



HAL
open science

A polynomial procedure for trimming visibly pushdown automata

Mathieu Caralp, Pierre-Alain Reynier, Jean-Marc Talbot

► **To cite this version:**

Mathieu Caralp, Pierre-Alain Reynier, Jean-Marc Talbot. A polynomial procedure for trimming visibly pushdown automata. 2011. hal-00606778v1

HAL Id: hal-00606778

<https://hal.science/hal-00606778v1>

Submitted on 7 Jul 2011 (v1), last revised 29 Mar 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A polynomial procedure for trimming visibly pushdown automata

Mathieu Caralp, Pierre-Alain Reynier, and Jean-Marc Talbot

Laboratoire d'Informatique Fondamentale de Marseille
Aix-Marseille Université & CNRS, France

Abstract. We describe a polynomial procedure which, given a visibly pushdown automaton that accepts only well-nested words, returns an equivalent visibly pushdown automaton that is trimmed.

1 Introduction

Visibly pushdown automata (VPA) are a particular class of pushdown automata working over an alphabet splitted into call, internal and return symbols [1,2]. In VPA's, the stack behaviour is imposed by the input word: on a call symbol, the VPA pushes a symbol onto the stack, on a return symbol, it must pop the top symbol of the stack, and on an internal symbol, the stack remains unchanged.

Trimming a finite state automaton amounts to remove useless states, that is states that do not occur in some accepting computation of the automaton. This can be done easily in linear time simply by solving two reachability problems in the graph representing the automaton. However, the problem is more difficult for VPA's as the current state of a computation (called a configuration) is given by both a "control" state and a stack content.

To the best of our knowledge, the only trimming procedure for VPA can be deduced from the one on pushdown automata given in [3] yielding an exponential algorithm.

2 Definitions

Words and nested words Let Σ be a finite alphabet partitioned into three disjoint sets Σ_c , Σ_r and Σ_i , denoting respectively the *call*, *return* and *internal* alphabets. We denote by Σ^* the set of (finite) words over Σ and by ϵ the empty word. The length of a word u is denoted by $|u|$. The set of *well-nested words* Σ_{wn}^* is the smallest subset of Σ^* such that $\Sigma_i^* \subseteq \Sigma_{\text{wn}}^*$ and for all $c \in \Sigma_c$, all $r \in \Sigma_r$, all $u, v \in \Sigma_{\text{wn}}^*$, $cur \in \Sigma_{\text{wn}}^*$ and $wv \in \Sigma_{\text{wn}}^*$. We define the height $h(u)$ of some well-nested word u by induction as follows: $h(u) = 0$ if $u \in \Sigma_i^*$, $h(uv) = \max(h(u), h(v))$ for $u, v \in \Sigma_{\text{wn}}^*$, and $h(cur) = 1 + h(u)$. Given a family of words w_1, w_2, \dots, w_n , we denote by $\prod_{i=1}^n w_i$ the concatenation $w_1 w_2 \dots w_n$.

Visibly pushdown automata (VPA) Visibly pushdown automata are a restriction of pushdown automata in which the stack behaviour is imposed by the input word. On a call symbol, the VPA pushes a symbol onto the stack, on a return symbol, it must pop the top symbol of the stack, and on an internal symbol, the stack remains unchanged. Formally:

Definition 1 (Visibly pushdown automata). A *visibly pushdown automaton* (VPA) on finite words over Σ is a tuple $A = (Q, I, F, \Gamma, \delta)$ where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ the set of final states, Γ is a finite stack alphabet, $\delta = \delta_c \uplus \delta_r \uplus \delta_i$ the (finite) transition relation, with $\delta_c \subseteq Q \times \Sigma_c \times \Gamma \times Q$, $\delta_r \subseteq Q \times \Sigma_r \times \Gamma \times Q$, and $\delta_i \subseteq Q \times \Sigma_i \times Q$.

A *configuration* of a VPA is a pair $(q, \sigma) \in Q \times \Gamma^*$. A *run* of A on a word $u = a_1 \dots a_l \in \Sigma^*$ from a configuration (q, σ) to a configuration (q', σ') is a finite sequence of configurations $\rho = \{(q_k, \sigma_k)\}_{0 \leq k \leq l}$ such that $q_0 = q$, $\sigma_0 = \sigma$, $q_l = q'$, $\sigma_l = \sigma'$ and for each $1 \leq k \leq l$, there exists $\gamma_k \in \Gamma$ such that either $(q_{k-1}, a_k, \gamma_k, q_k) \in \delta_c$ and $\sigma_k = \sigma_{k-1} \gamma_k$ or $(q_{k-1}, a_k, \gamma_k, q_k) \in \delta_r$ and $\sigma_{k-1} = \sigma_k \gamma_k$, or $(q_{k-1}, a_k, q_k) \in \delta_i$ and $\sigma_k = \sigma_{k-1}$. We write $(q, \sigma) \xrightarrow{u} (q', \sigma')$ when there exists a run on u from (q, σ) to (q', σ') . We may omit the superscript u when irrelevant. We denote by \perp the empty word on Γ .

Initial (resp. final) configurations are configurations of the form (q, \perp) , with $q \in I$ (resp. $q \in F$). A configuration (q, σ) is *accessible* (resp. is *co-accessible*) if there exist $u \in \Sigma^*$ and a configuration c such that c is initial and $c \xrightarrow{u} (q, \sigma)$ (resp. such that c is final and $(q, \sigma) \xrightarrow{u} c$).

Definition 2. An automaton A is *trimmed* if every configuration of A is accessible iff it is co-accessible.

We say that a run is accepting if it starts in an initial configuration and ends in a final configuration. A word is accepted by A iff there exists an accepting run of A on this word. The language of A , denoted by $L(A)$, is the set of words accepted by A . Note that we require here to end up with an empty stack, while the definition of [1] considers acceptance by final states only. This restriction implies that all accepted words are well-nested.

3 Trimming VPA

Let $A = (Q, I, F, \Gamma, \delta)$ be a VPA on the structured alphabet Σ . In this section, we define a new VPA $A' = (Q', I', F', \Gamma', \delta')$ on Σ , denoted $\text{trim}(A)$, which recognizes the same language, and in addition is trimmed.

First, we define the following set:

$$\text{WN} = \{(p, q, r) \in Q^3 \mid \exists (p, \perp) \rightarrow^* (q, \perp) \rightarrow^* (r, \perp)\}$$

This set can be computed in polynomial time. More precisely, the following set can be computed first: $\text{WN}_2 = \{(p, q) \in Q^2 \mid \exists (p, \perp) \rightarrow^* (q, \perp)\}$.

WN_2 can be defined as the least set such that

- $\{(q, q) \mid q \in Q\} \subseteq \text{WN}_2$,
- if $(p, r) \in \text{WN}_2$ and $(r, q) \in \text{WN}_2$, then $(p, q) \in \text{WN}_2$
- if $(p, q) \in \text{WN}_2$, and $\exists(q, i, q') \in \delta_l$, then $(p, q') \in \text{WN}_2$
- if $(p, q) \in \text{WN}_2$ and $\exists(p', c, \gamma, p) \in \delta_c, (q, r, \gamma, q') \in \delta_r$, then $(p', q') \in \text{WN}_2$

Then, WN is obtained from WN_2 by the following property:

$$(p, q, r) \in \text{WN} \iff (p, q) \in \text{WN}_2 \wedge (q, r) \in \text{WN}_2$$

We now define the four first components of the trimmed VPA A' as follows:

- $Q' = \text{WN}$
- $I' = \text{WN} \cap (I \times I \times F)$
- $F' = \text{WN} \cap (I \times F \times F)$
- $\Gamma' = \Gamma \times Q \times Q$

Intuitively, the VPA A' simulates the VPA A as follows: if a run of A' goes through a state (p, q, r) with a stack σ' of height n , then the run of A' at this position mimicks a run of A whose current configuration is (q, σ) , with σ of height n , and such that the top symbol of σ has been pushed when reaching the state p , and will be popped when leaving the state r . Moreover, from p to r in the run of A , the height of the stack is always larger or equal to n .

It remains to define the last component δ' . We define it by its restrictions on call, return and internal symbols respectively (namely δ'_c, δ'_r and δ'_l).

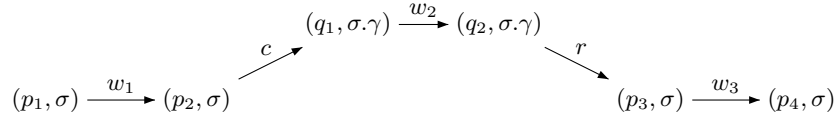
Call symbols. Let $(p_1, p_2, p_4) \in \text{WN}, c \in \Sigma_c$. Then $((p_1, p_2, p_4), c, (\gamma, p_1, p_4), (q_1, q_1, q_2)) \in \delta'_c$ iff the following three conditions hold:

- $(p_2, c, \gamma, q_1) \in \delta_c$,
- $(q_1, q_1, q_2) \in \text{WN}$,
- there exists a state p_3 such that $(q_2, r, \gamma, p_3) \in \delta_r$, and $(p_3, p_3, p_4) \in \text{WN}$.

Return symbols. Let $(q_1, q_2, q_2) \in \text{WN}, r \in \Sigma_r$. Then $((q_1, q_2, q_2), r, (\gamma, p_1, p_4), (p_1, p_3, p_4)) \in \delta'_r$ iff the following three conditions hold:

- $(q_2, r, \gamma, p_3) \in \delta_r$,
- $(p_1, p_3, p_4) \in \text{WN}$,
- there exists a state p_2 such that $(p_2, c, \gamma, q_1) \in \delta_c$, and $(p_1, p_2, p_2) \in \text{WN}$.

Schematically, for $w_1, w_2, w_3 \in \Sigma_{\text{wn}}^*, c \in \Sigma_c$, and $r \in \Sigma_r$, we have:



Internal symbols. Let $(p_1, q, p_2), (p_1, q', p_2) \in \text{WN}, a \in \Sigma_l$. Then $((p_1, q, p_2), a, (p_1, q', p_2)) \in \delta'_l$ iff the transition (q, a, q') belongs to δ_l .

Proposition 3. For any VPA A , $\text{trim}(A)$ can be computed in polynomial time.

This result easily follows from the definition of $\text{trim}(A)$.

Proposition 4. *For any VPA A , $L(A) = L(\text{trim}(A))$.*

To prove this result, we will first prove different lemmas.

Lemma 5. *Let A be a VPA and $A' = \text{trim}(A)$. Let ρ be a run of A' such that $\rho : ((p, q, r), \sigma) \xrightarrow{w}^* ((p', q', r'), \sigma)$, with $\sigma \in \Gamma'^*$ and $w \in \Sigma_{\text{wn}}^*$. Then we have $p = p'$ and $r = r'$.*

Proof. The proof goes by induction on the structure of the word w . It holds trivially when w is the empty word. Consider now a non-empty word w . There are two cases:

- First case: $w = aw'$, with $a \in \Sigma_\iota$, $w' \in \Sigma_{\text{wn}}^*$ and $((p_1, q_1, r_1), a, (p, q, r)) \in \delta'_\iota$. By construction of A' , $p_1 = p$ and $r_1 = r$. We conclude by using the induction hypothesis on w' .
- Second case: $w = cw'rw''$, with $c \in \Sigma_c$, $r \in \Sigma_r$, $w', w'' \in \Sigma_{\text{wn}}^*$. There exists a transition $((p_1, p_2, p_4), c, (\gamma, p_1, p_4), (q_1, q_1, q_2)) \in \delta'_c$ which is used when reading the first letter c . There exists a state (r_1, r_2, r_3) such that the run of A' on w' goes from state (q_1, q_1, q_2) to state (r_1, r_2, r_3) . As w' is well-nested, the induction hypothesis applied on w' entails that $r_1 = q_1$ and $r_3 = q_2$. In addition, as a return transition is used after reading w' , the definition of δ'_r implies that $r_2 = r_3$. Thus by definition of δ'_c and δ'_r , a transition of the form $((q_1, q_2, q_2), r, (\gamma, p_1, p_4), (p_1, p_3, p_4)) \in \delta'_r$ for some state p_3 is used when reading the letter r . We conclude by induction hypothesis applied on w'' . \square

We build a bijection between accepting runs of A and A' . First, we go from runs of A' to runs of A . We define the following standard projection mappings. Consider two integers $1 \leq j \leq k$. Given a tuple $s = (s_1, s_2, \dots, s_k)$, we denote by $\pi_j(s)$ the element s_j . In addition, we extend this mapping over words on tuples, by letting $\pi_j(\sigma) = \pi_j(\gamma_1)\pi_j(\gamma_2)\dots\pi_j(\gamma_n)$ where $\sigma = \gamma_1\gamma_2\dots\gamma_n$.

Lemma 6. *Let A be a VPA and $A' = \text{trim}(A)$. For any accepting run $\rho' = (q'_i, \sigma'_i)_{1 \leq i \leq k}$ of A' , $\rho = (\pi_2(q'_i), \pi_1(\sigma'_i))_{1 \leq i \leq k}$ is an accepting run of A .*

Proof. We prove by induction on ρ' that the projection run ρ defined above is a correct run of A . The fact that accepting runs of A' are projected on accepting runs of A is trivial. \square

Conversely, considering translation of runs of A to runs of A' . States of A' extend states of A by considering starting and ending states for the current stack level. Therefore we prove the following lemma in which we consider a word $w_2 \in \Sigma_{\text{wn}}^*$ embedded into a context $(w_1, w_3) \in \Sigma_{\text{wn}}^* \times \Sigma_{\text{wn}}^*$ such that $w_1w_2w_3$ corresponds to the current stack level.

Lemma 7. *Let A be a VPA and $A' = \text{trim}(A)$. For all $w_1, w_2, w_3 \in \Sigma_{\text{wn}}^*$, if there exists a run ρ of the form $(p, \perp) \xrightarrow{w_1}^* (q, \perp) \xrightarrow{w_2}^* (r, \perp) \xrightarrow{w_3}^* (s, \perp)$ in the VPA A , then there is a run ρ' of the form $((p, q, s), \perp) \xrightarrow{w_2}^* ((p, r, s), \perp)$ of A' .*

Proof. We prove this lemma by induction on the height of the word w_2 . If $h(w_2) = 0$, then $w_2 \in \Sigma_t^*$, assuming $w_2 = a_1 a_2 \dots a_n$, the restriction of ρ on w_2 is of the form

$$(q, \perp) \xrightarrow{a_1} (q_1, \perp) \dots \xrightarrow{a_n} (q_n, \perp) \text{ with } r = q_n.$$

Observe that for all i , $(p, q_i, s) \in \text{WN}$. By definition of A' the following is a run of A' on w_2 :

$$\rho' = ((p, q, s), \perp) \xrightarrow{a_1} ((p, q_1, s), \perp) \dots \xrightarrow{a_n} ((p, q_n, s), \perp) \text{ with } r = q_n.$$

We now assume for the induction that the property holds when $h(w_2) \leq n$ and consider w_2 such that $h(w_2) = n + 1$. There exists a unique decomposition of w_2 as follows:

$$w_2 = [\prod_{i=1}^k (w_i^{\text{int}} c_i w_i^{\text{wn}} r_i)] w_{k+1}^{\text{int}}$$

with for all i , $w_i^{\text{int}} \in \Sigma_t^*$, $w_i^{\text{wn}} \in \Sigma_{wn}^*$, $c_i \in \Sigma_c$, and $r_i \in \Sigma_r$.

Let $w_1, w_3 \in \Sigma_{wn}^*$ and ρ be a run of A on $w_1 w_2 w_3$. We decompose the run ρ as follows:

- on w_1 : $(p, \perp) \xrightarrow{w_1}^* (p_1^1, \perp)$
- on each $w_i^{\text{int}} c_i w_i^{\text{wn}} r_i$:

$$(p_i^1, \perp) \xrightarrow{w_i^{\text{int}}}^* (p_i^2, \perp) \xrightarrow{c_i} (q_i^1, \gamma_i) \xrightarrow{w_i^{\text{wn}}}^* (q_i^2, \gamma_i) \xrightarrow{r_i} (p_{i+1}^1, \perp)$$

- on w_{k+1}^{int} : $(p_{k+1}^1, \perp) \xrightarrow{w_{k+1}^{\text{int}}}^* (p_{k+1}^2, \perp)$
- on w_3 : $(p_{k+1}^2, \perp) \xrightarrow{w_3}^* (s, \perp)$

Note that $q = p_1^1$ and $r = p_{k+1}^2$.

As w_i^{wn} is well-nested, there exists a run in A of the form $(q_i^1, \perp) \xrightarrow{w_i^{\text{wn}}}^* (q_i^2, \perp)$.

By induction hypothesis, this implies that there exists a run $\rho'_i = ((q_i^1, q_i^1, q_i^2), \perp) \xrightarrow{w_i^{\text{wn}}}^* ((q_i^1, q_i^2, q_i^2), \perp)$ of A' on w_i^{wn} . Again, using the fact that w_i^{wn} is well-nested, we have $((q_i^1, q_i^1, q_i^2), \gamma'_i) \xrightarrow{w_i^{\text{wn}}}^* ((q_i^1, q_i^2, q_i^2), \gamma'_i)$ in A' for any $\gamma'_i \in \Gamma'$.

We now describe the other parts of the run ρ' of A' on w_2 .

Internal actions For all $i \in \{1, \dots, k+1\}$, $((p, p_i^1, s), \perp) \xrightarrow{w_i^{\text{int}}}^* ((p, p_i^2, s), \perp)$ on w_i^{int} , by the base induction with w_1 and w_3 well defined.

Calls For all $i \in \{1, \dots, k\}$, we have $(p_i^2, c_i, \gamma_i, q_i^1) \in \delta_c$ in A . Thus by definition of A' , there exists a transition $((p, p_i^2, s), c_i, (\gamma_i, p, s), (q_i^1, q_i^1, q_i^2)) \in \delta'_c$ in A' .

Returns For all $i \in \{1, \dots, k\}$, we have $(q_i^2, r_i, \gamma_i, p_{i+1}^1) \in \delta_r$ in A . Thus by definition of A' there exists a transition $((q_i^1, q_i^2, q_i^2), r_i, (\gamma_i, p, s), (p, p_{i+1}^1, s)) \in \delta'_r$ in A' .

One can then easily check that all the above transitions and partial runs can be gathered to obtain a run of A' on w_2 with the expected form. \square

To conclude the proof of Proposition 4, note that Lemma 6 implies that $L(\text{trim}(A)) \subseteq L(A)$. Conversely, using Lemma 7 for $w_1 = w_3 = \epsilon$ yields $p = q$ and $r = s$ and if moreover p is initial and q is final then any accepting run in A is associated with an accepting run in $\text{trim}(A)$. Hence, $L(A) \subseteq L(\text{trim}(A))$.

Proposition 8. *For any VPA A , $\text{trim}(A)$ is trimmed.*

Proof. Let A' be $\text{trim}(A)$. Therefore, we first prove that any accessible configuration of A' is also co-accessible, and then the converse.

Let $co = ((p, q, r), \sigma)$ be an accessible configuration of A' . There exists a run ρ of A' of the form $((i, i, f), \perp) \rightarrow^* co$ between an initial configuration $((i, i, f), \perp)$ and co . We show by induction on the height of the stack that we can reach a final configuration from co .

If $|\sigma| = 0$, by the lemma 5, we obtain $p = i$ and $r = f$. In particular, this implies $p \in I$ and $r \in F$. Since $(p, q, r) \in \text{WN}$, there is a run $(p, \perp) \rightarrow^* (q, \perp) \rightarrow^* (r, \perp)$ in A , thus by Lemma 7 we have a run $((p, q, r), \perp) \rightarrow^* ((p, r, r), \perp)$ of A' . This concludes this case as by the above observation, we have $(p, r, r) \in I \times F \times F = F'$.

We now assume for the induction that the property holds when $|\sigma| \leq n$ and we consider a stack σ such that $|\sigma| = n + 1$. Let denote by (γ, p', r') the top symbol of σ , write $\sigma = \sigma'.(\gamma, p', r')$, and consider the first position in the run ρ that pushes this symbol onto the stack. We denote by c the associated call. More precisely, there exists a unique decomposition of ρ as follows:

$$((i, i, f), \perp) \rightarrow^* ((p', q', r'), \sigma') \xrightarrow{c} ((p'', p'', r''), \sigma) \rightarrow^* ((p, q, r), \sigma)$$

such that the run from $((p'', p'', r''), \sigma)$ to $((p, q, r), \sigma)$ is associated with a well-nested word. By Lemma 5, we obtain $p'' = p$ and $r'' = r$. Considering the call transition associated with c , and by definition of δ'_c and δ'_r , there exists a return transition $((p, r, r), r, (\gamma, p', r'), (p', q'', r')) \in \delta'_r$ for some letter $r \in \Sigma_r$. In addition, as $(p, q, r) \in \text{WN}$, there is a run $(p, \sigma) \rightarrow^* (q, \sigma) \rightarrow^* (r, \sigma)$ in A , and thus by Lemma 7 we have a run $((p, q, r), \sigma) \rightarrow^* ((p, r, r), \sigma)$. As the top symbol of σ is (γ, p', r') , the above return transition can be used to reach configuration $((p', q'', r'), \sigma')$ whose height is n . The result follows by induction hypothesis.

Conversely, let $co = ((p, q, r), \sigma)$ be a co-accessible configuration of A' . There exists a run ρ of A' of the form $co \rightarrow^* ((i, f, f), \perp)$ between co and a final configuration $((i, f, f), \perp)$. We show by induction on the height of the stack that co can be reached from an initial configuration.

If $|\sigma| = 0$, by the lemma 5, we obtain $p = i$ and $r = f$. In particular, this implies $p \in I$ and $r \in F$. Since $(p, q, r) \in \text{WN}$, there is a run $(p, \perp) \rightarrow^* (q, \perp) \rightarrow^* (r, \perp)$ in A , thus by Lemma 7 we have a run $((p, p, r), \perp) \rightarrow^* ((p, q, r), \perp)$ of A' . This concludes this case as by the above observation, we have $(p, p, r) \in I \times I \times F = I'$.

We now assume for the induction that the property holds when $|\sigma| \leq n$ and we consider a stack σ such that $|\sigma| = n + 1$. Let denote by (γ, p', r') the top symbol of σ , write $\sigma = \sigma'.(\gamma, p', r')$, and consider the first position in the run ρ that pops this symbol from the stack. We denote by r the associated return. More precisely, there

exists a unique decomposition of ρ as follows:

$$((p, q, r), \sigma) \rightarrow^* ((p'', r'', r''), \sigma) \xrightarrow{r} ((p', q', r'), \sigma') \rightarrow^* ((i, f, f), \perp)$$

such that the run from $((p, q, r), \sigma)$ to $((p'', r'', r''), \sigma)$ is associated with a well-nested word. By Lemma 5, we obtain $p'' = p$ and $r'' = r$. Considering the return transition associated with r , and by definition of δ'_c and δ'_r , there exists a call transition $((p', q'', r'), c, (\gamma, p', r'), (p, p, r)) \in \delta'_c$ for some letter $c \in \Sigma_c$. In addition, as $(p, q, r) \in \text{WN}$, there is a run $(p, \sigma) \rightarrow^* (q, \sigma) \rightarrow^* (r, \sigma)$ in A , and thus by Lemma 7 we have a run $((p, p, r), \sigma) \rightarrow^* ((p, q, r), \sigma)$. As the top symbol of σ is (γ, p', r') , the above call transition can be used. As a consequence, we have proven that there exists a run from configuration $((p', q'', r'), \sigma')$ to configuration co . In particular, configuration $((p', q'', r'), \sigma')$ is a co-accessible configuration whose height is n , and the result follows by induction hypothesis. \square

To summarize the results presented in this paper, we have the following theorem:

Theorem 9. *Let A be a VPA. Then:*

- $\text{trim}(A)$ can be built in polynomial time
- $L(A) = L(\text{trim}(A))$
- $\text{trim}(A)$ is trimmed

Note that by definition of trimmed VPA, some useless state may remain in the construction producing $\text{trim}(A)$: indeed, $\text{trim}(A)$ may contain some state q such that for no stack σ , (q, σ) is accessible (and thus co-accessible). However, deciding whether for some stack σ , (q, σ) is accessible can be reduced to the emptiness problem of pushdown automata accepting on final states. This problem is decidable in polynomial time.

References

1. R. Alur and P. Madhusudan. Visibly pushdown languages. In *STOC*, pages 202–211, 2004.
2. R. Alur and P. Madhusudan. Adding nesting structure to words. *JACM*, 56(3):1–43, 2009.
3. D. Girault-Beauquier. Some results about finite and infinite behaviours of a pushdown automaton. In *Automata, Languages and Programming*, volume 172 of *Lecture Notes in Computer Science*, pages 187–195. Springer, 1984.