head placed on the left end of the tape. A computation of a restarting automaton consists of cycles; the input sentence is processed – according to the transition relation of the automaton – until the sentence is accepted / rejected or until a restart operation is performed. Then the position of the head as well as the inner state of the control unit are 'forgotten' and the automaton starts processing the (already shortened) sentence from the beginning in a new cycle.

Modeling analysis by reduction (and consequently also syntactic analysis) with the use of restarting automata reflects the paradigm of FGD better than earlier models based on pushdown automata:

- Restarting automaton models adequately the syntactic relations determined by valency characteristics of lexical words. It makes it possible to perform several rewrite steps in a single cycle and thus to process a single verb (or noun, adjective or adverb) and all its valency complementations (as e.g., subject and object(s)) in a single computational cycle [4]. Therefore the restarting automaton reflects the *complete valency structures* as it is understood in the concept of dependency valency syntax; that distinguishes this model significantly from the models based on pushdown automata, which model *syntactic pairs* consisting of a governing and a dependent word.

- Restarting automaton makes it possible to capture the concept of *lexicalization* – the approach characteristic for dependency-based language description, which collects essential linguistic information in a lexicon.

- Restarting automaton reflects *non-local behavior of languages with free word order* – rewrite steps in such general models of automata are not restricted to the continuous substring of an input sentence; they can reduce several symbols with distant word-order positions (stored as discontinuous substrings on a working tape). Thus it can process words (and their complementations) with unbounded positions in a sentence as well as words forming non-projective (surface) constructions.

- Restarting automaton working in cycles models *recursive properties of a natural language* appropriately – first, the deepest embedded language constructions are processed, which results in the simplification of an analyzed sentence; then the language constructions embedded in such simplified sentence are processed. After each simplifying operation, a new cycle starts (i.e., the automaton restarts). The computation proceeds until the so-called core predicative structure is reached and accepted without any further restart or until the simplified sentence is rejected as an ill-formed sentence.

- The recent models of FGD based on restarting automata capture explicitly tree structures [9, 10] – restarting automaton is able to assign a set of parallel dependency structures to every reduction of an input sentence; these structures capture the correspondence of dependency trees on different layers of linguistic description, namely the tectogrammatical representation and the surface syntactic representation.

The talk at the TheorieTag will focus on the following issues:
(i) A linguistic background of FGD,
(ii) Formal tools connected with FGD and their adequacy,
(iii) A comparison between FGD and the 'Abhängigkeitsgrammatik' (developed by the group around Jürgen Kunze) [3], and
(iv) Current tasks in formal models of natural languages.

References

[1] Holan, T., Kuboň, V., Oliva, K., Plátek, M.: On Complexity of Word Order. *Les grammaires de dépendance – Traitement automatique des langues (TAL)*. Vol. 41, No. 1 (q.ed. Kahane, S.), p. 273-300, 2000.

[2] Jančar, P., Mráz, F., Plátek, M. Vogel, J: On Monotonic Automata with a Restart Operation. *Journal of Automata, Languages and Combinatorics*, Vol. 4, No. 4, p. 287-311, 1999.

[3] Kunze, J.: *Abhängigkeitsgrammatik*. Volume XII of Studia Grammatica, Akademie Verlag, Berlin 1975.

[4] Lopatková, M., Plátek M., Kuboň, V.: Modeling Syntax of Free Word-Order Languages: Dependency Analysis by Reduction. In *Proceedings of TSD 2005*, LNCS 3658, Springer Verlag, Berlin Heidelberg, p. 140-147, 2005.

[5] Otto, F.: Restarting Automata. In *Recent Advances in Formal Languages and Applications, Studies in Computational Intelligence*, Studies in Computational Intelligence, Vol. 25, Springer-Verlag, Berlin Heidelberg, p. 269-303, 2006.

[6] Panevová, J.: Random Generation of Czech Sentences. In *Proceedings of COLING 1982*, p. 295–300, 1982.

[7] Petkevič, V.: A New Formal Specification of Underlying Structure. *Theoretical Linguistics* Vol.21, No.1, 1995.

[8] Plátek, M.: Composition of Translation with D - trees. In *Proceedings of COLING 1982*, p. 313-318, 1982.

[9] Plátek M., Mráz F., and Lopatková M.: Restarting Automata with Structured Output and Functional Generative Description, In *Proceedings of LATA 2010*, LNCS 6031, Springer-Verlag, Berlin Heidelberg, p. 500-511, 2010.

[10] Plátek M., Mráz F., and Lopatková M.: (In)dependencies in Functional Generative Description by Restarting Automata. Accepted for NCMA 2010 in Jena.

[11] Plátek, M., Sgall, P.: A Scale of Context-Sensitive Languages. *Aplication to Natural Language, Information and Control*, Vol. 38, No 1, 1978.

[12] Sgall P., Nebeský L., Goralčíková A., and Hajičová E.: *A Functional Approach to Syntax in Generative Description of Language*. Elsevier Science B.V., Amsterdam New York, 1969.