Information and Computation 237 (2014) 257–267

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Removing nondeterminism in constant height pushdown automata ☆

Zuzana Bednárová [a,1], Viliam Geffert [a,1], Carlo Mereghetti [b,*,2], Beatrice Palano [b,2]

[a] Department of Computer Science, P.J. Šafárik University, Jesenná 5, 04154 Košice, Slovakia
[b] Dipartimento di Informatica, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy

ARTICLE INFO

Article history:
Received 13 November 2012
Received in revised form 26 February 2014
Available online 25 March 2014

Keywords:
Descriptional complexity
Finite state automata
Regular languages
Deterministic and nondeterministic pushdown automata

ABSTRACT

We study the descriptional cost of removing nondeterminism in constant height pushdown automata—i.e., pushdown automata with a built-in constant limit on the height of the pushdown. We show a double-exponential size increase when converting a constant height nondeterministic pushdown automaton into an equivalent deterministic device. Moreover, we prove that such a double-exponential blow-up cannot be avoided by certifying its optimality.

As a direct consequence, we get that eliminating nondeterminism in classical finite state automata is single-exponential *even with the help of a constant height pushdown store.*

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

The first and most relevant generalization to computational devices is certainly the introduction of *nondeterminism*. Among others, nondeterminism represents an elegant tool to capture certain classes of problems and languages. Although not existing in nature, a nondeterministic dynamic may in principle be "simulated" by actual paradigms, such as probabilistic or quantum frameworks (see, e.g., [1–5]). Nondeterministic variants of Turing machines, finite state automata, pushdown automata, and many other devices, have been studied from the very beginning.

The investigation of nondeterminism may be pursued along several lines. First, we may want to know whether nondeterminism really increases *computational power* of the underlying computational model. For example (see, e.g., [6,7]), it is well-known that nondeterminism does not increase the computational power of Turing machines or finite state automata. On the other hand, it is well-known that nondeterministic pushdown automata (NPDAS) are strictly more powerful than their deterministic counterpart (DPDAS), corresponding to the respective classes of context-free and deterministic context-free languages. Finally, for certain devices, the problem is not yet solved, e.g., it is still not known whether the computational power of two-way DPDAS and NPDAS coincides.

☆ A preliminary version of this work was presented at the 14th International Workshop on Descriptional Complexity of Formal Systems (DCFS 2012), Braga, Portugal, July 23–25, 2012.

* Corresponding author.
E-mail addresses: ivazuzu@eriv.sk (Z. Bednárová), viliam.geffert@upjs.sk (V. Geffert), mereghetti@di.unimi.it (C. Mereghetti), palano@di.unimi.it (B. Palano).

[1] Supported by the Slovak Grant Agency for Science under contract VEGA 1/0479/12 "Combinatorial Structures and Complexity of Algorithms" and by the Slovak Research and Development Agency under contract APVV-0035-10 "Algorithms, Automata, and Discrete Data Structures".
[2] Supported by the Italian MIUR under project PRIN-2010LYA9RH_005 "PRIN: Automi e Linguaggi Formali: Aspetti Matematici e Applicativi".

http://dx.doi.org/10.1016/j.ic.2014.03.002
0890-5401/© 2014 Elsevier Inc. All rights reserved.

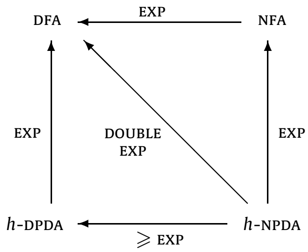

Fig. 1. Costs of conversion among different types of automata. Here h-NPDA (h-DPDA) denotes constant height NPDAS (DPDAS). An arrow labeled by EXP (DOUBLE EXP) from A to B means that an automaton of type A can be converted to an equivalent automaton of type B, paying by an exponential (double-exponential) increase in size. All costs displayed here are known to be *optimal*, except for the cost of h-DPDA $\leftarrow$ h-NPDA conversion, which we study in this paper.

Another interesting line of research is the study of how nondeterminism helps in presence of limited computational resources. Questions of this type, like P vs NP or L vs NL, go to the very heart of theoretical computer science (see, e.g., [6,7]).

In the realm of finite memory machines, the impact of nondeterminism on device efficiency is usually evaluated by considering its *size*. A classical result of this kind [8,9] establishes an optimal exponential gap between the number of finite control states in deterministic and nondeterministic finite state automata (DFAS and NFAS, respectively). This is usually credited as the first and motivating result in the area of descriptional complexity where the *descriptional power*, basically the size, of formalisms is under consideration (see, e.g., [10–12]).

In this paper, we tackle the impact of nondeterminism on descriptional complexity of *constant height pushdown automata* [13–16]. Roughly speaking, these devices are pushdown automata (see, e.g., [7]) with a built-in constant limit on the height of the pushdown, not depending on the input length. It is a routine exercise to show that their deterministic and nondeterministic versions accept exactly regular languages, and hence they share the same computational power. Nevertheless, a representation of regular languages by constant height pushdown automata can potentially be more succinct than by standard devices. In fact, in [16], optimal exponential and double-exponential gaps are proved between the size of constant height DPDAS, NPDAS, DFAS, and NFAS. The diagram in Fig. 1 briefly resumes such gaps.

The problem of optimal size cost of converting constant height NPDAS into their deterministic counterparts was left open in [16], and will be the subject of this contribution. What can be easily derived from Fig. 1 is that such a cost cannot be smaller than exponential. In fact, a sub-exponential cost combined with the optimal exponential cost for h-DPDA $\rightarrow$ DFA would lead to a sub-double-exponential cost of h-NPDA $\rightarrow$ DFA, thus contradicting the optimality of the double-exponential cost. We are going to show that *the cost of elimination of nondeterminism in constant height pushdown automata is double-exponential and, in general, this cost cannot be improved.*

After introducing some basic notation and results on constant height pushdown automata, we start by showing that any constant height NPDA, working with a finite state set Q, a pushdown alphabet Γ, and a constant pushdown height h, can be converted into an equivalent DFA with $2^{\|Q\|\cdot\|\Gamma^{\leqslant h}\|}$ states. This is obtained by first recording possible pushdown contents within the finite state control, so to obtain an equivalent NFA. From this NFA, by the standard power set construction, we obtain the claimed DFA, which is of course a constant height DPDA that does not actually use the power of the pushdown store.

One may think that a more sophisticated technique could lead to a smaller constant height DPDA, utilizing the capabilities of its pushdown store in a smarter way. However, we shall show that this is not the case. In fact, for each fixed $c \geqslant 2$, we can design $\{L_n\}_{n\geqslant 1}$, a family of languages over a $(c+1)$-letter input alphabet, such that:

- L_n can be accepted by a constant height NPDA using $O(c \cdot n)$ many states, pushdown height n, and c pushdown symbols, but
- no constant height DPDA for L_n can have, at the same time, the number of states, the pushdown height, as well as the number of pushdown symbols below $2^{\Omega(c^n)}$.

Stated in other words, the family $\{L_n\}_{n\geqslant 1}$ witnesses the optimality of the double exponential blow-up, obtained by the "naive" approach quoted above.

As a direct consequence, we get that eliminating nondeterminism in classical finite state automata is single-exponential *even with the help of a constant height pushdown store*.

Notice that a double-exponential lower bound cannot be established by standard pigeonhole arguments directly, since already a machine with a single-exponential pushdown height can reach a double-exponential number of different pushdown configurations. So we shall need to introduce a more sophisticated counting argument.

2. Preliminaries

The set of words on an alphabet Σ, including the empty word ε, is denoted here by Σ^*. By $|\varphi|$, we denote the length of a word $\varphi \in \Sigma^*$ and by Σ^i the set of words of length i, with $\Sigma^0 = \{\varepsilon\}$ and $\Sigma^{\leqslant h} = \bigcup_{i=0}^{h} \Sigma^i$. For a word $\varphi = a_1 \cdots a_\ell$, let $\varphi^{\mathrm{R}} = a_\ell \cdots a_1$ denote its reversal. By $\|S\|$, we denote the cardinality of a set S, and by S^{c} its complement.

Download English Version:

https://daneshyari.com/en/article/426757

Download Persian Version:

https://daneshyari.com/article/426757

Daneshyari.com