ON STATELESS PUSHDOWN AUTOMATA AND LIMITED PUSHDOWN ALPHABETS

Lukáš Vrábel
Doctoral Degree Programme (3), FIT BUT
E-mail: xvrabe01@stud.fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

Abstract: As its name suggests, a stateless pushdown automaton has no states. As a result, each of its computational steps depends only on the currently scanned symbol and the current pushdown-store top. Recently, there has been an interest in the investigation of limited pushdown alphabets. An infinite hierarchy of languages has been established based on this limitation. The proof was based on the language with growing input alphabet. This result was then improved by showing that the binary alphabet is sufficient for deterministic stateless automata. In this paper, we consider general nondeterministic stateless pushdown automata. We generalize these recent results by establishing an infinite hierarchy of language families resulting from stateless pushdown automata with limited pushdown alphabets and binary input alphabets.

Keywords: stateless pushdown automata, limited pushdown alphabets, binary input alphabet, generative power, infinite hierarchy of language families

1 INTRODUCTION

A *stateless pushdown automaton* (see [5, 6, 13, 14]) is an ordinary pushdown automaton with only a single state. Consequently, the moves of a stateless pushdown automaton do not depend on internal states but solely on the symbols currently scanned by its head accessing the input tape and pushdown store. Recently, there has been a renewed interest in the investigation of various types of stateless automata. Namely, consider stateless restarting automata [8, 9], stateless multihead automata [4, 7], a relation of stateless automata to P systems [15], and stateless multicounter machines and Watson-Crick automata [1–3].

It has been also shown that limiting the pushdown alphabet of stateless pushdown automata results in infinite hierarchy of languages accepted by these automata (see [12]). However, the witness language for the n-th level of the hierarchy is over an input alphabet with 2(n-1) elements. This result was improved by showing that a binary input alphabet is sufficient to establish infinite hierarchy for deterministic stateless pushdown automata (see [10]). However, deterministic stateless pushdown automata are less powerfull than their nonodeterministic counterpart.

In this paper, we consider the impact of the size of pushdown alphabets to the power of general nondeterministic stateless pushdown automata with binary input alphabet. More specifically, we establish an infinite hierarchy of language families over a binary alphabet resulting from stateless pushdown automata with limited pushdown alphabets. For every positive integer n, we give a language over binary alphabet which can only be accepted by a stateless pushdown automaton with at least $n+1$ pushdown symbols.

The achieved results can be seen as a continuation of existing studies on infinite hierarchies resulting from limited resources of various types of stateless automata (see [1–4, 7, 10, 12]).

The paper is organized as follows. First, Section 2 gives all the necessary terminology. Then, Section 3 establishes the infinite hierarchies mentioned above. In the conclusion, Section 4 states an open problem related to the achieved results.

2 PRELIMINARIES AND DEFINITIONS

In this paper, we assume that the reader is familiar with the theory of formal languages (see [6]). For a set Q, $\mathrm{card}(Q)$ denotes the cardinality of Q. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $|w|$ denotes the length of w. For $w \in V^*$ and $a \in V$, $\#_a w$ denotes the number of occurrences of a in w.

Next, we define stateless pushdown automata. Since these automata have only a single state, for brevity, we define them without any states at all.

Definition 1 (see [13])**.** A *stateless pushdown automaton* (an *SPDA* for short) is a quadruple

$$M = \big(\Sigma, \Gamma, R, \alpha\big),$$

where Σ is an *input alphabet*, Γ is a *pushdown alphabet*, $R \subseteq \Gamma \times \Sigma \times \Gamma^*$ is a finite relation, called the set of *rules*, and $S \in \Gamma$ is the *initial pushdown symbol*. Instead of $(A, a, w) \in R$, we write $Aa \to w$ throughout the paper. For $r = (Aa \to w) \in R$, Aa and w represent the *left-hand side* of r and the *right-hand side* of r, respectively.

The *configuration* of M is any element of $\Gamma^* \times \Sigma^*$. For a configuration $c = (\pi, w)$, π is called the *pushdown* of c and w is called the *unread part of the input string* (or just *input string* for short) of c.

The *direct move relation* over the set of all configurations, symbolically denoted by $\vdash$, is a binary relation over the set of all configurations defined as follows: $(\pi A, au) \vdash (\pi w, u)$ in M if and only if $Aa \to w \in R$, where $\pi, w \in \Gamma^*$, $A \in \Gamma$, $a \in \Sigma$, and $u \in \Sigma^*$. Let $\vdash^k$, $\vdash^+$, and $\vdash^*$ denote the kth power of $\vdash$, for some $k \geq 1$, transitive closure of $\vdash$, and the reflexive-transitive closure of $\vdash$, respectively.

The *language accepted by M* is denoted by $L(M)$ and defined as

$$L\big(M\big) = \big\{w \in \Sigma^* \mid (S, w) \vdash^* (\varepsilon, \varepsilon)\big\}. \qquad \square$$

In some proofs, we will need to point out the fact that the part of the pushdown is not used in the acceptation of a string by an SPDA. In order to simplify these proofs, we will introduce the following notion.

Definition 2. Let $M = (\Sigma, \Gamma, R, S)$ be an SPDA and let

$$(\pi\pi_1, w_1) \vdash (\pi\pi_2, w_2) \vdash \cdots \vdash (\pi\pi_n, w_n)$$

where $\pi \in \Gamma^*$, $\pi_1 \ldots \pi_n \in \Gamma^*$, and $w_1 \ldots w_n \in \Sigma^*$. If $\pi_i \in \Gamma^+$ for all $1 \leq i \leq n$, then we say that π *is not used in* $(\pi\pi_1, w_1) \vdash^* (\pi\pi_n, w_n)$. $\square$

Furthermore, in order to show the infinite hierarchy, we will use the following language, L_n, defined over the binary alphabet $\{a, b\}$ as follows:

Definition 3. Let $n \geq 2$ be a positive integer, and consider the $(n+1)$-SPDA $M = \big(\Sigma, \Gamma, R, S\big)$, where

$$\begin{aligned} \Sigma &= \{a, b\} \\ \Gamma &= \{S\} \cup \{A_i \mid 1 \leq i \leq n\}, \\ R &= \textstyle\bigcup_{1 \leq i \leq n} \{Sb \to A_i^i, A_i a \to \varepsilon, A_i b \to A_i^{i+1}\}, \end{aligned}$$

with $S \notin \{A_i \mid 1 \leq i \leq n\}$. Let $L_n = L(M)$. $\square$

Note, that for each $w \in L_n$, there is such k that $\#_a(w) = k\#_b(w)$, where $1 \leq k \leq n$ is a natural number. This fact will be used in some of the proofs.

3 RESULTS

Following two lemmas will help us reduce the complexity of the proofs presented later. First lemma is the well-known pumping lemma, which illustrates the effect of finite resources on the accepted language.

Lemma 1. (The Pumping Lemma) *Let L be a regular language. Then, there exists a natural number k such that every word,* $z \in L$*, satisfying* $|z| \geq k$ *can be expressed as* $z = uvw$ *where* $v \neq \varepsilon$*,* $|uv| \leq k$*, and* $uv^m w \in L$ *for all* $m \geq 0$*.*

Proof. See [11], page 230. □

Next lemma illustrates the effect of the statelessness on the direct move relation. It shows that a sequence of moves is independent of the pushdown contents and part of input string not used during these moves.

Lemma 2. *Let* $M = (\Sigma, \Gamma, R, S)$ *be an SPDA. If* $(\pi_1, u_1) \vdash^* (\pi_2, u_2)$ *for some* $\pi_1, \pi_2 \in \Gamma^*$ *and* $u_1, u_2 \in \Sigma^*$*, then* $(\pi\pi_1, u_1 u) \vdash^* (\pi\pi_2, u_2 u)$ *for all* $\pi \in \Gamma^*$ *and* $u \in \Sigma^*$*.*

Proof. This lemma follows from the fact that the definition of $\vdash$ depends only on the topmost symbol of the pushdown and on the leftmost symbol of the input string. □

Notice that Lemma 2 implies that if $(\pi_1, u_1) \vdash^* (\varepsilon, \varepsilon)$, then $(\pi\pi_1, u_1 u) \vdash^* (\pi, u)$ for each $\pi \in \Gamma^*$ and $u \in \Sigma^*$. This implication is used throughout the rest of this paper.

Following lemma covers the most important result of this paper. It shows that we need at least $n+1$ pushdown symbols in order to accept the language L_n.

Lemma 3. $L_n \notin {}_n$**SPDA**

Proof. Let $M = (\Sigma, \Gamma, R, S)$ be an SPDA accepting L_n. We will show, that card$(\Gamma) \geq n+1$. First, Claim 1 will show that S can be used only in the beginning of an acceptation of any word. Then, Claim 2 will show that there have to be at least n additional distinct symbols in Γ.

Recall that for each $w \in L_n$, there is such k that $\#_a(w) = k\#_b(w)$, where $1 \leq k \leq n$ is a natural number. This fact will be used for proving both claims.

Claim 1. *Let* $M = (\Sigma, \Gamma, R, S)$ *be an SPDA accepting* L_n*. Then, S can occur on the pushdown only in the first configuration of any accepting move sequence.*

Proof. By contradiction. For the sake of contradiction, assume that there is such $x \in L_n$ where S occurs on the pushdown more than once during the acceptation of x. Let $x = uvw$, where $u, v, w \in \Sigma^*$, such that $(S, uvw) \vdash^+ (\alpha S, vw) \vdash^* (\alpha, w) \vdash^* (\varepsilon, \varepsilon)$, where $\alpha \in \Gamma^*$. Thus, $|u| \geq 1$ and $|v| \geq 1$.

As $|uw| \geq 1$, there is such $y \in L_n$ that the ratio of $\#_a(uyw)$ to $\#_b(uyw)$ is not a natural number.

As $y \in L_n$, $(S, y) \vdash^* (\varepsilon, \varepsilon)$. Then, by Lemma 2, $(S, uyw) \vdash^* (\alpha S, yw) \vdash^* (\alpha, w) \vdash^* (\varepsilon, \varepsilon)$. Therefore, $uyw \in L_n$. But as there is no such k that $\#_a(uyw) = k\#_b(uyw)$, $uyw \notin L_n$, which is a contradiction. Thus, the lemma holds. □

Now we will show, that there have to be unique $A_i \in \Gamma$ for each $1 \leq i \leq n$ such that $A_i \neq S$. In the proof, we will concentrate just on some of the strings contained in L_n. More precisely, we will use just the following strings defined over the natural number i:

$$L_n(i) = \{w \mid w = b(ba^i)^k a^i \text{ for some } k \geq 1\}.$$

Observe that $L_n(i) \subseteq L_n$ for each $1 \leq i \leq n$, so each SPDA accepting L_n have to accept all of the strings from $L_n(i)$. Furthermore, note that $L_n(i)$ is regular language and $\#_a(w) = i\#_b(w)$ for each $w \in L_n(i)$.

Claim 2. *There is distinct $A_i \in \Gamma$ for each $1 \leq i \leq n$ such that $A_i \neq S$.*

Proof. By contradiction. Let $\Gamma_{\not S} = \Gamma - \{S\}$. For the sake of contradiction, assume that $\mathrm{card}(\Gamma_{\not S}) < n$. Then there are sufficiently long $x_1 \in L_n(j_1)$ and $x_2 \in L_n(j_2)$, where $1 \leq j_1 < j_2 \leq n$, such that the same $A \in \Gamma_{\not S}$ is used in the acceptation of both x_1 and x_2. Furthermore, as $L_n(j_1)$ and $L_n(j_2)$ are regular languages, by Lemma 1, $x_1 = u_1w_1v_1$ and $x_2 = u_2w_2v_2$ such that $u_1, u_2, v_1, v_2, w_1, w_2 \in \Sigma^*$, $u_1w_1^kv_1 \in L_n(j_1)$, and $u_2w_2^kv_2 \in L_n(j_2)$ for every $k \geq 1$.

By Claim 1, S can occur only in the beginning of acceptation, and by contradiction assumption, $\mathrm{card}(\Gamma_{\not S}) < n$. Therefore, there has to be $A \in \Gamma_{\not S}$ such that

$$(S, u_1w_1^kv_1) \vdash^* (\alpha_1 A, w_1^kv_1) \vdash^* (\alpha_1, v_1) \vdash^* (\varepsilon, \varepsilon)$$

and

$$(S, u_2w_2^kv_2) \vdash^* (\alpha_2 A, w_2^kv_2) \vdash^* (\alpha_2, v_2) \vdash^* (\varepsilon, \varepsilon)$$

where $\alpha_1, \alpha_2 \in \Gamma^*$ are not used during the $(\alpha_1 A, w_1^kv_1) \vdash^* (\alpha_1, v_1)$ and $(\alpha_2 A, w_2^kv_2) \vdash^* (\alpha_2, v_2)$ respectively.

Then, as w_1 and w_2 can be iterated, $\#_a(w_1) = j_1\#_b(w_1)$ and $\#_a(w_2) = j_2\#_b(w_2)$. Furthermore, as each move removes one symbol from the input, $|u_1| \geq 1$ and $|u_2| \geq 1$. Therefore, $\#_a(u_1v_1) = j_1\#_b(u_1v_1)$ and $\#_a(u_2v_2) = j_2\#_b(u_2v_2)$.

Now we can construct such $x = u_2w_1v_2$ that would lead to contradiction. According to Lemma 2, there is

$$(S, u_2w_1^kv_2) \vdash^* (\alpha_2 A, w_1^kv_2) \vdash^* (\alpha_2, v_2) \vdash^* (\varepsilon, \varepsilon)$$

so $u_2w_1^kv_2 \in L_n$ for any $k \geq 1$. However, as $j_1 \neq j_2$, there is no $m \geq 1$ such that $\#_a(u_2w_1^kv_2) = m\#_b(u_2w_1^kv_2)$ for each $k \geq 1$, which is a contradiction to $u_2w_1^kv_2 \in L_n$. Thus, the claim holds. □

By Claim 2, $\mathrm{card}(\Gamma) \geq n+1$. Therefore, $L_n \notin {}_n\mathbf{SPDA}$, so the lemma holds. □

Based on these lemmas, we present the final result in the following theorem.

Theorem 1. ${}_n\mathbf{SPDA} \subset {}_{n+1}\mathbf{SPDA}$ *for each $n \geq 2$.*

Proof. By Lemma 3, $L_n \notin {}_n\mathbf{SPDA}$, and by Definition 3, $L_n \in {}_{n+1}\mathbf{SPDA}$. By Definition 1, ${}_n\mathbf{SPDA} \subseteq {}_{n+1}\mathbf{SPDA}$. Thus, the theorem holds. □

4 CONCLUSION

In this paper, we have shown an infinite hierarchy of languages over binary alphabet resulting from limiting the pushdown alphabet of stateless pushdown automata. However, the hierarchy is established on pushdown alphabets of sizes two and more. We conclude this paper by presenting an open problem: can this hierarchy be extended also to stateless pushdown automata with single pushdown symbol?

ACKNOWLEDGEMENT

This work was supported by the following grants: BUT FIT-S-11-2, MŠMT ED1.1.00/02.0070, and CEZ MŠMT MSM0021630528.

REFERENCES

[1] Ö. Eğecioğlu, L. Hegedüs, and B. Nagy. Hierarchies of stateless multicounter $5' \to 3'$ Watson-Crick automata languages. *Fundamenta Informaticae*, 110(1-4):111–123, 2011.

[2] Ö. Eğecioğlu, L. Hegedüs, and B. Nagy. Hierarchy results on stateless multicounter $5' \to 3'$ Watson-Crick automata. In *Advances in Computational Intelligence*, volume 6691 of *Lecture Notes in Computer Science*, pages 465–472. Springer, Berlin/Heidelberg, 2011.

[3] Ö. Eğecioğlu and O. H. Ibarra. On stateless multicounter machines. In *CiE'09: Proceedings of the 5th Conference on Computability in Europe: Mathematical Theory and Computational Practice*, pages 178–187, Heidelberg, DE, 2009.

[4] P. Frisco and O. H. Ibarra. On stateless multihead finite automata and multihead pushdown automata. In *Developments in Language Theory*, volume 5583 of *Lecture Notes in Computer Science*, pages 240–251. Springer, Berlin/Heidelberg, 2009.

[5] J. H. Gallier. DPDA's in 'atomic normal form' and applications to equivalence problems. *Theoretical Computer Science*, 14(2):155–186, 1981.

[6] M. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, Boston, 1978.

[7] O. H. Ibarra, J. Karhumäki, and A. Okhotin. On stateless multihead automata: Hierarchies and the emptiness problem. In *LATIN 2008: Theoretical Informatics*, volume 4957 of *Lecture Notes in Computer Science*, pages 94–105. Springer, Berlin/Heidelberg, 2008.

[8] M. Kutrib, H. Messerschmidt, and F. Otto. On stateless deterministic restarting automata. *Acta Informatica*, 47(7):391–412, 2010.

[9] M. Kutrib, H. Messerschmidt, and F. Otto. On stateless two-pushdown automata and restarting automata. *International Journal of Foundations of Computer Science*, 21(5):781–798, 2010.

[10] Tomás Masopust. A note on limited pushdown alphabets in stateless deterministic pushdown automata. *CoRR*, abs/1208.5002, 2012.

[11] A. Meduna. *Automata and Languages: Theory and Applications*. Springer, London, 2000.

[12] Alexander Meduna, Lukáš Vrábel, and Petr Zemek. An infinite hierarchy of language families resulting from stateless pushdown automata with limited pushdown alphabets. In *DCFS'12: 14th International Workshop on Descriptional Complexity of Formal Systems*, volume 2012 of *LNCS 7386*, pages 236–243. Springer Verlag, 2012.

[13] M. Oyamaguchi and N. Honda. The decidability of equivalence for deterministic stateless pushdown automata. *Information and Control*, 38(3):367–376, 1978.

[14] L. G. Valiant. Decision procedures for families of deterministic pushdown automata. Research Report CS-RR-001, Department of Computer Science, University of Warwick, Coventry, UK, 1973.

[15] L. Yang, Z. Dang, and O. H. Ibarra. On stateless automata and P systems. *International Journal of Foundations of Computer Science*, pages 1259–1276, 2008.