Journal of Automata, Languages and Combinatorics **6** (2001) 3, 351–357

ON THE INCLUSION PROBLEM FOR VERY SIMPLE DETERMINISTIC PUSHDOWN AUTOMATA

Erkki Mäkinen
Department of Computer and Information Sciences
P.O. Box 607, FIN-33014 University of Tampere, Finland
e-mail: `em@cs.uta.fi`

ABSTRACT

We present a new algorithm for checking the inclusion of very simple deterministic pushdown automata. The inclusion "$L(M_1) \subseteq L(M_2)$?" is checked by first constructing a finite characteristic set R of M_1, and then checking whether or not the inclusion $R \subseteq L(M_2)$ holds.

Keywords: decidability problems, deterministic pushdown automaton, left Szilard language, characteristic set.

1. Introduction

The inclusion problem for simple deterministic pushdown automata is known to be decidable [2]. (For general deterministic pushdown automata, only the equivalence problem is decidable [6].) Wakatsuki and Tomita [7] have later studied the inclusion problem for very simple deterministic pushdown automata. They have shown that there exists a direct branching algorithm for solving this problem.

In this paper we show that there exists even a simpler algorithm for solving the inclusion problem for very simple deterministic pushdown automata. When solving the problem "$L(M_1) \subseteq L(M_2)$?" the algorithm first constructs a characteristic set of M_1. Characteristic sets of languages are used in grammatical inference algorithms [1]. Yokomori [8] has defined characteristic sets for very simple context-free grammars, i. e., for the grammars generating the class of languages accepted by very simple deterministic pushdown automata. Given a language $L(M_1)$, its characteristic set R has the property that $L(M_1)$ is the smallest language in the language class in question containing R. Hence, if also $L(M_2)$ contains R, then the smallest language $L(M_1)$ containing R must be included in $L(M_2)$. Otherwise ($L(M_2)$ does not contain R), it is clear that $L(M_1) \subseteq L(M_2)$ cannot hold. Thus, the characteristic set R is used as a kind of "test set": it is sufficient to check whether $L(M_2)$ contains the elements of the characteristic set.

The Wakatsuki-Tomita algorithm [7] is polynomial on the maximal *thickness* of the stack symbols. The thickness of a stack symbol is the length of the shortest derivation