Journal of Automata, Languages and Combinatorics **12** (2007) 1/2, 117–138

REFINING THE NONTERMINAL COMPLEXITY OF GRAPH-CONTROLLED, PROGRAMMED, AND MATRIX GRAMMARS [1]

HENNING FERNAU

Fachbereich IV, Abteilung Informatik, Universität Trier
D-54286 Trier, Germany
e-mail: `fernau@uni-trier.de`

RUDOLF FREUND, MARION OSWALD

Institut für Computersprachen, Fakultät für Informatik, Technische Universität Wien
Favoritenstr. 9–11, A-1040 Wien, Austria
e-mail: `{rudi,marion}@emcc.at`

and

KLAUS REINHARDT

Wilhelm-Schickard-Institut für Informatik, Universität Tübingen
Sand 13, D-72076 Tübingen, Germany
e-mail: `reinhardt@informatik.uni-tuebingen.de`

ABSTRACT

We refine the classical notion of the nonterminal complexity of graph-controlled grammars, programmed grammars, and matrix grammars by also counting, in addition, the number of nonterminal symbols that are actually used in the appearance checking mode. We prove that every recursively enumerable language can be generated by a graph-controlled grammar with only two nonterminal symbols when both symbols are used in the appearance checking mode. This result immediately implies that programmed grammars with three nonterminal symbols where two of them are used in the appearance checking mode as well as matrix grammars with three nonterminal symbols all of them used in the appearance checking mode are computationally complete. Moreover, we prove that matrix grammars with four nonterminal symbols with only two of them being used in the appearance checking mode are computationally complete, too. On the other hand, every language is recursive if it is generated by a graph-controlled grammar with an arbitrary number of nonterminal symbols but only one of the nonterminal symbols being allowed to be used in the appearance checking mode. This implies, in particular, that the result proving the computational completeness of graph-controlled grammars with two nonterminal symbols and both of them being used in the appearance checking mode is already optimal with respect to the overall number of nonterminal symbols as well as with respect to the number of nonterminal

[1] Full version of a submission presented at the 7th Workshop on *Descriptional Complexity of Formal Systems* (Como, Italy, June 30 – July 2, 2005).

symbols used in the appearance checking mode, too. Finally, we also investigate in more detail the computational power of several language families which are generated by graph-controlled, programmed grammars or matrix grammars, respectively, with a very small number of nonterminal symbols and therefore are proper subfamilies of the family of recursively enumerable languages.

Keywords: Matrix grammars, graph-controlled grammars, programmed grammars, nonterminal complexity, appearance checking

1. Introduction

Nonterminal complexity is a classical measure of descriptional complexity of grammars. Within the area of regulated rewriting, graph-control can be seen as a framework in which many other rewriting mechanisms can be expressed, see [2, 3, 4, 16]. In particular, programmed grammars and matrix grammars can be seen as special cases of graph-controlled rewriting.

In 1984, Gheorghe Păun published a paper [14] where he showed that "six nonterminal symbols are enough for generating each r.e. language by a matrix grammar," as already the title of the paper indicates. Up to 2001, no better bound was published in this area. Then, at the MCU conference of 2001, two independent papers [5, 8] showed that actually three nonterminal symbols are sufficient to generate each recursively enumerable language by a graph-controlled grammar. However, there were some distinctive differences in the proofs of this result:

- Fernau (also see the journal version [6]) used a Turing machine simulation to obtain his result. He actually showed that three nonterminal symbols are enough in programmed grammars (a model that is more restrictive than graph control) to generate each recursively enumerable language. The results in [8] are weaker in this respect, since they only provide an upper bound of four on the nonterminal complexity of programmed grammars.
- The simulation of Freund and Păun makes use of the universality result of register machines with two registers as obtained by Minsky [13]. Since appearance checking is only needed in the actual simulation of the register machine with its two registers, only two out of these three nonterminal symbols are ever used in appearance checking mode; the third nonterminal symbol is only of interest to compute the conversion between the concrete recursively enumerable language whose acceptance is to be simulated and the specific representation of strings used in Minsky's result. In this respect, Fernau's result was weaker, since he used all nonterminal symbols in the appearance checking mode.

With respect to the two MCU'01 papers [5] and [8] mentioned above, a couple of natural questions regarding the optimality of the results obtained there remained open:

- What is the computational power of graph-controlled grammars with only one or two nonterminal symbols? In other words: are the MCU'01 universality results optimal?