Natural Language Engineering **2** (4) 365–366. Printed in the United Kingdom

© 1996 Cambridge University Press

Between finite state and Prolog: constraint-based automata for efficient recognition of phrases

KLAUS U. SCHULZ and TOMEK MIKOŁAJEWSKI
CIS, University of Munich, Oettingenstr. 67, 80538 Munich, Germany
e-mail: schulz@cis.uni-muenchen.de

(*Received 17 December 1996; Revised 5 February 1997*)

In computational linguistics, efficient recognition of phrases is an important prerequisite for many ambitious goals, such as automated extraction of terminology, part of speech disambiguation, and automated translation. If one wants to recognize a certain well-defined set of phrases, the question of which type of computational device to use for this task arises. For sets of phrases that are not too complex, as well as for many subtasks of the recognition process, finite state methods are appropriate and favourable because of their efficiency Gross and Perrin 1989; Silberztein 1993; Tapanainen 1995. However, if very large sets of possibly complex phrases are considered where correct resolution of grammatical structure requires morphological analysis (e.g. verb argument structure, extraposition of relative clauses, etc.), then the design and implementation of an appropriate finite state automaton might turn out to be infeasible in practice due to the immense number of morphological variants to be captured.

In such a situation, Prolog/Constraint Logic Programming (CLP) is often used as an alternative, since it allows for a much more compact representation of grammatical rules. However, it is questionable whether Prolog is an optimal choice: the built-in constraints of CLP languages Colmerauer 1990; Dincbas et al. 1988 are not tuned to control of morphological agreement conditions, and the lexical lookup in Prolog/CLP is based on traditional data base techniques or organized in an *ad hoc* way. Recognition of phrases is a very special task, and it is an open question if the heavy machinery of a general system like a CLP language leads to a loss of efficiency if compared to more specialized and limited concepts.

The full paper describes a new type of automaton that has been developed at CIS Munich for efficient recognition of phrases in large German corpora. The concept of a *constraint-based automaton* is tailored to sets of phrases where grammaticality depends on morphological agreement conditions. The new concept incorporates characteristic features both from finite state techniques and constraint programming. Efficient lookup technology based on transducers, as used for large scale electronic dictionaries of the DELA-type Silberztein 1993; Courtois 1990; Guenthner and Maier 1996, has been integrated. From finite state automata we inherit the linear surface structure of rules and deterministic execution. A special constraint solver, based on arc-consistency techniques Mackworth 1977, facilitates the control of

https://doi.org/10.1017/S1351324997001630 Published online by Cambridge University Press

agreement conditions for inflectional features. An adequate treatment of other linguistic concepts is supported through unification.

A miniature programming language has been developed and implemented in C for supporting efficient implementation of constraint-based automata. This language comes with a special optimization that temporarily gives back the full efficiency of pure finite state techniques during execution: a built-in preprocessing step for constraint-based automata extracts a finite state model for deterministic control of the automaton. Phrase recognition is then organized as a two phase process. In the first phase, lexical categories of input words are matched using only the deterministic finite state control. As a consequence, unification and constraint solving are eliminated in this phase, which leads to a considerable improvement of the performance. Once an admissible sequence of categories has been found by this filtering process, in the second phase the relevant segment of the input is fully analyzed by the constraint-based automaton. Here unification and constraint solving techniques as known from Prolog and CLP are invoked for checking morphological conditions.

In the full paper we present the results of a preliminary empirical evaluation, showing a sizable gain in efficiency compared to CLP implementations. We see various further possibilities for optimization, due to the special nature of the problem: for example, memoization techniques for the inflectional structure of German noun phrases can be used to reduce the amount of constraint solving during the analysis of a large corpus. On the implementation side, it should be possible to reduce the internal renaming (copying) of rules during phase two, since in practice each rule is only used a restricted number of times (typically once or twice) during the analysis of a candidate sequence.

References

Colmerauer, A. 1990. An introduction to PROLOG III. *Comm. of the ACM* **33**: 69–90.

Courtois, B. 1990. Un système de dictionaires électroniques pour les mots simples du français. *Language Français* **78**.

Dincbas, M. *et al.* 1988. The Constraint Logic Programming Language CHIP. In *Proc. of the 2nd International Conf. on Fifth Generation Computer Systems*, pp. 249–264.

Gross, M., Perrin, D. (eds.): 1989. Electronic Dictionaries and Automata in Computational Linguistics. Lecture Notes in Computer Science **377**. Berlin: Springer-Verlag.

Guenthner, F., Maier, P. 1996. Überblick zum CISLEX-Wörterbuch-System. *Lexikographica* **73**: 69–82.

Mackworth, A.K. 1977. Consistency in Networks of Relations. *AI Journal* **8** (1): 99-118.

Silberztein, M. 1993. *Dictionnaires électroniques et analyse automatique de textes.* Masson, Paris.

Tapanainen, P. 1995. RXRC Finite-State Compiler. Technical Report MLTT-020, Rank Xerox Research Center, Meylan, France.

https://doi.org/10.1017/S1351324997001630 Published online by Cambridge University Press