A lower bound for the coverability problem in acyclic pushdown VAS☆

Matthias Englert[b], Piotr Hofman[a], Sławomir Lasota[a], Ranko Lazić[b], Jérôme Leroux[c], Juliusz Straszyński[a]

[a] *University of Warsaw*
[b] *University of Warwick*
[c] *University of Bordeaux*

Abstract

We investigate the coverability problem for a one-dimensional restriction of pushdown vector addition systems with states. We improve the lower complexity bound to PSPACE, even in the acyclic case.

Keywords: context-free grammars, pushdown vector addition systems, one-counter machines, coverability problem, reachability problem

1. Preliminaries

This paper is on extension of the classical model of vector addition system (VAS) by a pushdown store. For convenience, we prefer to work with an equivalent model of *grammar-controlled VAS* (GVAS) [1], i.e., a VAS whose transitions are controlled by a context-free grammar. We restrict our attention to one-dimensional GVAS, referred to as 1*GVAS*, which is just a context free grammar whose terminal symbols are a finite subset of integers[1]. As an example, consider the following 1GVAS with one nonterminal **S** and two terminals $\{-1, 1\}$:

$$\mathbf{S} \;\rightarrow\; 1\,\mathbf{S}\,-1 \mid -1\,\mathbf{S}\,1 \mid \varepsilon \tag{1}$$

which generates, as a context-free grammar, *anti*palindromes over $\{-1, 1\}$ of even length.

☆This work has been partially supported by the NCN grant 2017/27/B/ST6/02093.

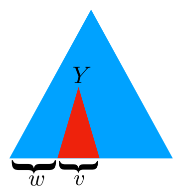
[1]In case of d-dimensional GVAS, the terminal symbols would be a finite subset of $\mathbb{Z}^d$.

For a word over terminal symbols $w = a_1 \dots a_k$ we write $\sum w$ to denote the sum $\sum_{j=1}^{k} a_j$ of its letters.

A word $w = a_1 \dots a_k$ is called *admissible* if the sum of every prefix of w is nonnegative: $\sum a_1 \dots a_i \geq 0$ for every $i = 1 \dots k$. The sums $\sum a_1 \dots a_i$ are called *prefix sums* in the sequel. For instance, an antipalidrome generated by the grammar (1) is admissible if the sum of every prefix of its first half is nonnegative (then the sum of every suffix of its second half is forcedly nonnegative too, and moreover the total sum is necessarily 0).

A derivation of an admissible word from the starting nonterminal is called admissible too. The *covering problem* is to decide, given a 1GVAS with terminals encoded in binary, whether it has an admissible derivation.

The prefix sums allow us to speak of input and output value of every subtree of a derivation tree, or even of every infix of the derived word. Indeed, consider a derivation of an admissible word wvu of terminals, where the infix v is derived from a nonterminal $\mathbf{Y}$:



The prefix sum $\sum w$ can be called the *input* of the subtree derived from the symbol Y, and the prefix sum $\sum wv = \sum w + \sum v$ can be called the *output* of that subtree. Note that the input of the whole derivation tree is 0, and the output is non-negative.

As context-free grammars are essentially stateless pushdown automata, we could consider equivalently the model of one-dimensional pushdown vector addition systems (1PVAS), i.e., 1VAS (one-dimensional VAS) extended with a pushdown store. Furthermore, this model is expressively equivalent to pushdown automata extended with a counter which can be incremented and decremented by every transition but cannot be tested for 0, and which is not allowed to drop below 0 during a run (the counter values along a run correspond to prefix sums). This latter model is known as one-dimensional pushdown vector addition systems with states (1PVASS). In this setting the coverability problem is equivalently rephrased as follows: given a counter-extented pushdown automaton, decide whether it has a run starting with the empty stack and counter value 0, and ending with the empty stack (and arbitrary non-negative counter value). The problem remains equivalent if the stack emptiness at the end of a run is dropped, but the run is required to end in an accepting control state. The mentioned equivalence of 1GVAS, 1PVAS and 1PVASS extends to higher dimensions.

The complexity of the problem does not change if unary encoding of terminals is assumed instead of binary one:

Proposition 1. *At the cost of a logarithmic-space reduction, we may assume that the terminals in the given* $1GVAS$ *are from* $\{-1, 1\}$.

Proof. Let $t \notin \{-1, 0, 1\}$ be a terminal of absolute value $|t| > 1$. Let $b_\ell \ldots b_1$ be the binary representation (where b_ℓ is the most significant bit) of $|t|$ if $t \geq 0$, and symmetrically let $-b_\ell \ldots -b_1$ be the binary representation of $|t|$ if $t < 0$. Thus $b_i \in \{-1, 0, 1\}$ for every $i \leq \ell$. We add ℓ new nonterminal symbols $\mathbf{X}_t^1, \ldots, \mathbf{X}_t^\ell$ to our grammar as well as the rule $\mathbf{X}_t^\ell \to b_\ell$ and, for each $i \in \{2, \ldots, \ell\}$, a rule $\mathbf{X}_t^{i-1} \to b_{i-1}\mathbf{X}_t^i\mathbf{X}_t^i$. This way, $\mathbf{X}_t^1$ can only derive a single word w of terminals such that $\sum w$ is exactly the value of t. We can therefore replace every occurrence of the terminal t by $\mathbf{X}_t^1$.

The construction, applied to all terminals t with $|t| > 1$, yields a grammar with terminals in $\{-1, 0, 1\}$. Finally, all appearances of the terminal 0 can be safely removed. □

The result. Prior to this work, the best upper complexity bound for the coverability problem in 1GVAS was ExpSpace and the best lower complexity bound was NP, both due to [1]. Our contribution is a simple proof of PSpace-hardness of the problem, even in the case of *acyclic* 1GVAS. In consequence, we obtain PSpace-completeness in case of acyclic 1GVAS.

Related work. We remark that the coverabilty problem easily reduces to the *reachability* problem, in which we seek a derivation of an admissible word w with zero output: $\sum w = 0$. For the latter problem, decidability remains open even in 1GVAS and we know no better lower bound than the one for coverability. It is thus feasible (and believable) that both the problems are PSpace-complete in 1GVAS. For 1GVAS extended with resets of the counter the coverability problem becomes undecidable, as recently shown in [2].

Nothing is known on GVAS in arbitrary dimension except for a non-elementary lower bound for the coverability problem shown in [3], and decidability of the termination and boundedness problems [4]. The latter problems are known to be decidable in exponential time in dimension 1, as shown in [5]. In arbitrary dimension reachability reduces (and is thus equivalent) to the coverability problem, due to a simple logarithmic-space reduction that increases dimension by 1. Hence the lower bound of [3] is subsumed by a recent non-elementary lower bound for the reachability problem in VASS [6].

2. Lower Bound for acyclic 1GVAS

By a *cycle* of G we mean a derivation of a word $w\mathbf{Y}w'$ from a nonterminal $\mathbf{Y}$, for some words w, w' over terminals. The lower bound shown in this section applies even to *acyclic* 1GVAS, i.e., ones without cycles. Its proof is based on the masters thesis of the last author [7].

Theorem 1. *For acyclic* 1*GVAS, the coverability problem is* PSPACE*-hard.*

Proof. We reduce from the alternating subset sum problem [8]: given non-negative integers a_1, a'_1, e_1, e'_1, …, a_k, a'_k, e_k, e'_k and s, all encoded in binary, to decide

$$\forall x_1 \in \{a_1, a'_1\}\ \exists y_1 \in \{e_1, e'_1\} \quad \cdots \quad \forall x_k \in \{a_k, a'_k\}\ \exists y_k \in \{e_k, e'_k\} \\ x_1 + y_1 + \cdots + x_k + y_k = s\ .$$

Equivalently: in a k-round game between two players, the universal and the existential one (where in every ith round the former player chooses a number $x_i \in \{a_i, a'_i\}$ and then the latter one chooses a number $y_i \in \{e_i, e'_i\}$), decide whether the existential player has a strategy to enforce the *end sum* $x_1 + y_1 + \cdots + x_k + y_k$ to be equal to s. This problem is PSPACE-complete [8].

Construction of a 1*GVAS.* Given an instance of the alternating subset sum problem, we produce an acyclic 1GVAS whose derivations are essentially existential player's strategy trees. Formally, an admissible derivation will correspond to an existential player's strategy such that:

1. the end sum of a play against every universal player's counter-strategy is *at least* s;
2. the cumulative sum of end sums of all plays against all universal player's counter-strategies is *at most* $2^k s$.

As there are exactly 2^k counter-strategies, an existential strategy verifying conditions (1) and (2) enforces the end sum to be equal to s, irrespectively of universal player's counter-strategy.

Admissible derivations of the following acyclic 1GVAS G_1 with $2k+1$ nonterminals ($\mathbf{A}_1$ is the starting one) correspond to existential player's strategies verifying condition (1):

$$\begin{aligned}
\mathbf{A}_i &\to a_i\ \mathbf{E}_i\ (-a_i + a'_i)\ \mathbf{E}_i\ (-a'_i) && (i = 1, \ldots, k)\\
\mathbf{E}_i &\to e_i\ \mathbf{A}_{i+1}\ (-e_i)\ \mid\ e'_i\ \mathbf{A}_{i+1}\ (-e'_i) && (i = 1, \ldots, k)\\
\mathbf{A}_{k+1} &\to (-s)\ s
\end{aligned}$$

(Nonterminals $\mathbf{A}_i$ correspond to moves of the universal player, while nonterminals $\mathbf{E}_i$ correspond to the responses of the existential one.) Indeed, the input to every subtree derived from $\mathbf{A}_{k+1}$ corresponds to the end sum of a play, and every admissible derivation enumerates all plays of some fixed strategy of the existential player.

Similarly, admissible derivations of the following acyclic 1GVAS G_2 ($\mathbf{X}$ is the starting nonterminal) correspond to strategies of the existential player verifying condition (2):

$$\begin{aligned}
\mathbf{X} &\to (2^k s)\ \mathbf{A}_1\\
\mathbf{A}_i &\to (-2^{k-i} a_i)\ \mathbf{E}_i\ (-2^{k-i} a'_i)\ \mathbf{E}_i && (i = 1, \ldots, k)\\
\mathbf{E}_i &\to (-2^{k-i} e_i)\ \mathbf{A}_{i+1}\ \mid\ (-2^{k-i} e'_i)\ \mathbf{A}_{i+1} && (i = 1, \ldots, k)\\
\mathbf{A}_{k+1} &\to \varepsilon
\end{aligned}$$

Indeed, the initial credit $2^k s$ is decremented by integers chosen by players in all plays of some fixed existential player's strategy, and the multiplicity 2^{k-i} of every integer chosen in ith round corresponds to the number of plays this integer takes part in.

Crucially, if we ignore the terminals appearing in the rules, the two grammars are (almost) the same. Moreover, the prefix sums in G_2 are bounded by the initial credit $2^k s$. Therefore, we are able to combine G_1 with G_2, if we multiply all integers appearing in the first one by $S := 2^k s + 1$. Intuitively, a prefix sum of G_1 becomes a more significant digit, and a prefix sum of G_2 becomes a less significant one, of a 2-digit number in the base S. This results in the following acyclic 1GVAS G with $2k+2$ nonterminals ($\mathbf{X}$ is the starting one), where i ranges over $1, \ldots, k$ as before:

$$\begin{aligned}
\mathbf{X} &\rightarrow (2^k s)\ \mathbf{A}_1 \\
\mathbf{A}_i &\rightarrow (Sa_i - 2^{k-i}a_i)\ \mathbf{E}_i\ (S(-a_i + a'_i) - 2^{k-i}a'_i)\ \mathbf{E}_i\ (-Sa'_i) \\
\mathbf{E}_i &\rightarrow (Se_i - 2^{k-i}e_i)\ \mathbf{A}_{i+1}(-Se_i)\ \mid\ (Se'_i - 2^{k-i}e'_i)\ \mathbf{A}_{i+1}(-Se'_i) \\
\mathbf{A}_{k+1} &\rightarrow (-Ss)\ (Ss)
\end{aligned}$$

Example. As a simple illustrating example, consider the following (positive) instance of the alternating subset sum problem: $k = 1,\ a_1 = 3,\ a'_1 = 5,\ e_1 = 4,\ e'_1 = 6$ and $s = 9$. Here are the corresponding grammars G_1 (one the left) and G_2 (on the right):

$$\begin{aligned}
\mathbf{A}_1 &\rightarrow 3\ \mathbf{E}_1\ 2\ \mathbf{E}_1\ -5 \\
\mathbf{E}_1 &\rightarrow 4\ \mathbf{A}_2\ -4\ \mid\ 6\ \mathbf{A}_2\ -6 \\
\mathbf{A}_2 &\rightarrow -9\ 9
\end{aligned}
\qquad
\begin{aligned}
\mathbf{X} &\rightarrow 18\ \mathbf{A}_1 \\
\mathbf{A}_1 &\rightarrow -3\ \mathbf{E}_1\ -5\ \mathbf{E}_1 \\
\mathbf{E}_1 &\rightarrow -4\ \mathbf{A}_2\ \mid -6\ \mathbf{A}_2 \\
\mathbf{A}_2 &\rightarrow \varepsilon
\end{aligned}$$

The terminals are coloured to depict the way of obtaining the GVAS G as a combination of G_1 and G_2 ($S = 2{\cdot}9 + 1 = 19$):

$$\begin{aligned}
\mathbf{X} &\rightarrow 18\ \mathbf{A}_1 \\
\mathbf{A}_1 &\rightarrow (19{\cdot}3\ -3)\ \mathbf{E}_i\ (19{\cdot}2\ -5)\ \mathbf{E}_i\ (-19{\cdot}5) \\
\mathbf{E}_1 &\rightarrow (19{\cdot}4\ -4)\ \mathbf{A}_2\ (-19{\cdot}4)\ \mid\ (19{\cdot}6\ -6)\ \mathbf{A}_2(-19{\cdot}6) \\
\mathbf{A}_2 &\rightarrow (-19 \cdot 9)\ (19 \cdot 9)
\end{aligned}$$

The existential player wins by answering 3 by 6, and 5 by 4, and hence this strategy satisfies conditions (1) and (2). Consequently, both G_1 and G_2 have admissible derivations, of words $3\ 6\ (-9)\ 9\ (-6)\ 2\ 4\ (-9)\ 9\ (-4)\,(-5)$ and $18\ (-3)\ (-6)\ (-5)\ (-4)$, respectively, corresponding to the winning strategy. The two derivations have both output 0, and can be combined into an admissible derivation in G of a word $18\ (19{\cdot}3{-}3)\ (19{\cdot}6{-}6)\ ({-}19{\cdot}9)\ 19{\cdot}9\ ({-}19{\cdot}6)\ (19{\cdot}2{-}5)\ (19{\cdot}4{-}4)\ ({-}19{\cdot}9)\ 19{\cdot}9\ ({-}19{\cdot}4)\,({-}19{\cdot}5)$. Note that G_1 and G_2 have other admissible derivations, e.g., of $3\ 6\ (-9)\ 9\ (-6)\ 2\ 6\ (-9)\ 9\ (-6)\,(-5)$ and $18\ (-3)\ (-4)\ (-5)\ (-4)$, respectively, with positive outputs, which however can not be combined into a a single admissible derivation of G.

Correctness. Derivations in G are in one-to-one correspondence with existential player's strategies. We need to argue that existential player has a winning strategy if, and only if the corresponding derivation of G is admissible. In one direction, we observe that whenever an existential player's strategy enforces the end sum to be equal to s, in the corresponding derivation the input to every subtree derived from $\mathbf{A}_{k+1}$ equals $Ss + o$ for some $0 \leq o \leq 2^k s = S - 1$, and hence the derivation is forcedly admissible. For the other direction, supposing the 1GVAS has an admissible derivation, we argue that the corresponding existential strategy verifies conditions (1) and (2) and hence enforces the end sum equal to s. Condition (1) follows similarly as for G_1: the input to every subtree derived from $\mathbf{A}_{k+1}$ is, on one hand, at most $Sr + 2^k s < S(r+1)$, where r is the end sum of the corresponding play, and on the other hand at least Ss; the two inequalities yield

$$S(r+1) > Ss$$

which implies $r \geq s$. For condition (2) observe, similarly as in case of G_1, that the sum of all S-multiplied integers in the derivation equals 0. Therefore, as the output value is nonnegative, the initial credit $2^k s$ is decreased by at most this value, and hence the cumulative sum of all end sums is at most $2^k s$ as required.

The reduction can be performed in polynomial time (in particular all numbers appearing as terminals in the grammar are of polynomial bit-size), which proves PSPACE-hardness of the coverability problem for acyclic 1GVAS. □

Remark 1. *As the proof of Proposition 1 preserves acyclicity, the lower bound applies to acyclic* 1*GVAS with terminals from* $\{-1, 1\}$.

The coverability problem is easily shown to be decidable in polynomial space for acyclic GVAS in *arbitrary* dimension; in consequence, of Theorem 1 the problem is therefore robustly PSPACE-complete for acyclic GVAS, also in dimension 1, no matter what encoding of numbers is chosen:

Proposition 2. *The coverability problem is* PSPACE*-complete, both for acyclic GVAS and acyclic* 1*GVAS, under both unary and binary encoding of numbers.*

Proof sketch. We only show the upper bound, for the lower bound relying on Proposition 1 and Theorem 1. Given an acyclic GVAS G with numbers in binary, a nondeterministic PSPACE procedure performs a left-to-right traversal of a derivation tree which is guessed on the fly. The depth of the tree is bounded, due to acyclicity of G, by the number n of nonterminals, therefore the left-to-right traversal is doable using a stack of depth n. The space used by the procedure is thus polynomial (linear in fact) in the size of G. □

We conclude with the remark concerning pushdown VAS and VASS:

Remark 2. *The* PSPACE*-completeness applies also to the two equivalent models, namely PVAS and PVASS, and for their one-dimensional subclasses* 1*PVAS and* 1*PVASS, under a suitably translated acyclicity assumption.*

Acknowledgements. We thank Wojtek Czerwiński, Filip Mazowiecki and Radek Piórkowski for the collaborative work on the upper complexity bound for the coverability problem. We are also grateful to Alain Finkel for promoting the study of one-dimensional Petri nets with a pushdown stack. Finally, we are grateful to anonymous reviewers for valuable and helpful comments.

[1] J. Leroux, G. Sutre, P. Totzke, On the Coverability Problem for Pushdown Vector Addition Systems in One Dimension, in: ICALP, Part II, vol. 9135 of *LNCS*, Springer, 324–336, 2015.

[2] S. Schmitz, G. Zetzsche, Coverability Is Undecidable in One-Dimensional Pushdown Vector Addition Systems with Resets, in: Proc. Reachability Problems 2019, vol. 11674 of *LNCS*, Springer, 193–201, 2019.

[3] R. Lazic, P. Totzke, What Makes Petri Nets Harder to Verify: Stack or Data?, in: Concurrency, Security, and Puzzles - Essays Dedicated to Andrew William Roscoe on the Occasion of His 60th Birthday, 144–161, 2017.

[4] J. Leroux, M. Praveen, G. Sutre, Hyper-Ackermannian bounds for pushdown vector addition systems, in: Proc. CSL-LICS'14, 63:1–63:10, 2014.

[5] J. Leroux, G. Sutre, P. Totzke, On Boundedness Problems for Pushdown Vector Addition Systems, in: Proc. Reachability Problems 2015, 101–113, 2015.

[6] W. Czerwiński, S. Lasota, R. Lazic, J. Leroux, F. Mazowiecki, The reachability problem for Petri nets is not elementary, in: Proc. STOC 2019, ACM, 24–33, 2019.

[7] J. Straszyński, Complexity of the reachability problem for pushdown Petri nets, Master's thesis, University of Warsaw, 2017.

[8] S. Travers, The complexity of membership problems for circuits over sets of integers, Theor. Comput. Sci. 369(1–3) (2006) 211–229.