On Pushdown Store Languages* **

Andreas Malcher[1], Katja Meckel[1], Carlo Mereghetti[2], and Beatrice Palano[2]

[1] Institut für Informatik, Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany
{malcher,meckel}@informatik.uni-giessen.de
[2] Dipartimento di Informatica, Università degli Studi di Milano
via Comelico 39/41, 20135 Milano, Italy
{carlo.mereghetti,beatrice.palano}@unimi.it

Abstract. We design *succinct nondeterministic finite automata* accepting *pushdown store languages* — i.e., the languages consisting of the pushdown contents along accepting computations of pushdown automata. Then, several restricted variants of pushdown automata are considered, leading to improved constructions. Finally, we apply our results to decidability questions related to pushdown automata.

Keywords: pushdown automata; pushdown store languages;

1 Introduction

Beside the formal definition of the accepted or generated language, the introduction of an accepting or generating device always brings the attention to several "auxiliary" formal structures related to the device itself (see, e.g., [5, 10]). Such structures are not only interesting *per se*, but their investigation has often other relevant motivations.

In this paper, we focus on *pushdown store languages* for pushdown automata (PDA). Given a PDA M, its pushdown store language $P(M)$ consists of all words occurring on the pushdown store along *accepting* computations of M. It is known from [1, 4] that, surprisingly enough, $P(M)$ is *regular*. Here, we design *succinct nondeterministic finite automata (NFA)* for $P(M)$. In Section 3, we outline the construction of NFA, whose size (i.e., number of states) is *quadratic* in the number of states and linear in the number of pushdown symbols of M. Then, we show that this size bound cannot be improved in general by pointing out its asymptotical optimality. In Section 4, we deal with restricted versions of PDA, namely: PDA which never pop, stateless PDA, and counter machines. For any of these restrictions, we present optimal NFA for pushdown store languages, which are strictly smaller than the NFA given for the general case. Finally, in Section 5, we apply these results to the analysis of the hardness of some decision

* Partially supported by CRUI/DAAD under the project "Programma Vigoni: Descriptional Complexity of Non-Classical Computational Models."

** An enlarged version of this work has been accepted at the *14th Int. Workshop on Descriptional Complexity of Formal Systems (DCFS) 2012*, and will appear in the Springer LNCS series.

problems related to PDA. We show that the questions of whether $P(M)$: (i) is a finite set, or (ii) is a finite set of words having at most length k, for a given $k \geq 1$, or (iii) is unary, can be answered in deterministic polynomial time. Moreover, we also prove the P-completeness of these questions. As an application, we obtain that it is P-complete to decide whether a given unambiguous PDA is a constant height PDA [2, 3], or is a PDA of constant height k, for a given $k \geq 1$, or to decide whether a given PDA is essentially a counter machine. Due to lack of space, some proof details are moved to the Appendix.

2 Preliminaries

A *pushdown automaton* (PDA, see e.g., [6]) is formally defined to be a 7-tuple $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$, where Q is a finite set of states, Σ is a finite input alphabet, Γ is a finite pushdown alphabet, δ is the transition function mapping[3] $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma$ to finite subsets of $Q \times \Gamma^*$, $q_0 \in Q$ is the initial state, $Z_0 \in \Gamma$ is a particular pushdown symbol, called the bottom-of-pushdown symbol, initially appearing on the pushdown store, and $F \subseteq Q$ is a set of accepting (or final) states. Roughly speaking, a *nondeterministic finite automaton* (NFA) is a PDA where the pushdown store is never used. A *configuration* of M is a triple (q, w, γ), where q is the current state, w the unread part of the input, and γ the current content of the pushdown store, the *leftmost* symbol of γ being the *top* symbol. For $p, q \in Q$, $a \in \Sigma \cup \{\lambda\}$, $w \in \Sigma^*$, $\gamma, \beta \in \Gamma^*$, and $Z \in \Gamma$, we write $(q, aw, Z\gamma) \vdash (p, w, \beta\gamma)$ if $(p, \beta) \in \delta(q, a, Z)$. The reflexive transitive closure of $\vdash$ is denoted by $\vdash^*$. The language accepted by M by *accepting states* is the set $L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (f, \lambda, \gamma),\ \text{for some } f \in F \text{ and } \gamma \in \Gamma^*\}$.

The *pushdown store language* of M (see, e.g., [1, 4]) is the set $P(M)$ of all words occurring on the pushdown store along accepting computations of M:

$$P(M) = \{u \in \Gamma^* \mid \exists x, y \in \Sigma^*,\ q \in Q,\ f \in F : (q_0, xy, Z_0) \vdash^* (q, y, u) \vdash^* (f, \lambda, \gamma),\ \text{for some } \gamma \in \Gamma^*\}.$$

Throughout the rest of the paper, we assume PDA to be in *normal* form, i.e., they can push at most two symbols at each move.

3 Pushdown Store Languages: the General Case

Already in [4], it is proved that $P(M)$ is regular. Here, inspired by [1], we construct an optimal size NFA for $P(M)$ as:

Theorem 1. *Let $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ be a PDA. Then, $P(M)$ is accepted by an NFA with $|Q|^2(|\Gamma| + 1) + |Q|(2|\Gamma| + 3) + 2$ states. Moreover, there exist infinitely many PDA $M_{Q,\Gamma}$ such that every NFA accepting $P(M_{Q,\Gamma})$ needs $\Omega(|Q|^2|\Gamma|)$ states.*

[3] The empty word is here denoted by λ.

Proof. (outline, see Appendix) We define the set $Acc(Q)$ (resp., $Co\text{-}Acc(Q)$) representing all the pushdown contents reachable from the initial configuration (resp., from which a final state can be reached). We let $[Q] = \{[q] \mid q \in Q\}$, and define:

$$Acc(Q) = \{[q]u \in [Q]\Gamma^* \mid \exists x, y \in \Sigma^* : (q_0, xy, Z_0) \vdash^* (q, y, u)\},$$
$$Co\text{-}Acc(Q) = \{[q]u \in [Q]\Gamma^* \mid \exists y \in \Sigma^*,\ f \in F,\ u' \in \Gamma^* : (q, y, u) \vdash^* (f, \lambda, u')\}.$$

We get[4] $P(M) = [Q]^{-1}(Acc(Q) \cap Co\text{-}Acc(Q))$. A left-linear (resp., right-linear) grammar for $Acc(Q)$ (resp., $Co\text{-}Acc(Q)$) can be built, and turned into an equivalent NFA with $|Q| \cdot (|\Gamma| + 1) + 1$ (resp., $|Q| + 2$) states. From these two NFA, an NFA for $P(M)$ with $|Q|^2(|\Gamma| + 1) + |Q|(2|\Gamma| + 3) + 2$ states is built. □

4 Pushdown Store Languages for Special Cases

For restricted models of PDA, we are able to provide NFA for their pushdown store languages whose size is strictly below the general upper bound in Theorem 1. Namely, we focus on: PDA *which never pop* a symbol from the pushdown, *stateless* PDA (i.e., with a single state [6]), and *counter machines* (i.e., PDA with pushdown alphabets having a single symbol Z beside Z_0 [6]):

Theorem 2. *Let $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ be a PDA of type displayed in the first column of Table 1. Then, an NFA for $P(M)$ can be built, whose number of states is bounded as in the second column. These size bounds are optimal.*

PDA type	**Size of NFA for $P(M)$**
never popping	$\|Q\| \cdot \|\Gamma\| + 1$
stateless	$\|\Gamma\| + 1$
counter machine	$\|Q\| + 2$

Table 1. Size of NFA accepting pushdown store languages of restricted PDA.

Proof. (outline, see Appendix) For never popping PDA and stateless PDA, the grammars for $Acc(Q)$ and $Co\text{-}Acc(Q)$ in Theorem 1 can be "optimized". For counter machines, by pigeonhole arguments on possible pushdown contents, we prove that $P(M)$ can be only of the form Z^*Z_0 or Z^hZ_0, with $h \leq |Q|$. In all three cases, PDA witnessing optimality can be exhibited. □

5 Computational Complexity of Decidability Questions

The complexity of deciding some properties of $P(M)$ for a given PDA M, namely, finiteness and being subset of Z^*Z_0, can be answered by first constructing the NFA N for $P(M)$ and then deciding finiteness or inclusion in Z^*Z_0 for $L(N)$, respectively. For the first step, we get (see Appendix):

Theorem 3. *Let $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ be a PDA. Then, an NFA for $P(M)$ can be constructed in deterministic polynomial time.*

[4] Given $A, B \subseteq \Sigma^*$, we let $A^{-1}B = \{y \in \Sigma^* \mid \exists x \in A : xy \in B\}$.

This leads to P-completeness of the following decision problems:

Theorem 4. *Given a PDA M, it is* P*-complete to decide whether $P(M)$: (i) is a finite set, (ii) is a finite set of words having at most length k, for a given $k \geq 1$, (iii) is a subset of Z^*Z_0.*

Proof. (outline, see Appendix) We consider only point (i). The problem belongs to P: by Theorem 3, an NFA N for $P(M)$ is built in polynomial time. Then, the infiniteness of $L(N)$ can be decided in NLOGSPACE $\subseteq$ P [8], which is closed under complementation [7, 11]. Hence, the finiteness of $L(N)$ can be decided in NLOGSPACE as well. For completeness, we log-space reduce the emptiness problem for context-free grammars, which is known to be P-complete [9]. □

As a consequence, we get the P-completeness of deciding whether a PDA is of a certain "nature". More precisely, a PDA M is of *constant height* if there is a constant $k \geq 1$ such that, for any word in $L(M)$, there exists an accepting computation along which the pushdown store never contains more than k symbols [2, 3]. M is *essentially* a counter machine [6] if in all of its accepting computations the pushdown storage is used as a counter. By Theorem 4, we get

Corollary 5. *For an unambiguous PDA M, it is* P*-complete to decide whether M: (i) is of constant height, (ii) is of constant height k, for a given $k \geq 1$. If M is a PDA, it is* P*-complete to decide whether it is essentially a counter machine.*

References

1. Autebert, J.-M., Berstel, J., Boasson, L.: Context-free languages and pushdown automata. In: Handbook of Formal Languages, Vol. 1. pp. 111–174. Springer (1997)
2. Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: The size-cost of Boolean operations on constant height deterministic pushdown automata. In: DCFS 2011. LNCS 6808, pp. 80–92. Springer (2011)
3. Geffert, V., Mereghetti, C., Palano, B.: More concise representation of regular languages by automata and regular expressions. Inf. Comput. 208, 385–394 (2010)
4. Greibach, S.A.: A note on pushdown store automata and regular systems. Proc. Amer. Math. Soc. 18, 263–268 (1967)
5. Hartmanis, J.: Context-free languages and Turing machines computations. Proc. Symposium on Applied Mathematics 19, 42–51 (1967)
6. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, Massachusetts (1979)
7. Immerman, N.: Nondeterministic space is closed under complementation. SIAM J. Comput. 17, 935–38 (1988)
8. Jones, N.D.: Space-bounded reducibility among combinatorial questions. J. Comput. System. Sci. 11, 68–85 (1975)
9. Jones, N.D., Laaser, W.T.: Complete problems for deterministic polynomial time. Theoretical Computer Science 3, 105–118 (1976).
10. Mereghetti, C., Palano, B.: Quantum finite automata with control language. Theoretical Informatics and Applications 40, 315–332 (2006)
11. Szelepcsényi, R.: The method of forced enumeration for nondeterministic automata. Acta Inform. 26, 279–84 (1988)