The Covering and Boundedness Problems for Branching Vector Addition Systems

Stéphane Demri[1], Marcin Jurdziński[2], Oded Lachish[2], Ranko Lazić[2]

[1]LSV, ENS Cachan, CNRS, INRIA Saclay, France

[2]DIMAP, Department of Computer Science, University of Warwick, UK

ABSTRACT. The covering and boundedness problems for branching vector addition systems are shown complete for doubly-exponential time.

1 Introduction

Vector addition systems (shortly, VAS), or equivalently Petri nets (e.g., [14]), are a fundamental model of computation, which is more expressive than finite-state machines and less than Turing-powerful. Decidability and complexity of a variety of problems have been extensively studied ([6] is a comprehensive survey).

A k-dimensional VAS consists of an initial vector of non-negative integers, and a finite set of vectors of integers, all of dimension k. Let us call the initial vector *axiom*, and the other vectors *rules*. A computation can then be thought of as a *derivation*: it starts with the axiom, and at each step, the next vector is derived from the current one by adding a rule. The vectors of interest are the ones derived *admissibly*, i.e. at the end of a derivation which is such that none of the vectors derived during it contains a negative entry.

Covering and boundedness are two central decision problems for VAS. The former asks whether a vector that is pointwise greater than or equal to a given vector can be admissibly derived, and the latter asks whether the set of all admissibly derived vectors is finite. In a landmark article [12], Rackoff showed that covering and boundedness for VAS are in EXPSPACE, matching Lipton's lower bound of EXPSPACE-hardness [10].* Considering the expressively equivalent VAS with states (shortly, VASS), Rosier and Yen refined the proofs of Lipton and Rackoff to obtain almost matching lower and upper bounds in terms of three parameters: the dimension, the binary size of the maximum absolute value of an entry in a rule, and the number of states [15]. Lipton's result was also extended by Mayr and Meyer to reversible Petri nets, which are equivalent to commutative semigroups [11]. Building further on Rosier and Yen's work, Habermehl showed that space exponential in the size of the system and polynomial in the size of the formula suffices for model checking the propositional linear-time μ-calculus on VASS, and he obtained a matching lower bound already for LTL on BPP [7].

*We recommend `http://rjlipton.wordpress.com/2009/04/08/an-expspace-lower-bound/`.

Foundations of Software Technology and Theoretical Computer Science (Kanpur) 2009.
Editors: Ravi Kannan and K. Narayan Kumar; pp 181–192
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany.
Digital Object Identifier: 10.4230/LIPIcs.FSTTCS.2009.2317

The following is a natural extension of VAS: instead of linearly, computation proceeds from the leaves to the root of a tree. For each node which is not a leaf, its vector is derived by summing the vectors derived at its children and adding a rule vector. The same condition of admissibility applies, i.e. no derived vector may contain a negative entry. This model of computation is branching VAS (shortly, BVAS).

In recent years, it has turned out that BVAS have interesting connections to a number of formalisms:

- BVAS correspond to a class of linear index grammars in computational linguistics [13];
- reachability (i.e. admissible derivability) for BVAS is decidable iff provability in multiplicative exponential linear logic is decidable [4];
- Verma and Goubault-Larrecq have extended the computation of Karp and Miller trees [8] to BVAS, and used it to draw conclusions about a class of equational tree automata which are useful for analysing cryptographic protocols [17];
- if first-order logic with 2 variables on finite data trees (which has applications to the XPath query language for XML) is decidable, then so is reachability for BVAS [1].

Covering and boundedness for BVAS are decidable easily using the branching extension of Karp and Miller's procedure [17]. However, the resulting algorithms do not operate in primitive recursive time or space, even in the linear case [16].

The main results we report are that, by switching from VAS to BVAS, covering and boundedness move two notches up the complexity hierarchy, to 2EXPTIME-complete.

For the 2EXPTIME-memberships, consider the following simple-minded idea for transferring knowledge about VAS derivations to the branching case:

✳ Every simple path from a leaf to the root in a BVAS derivation is a VAS derivation.

We show that the idea can give us mileage, but only after the following new insight, which is needed because the subderivations that grow off the simple path and hence contribute summands to it make the resulting VAS contain rules with unbounded positive entries.

☞ For VAS, we can obtain similar upper bounds to Rackoff's, but which depend only on the dimension and the minimum negative entry in a rule, i.e. not on the maximum positive entry in a rule.

The insight is at the centre of our proofs. In the case of covering, we show it essentially by inspecting carefully a proof of Rackoff, but in the case of boundedness, it relies on proving a new result on small solutions of integer programming problems, which extends a classical theorem of Borosh and Treybig and may also be a contribution of wider interest. To complete the proofs of the 2EXPTIME-memberships, we provide arguments for reducing the heights of appropriate BVAS derivations to at most doubly-exponential, and for why resulting small witnesses can be guessed and verified by alternating Turing machines in exponential space.

To obtain 2EXPTIME-hardness for covering and boundedness for BVAS, we extend the proof of Lipton to show that computations of alternating machines of size N with counters bounded by 2^{2^N} can be simulated in reverse by BVAS of size $O(N^2)$. Although universal branchings of alternating counter machines copy counter valutations whereas BVAS sum vectors derived at children nodes, the inner workings of Lipton's construction enable us to add a bit of machinery by which the BVAS can simulate the copying. We remark that, as is the case with Lipton's result, the lower bound is shown already for BVAS whose rules contain only entries -1, 0 or 1.

After fixing notations and making some preliminary observations in the next section, that covering and boundedness are in 2EXPTIME is shown in Sections 3 and 4, respectively. We then argue in Section 5 that both problems are 2EXPTIME-hard.

2 Preliminaries

Numbers, vectors and matrices. We write $\mathbb{N}_+$, $\mathbb{N}$ and $\mathbb{Z}$ for the sets of all positive, non-negative and arbitrary integers, respectively. Since we shall only work with integers, let the open interval (a, b) denote $(a, b) \cap \mathbb{Z}$, and analogously for half-open and closed intervals.

Given a dimension $k \in \mathbb{N}$, let $\mathbf{0}$ denote the zero vector and, for each $i \in [1, k]$, $\mathbf{e}_i$ denote the ith unit vector. For $\mathbf{v}, \mathbf{w} \in \mathbb{Z}^k$ and $B \in \mathbb{Z}$, we write:

- $\mathbf{v}(1), \ldots, \mathbf{v}(k)$ for the entries of $\mathbf{v}$;
- $\mathrm{supp}(\mathbf{v})$ for the set of all $i \in [1, k]$ such that $\mathbf{v}(i) \neq 0$;
- $\mathbf{v} \leq \mathbf{w}$ iff $\mathbf{v}(i) \leq \mathbf{w}(i)$ for all $i \in [1, k]$, and $\mathbf{v} < \mathbf{w}$ iff $\mathbf{v} \leq \mathbf{w}$ and $\mathbf{v} \neq \mathbf{w}$;
- $\min(B, \mathbf{v})$ for the vector $\langle \min\{B, \mathbf{v}(1)\}, \ldots, \min\{B, \mathbf{v}(k)\} \rangle$, and analogously for max;
- $\mathbf{v}^-$ for the vector $-\min(0, \mathbf{v})$, and $\mathbf{v}^+$ for the vector $\max(0, \mathbf{v})$.

For $\mathbf{v} \in \mathbb{N}^k$, let $\max(\mathbf{v}) = \max\{\mathbf{v}(1), \ldots, \mathbf{v}(k)\}$, where in case $k = 0$, we have $\max(\langle\rangle) = \max \emptyset = 0$. For finite $R \subseteq \mathbb{Z}^k$, let $\max(R^{-/+})$ denote $\max\{\max(\mathbf{r}^{-/+}) : \mathbf{r} \in R\}$, respectively.

Let $S^{k \times n}$ denote the set of all matrices with k rows, n columns and entries from S. Conveniently albeit slightly eccentrically, we use $-i$ for an index i to denote all rows or columns other than the ith, and $\bullet$ to denote all rows or columns. For example, $\mathbf{A}_{i\bullet}$ is row i of $\mathbf{A}$, and $\mathbf{A}_{\bullet(-j)}$ is $\mathbf{A}$ with column j removed.

Trees. A finite binary tree $\mathcal{T}$, which may contain nodes with one child, is a non-empty finite subset of $\{1, 2\}^*$ such that, for all $n \in \{1, 2\}^*$ and $i \in \{1, 2\}$, $n \cdot 2 \in \mathcal{T}$ implies $n \cdot 1 \in \mathcal{T}$, and $n \cdot i \in \mathcal{T}$ implies $n \in \mathcal{T}$. The nodes of $\mathcal{T}$ are its elements. The root of $\mathcal{T}$ is ε, the empty word. All notions such as parent, first child, second child, subtree and leaf, have their standard meanings. The height of $\mathcal{T}$ is the length, i.e. the number of nodes, of the longest simple path from the root to a leaf.

BVAS. The systems we define are equivalent to the branching vector addition systems with states [17] and the vector addition tree automata [4, 1]. To simplify our technical life, we work with stateless systems. In the linear case, it is well-known that states can be eliminated in logarithmic space, e.g. by adding the number of states to the dimension. For branching systems, the same is true, but computation steps that join two vectors by addition need to be generalised so that a vector from a fixed finite set (which may contain negative entries) is added also. Since we are not studying the systems as recognisers of languages, we do not have to work with alphabets either. Another simplification which costs only a logarithmic amount of space is in relation to the VATA [4], where branching up to a fixed finite arity was permitted. Hence, adopting a proof-theoretic terminology like that of Verma and Goubault-Larrecq [17], a system will consist of finite sets of axioms, unary rules and binary rules, all of which are simply integral vectors. The unary rules are present for easy compatibility with the linear case.

Let a *branching vector addition system* (*BVAS*) be a tuple $\mathcal{B} = \langle k, A_0, R_1, R_2 \rangle$, where:

- $k \in \mathbb{N}$ is the dimension;
- $A_0 \subseteq \mathbb{N}^k$ is a non-empty finite set of axioms;
- $R_1, R_2 \subseteq \mathbb{Z}^k$ are finite sets of unary and binary rules, respectively.

A derivation starts with a number of integral vectors, proceeds by applying the rules, and finishes with a single vector. Applying a unary rule means adding it to a derived vector, and applying a binary rule means adding it to the sum of two derived vectors. For a vector to be considered produced by the system, it needs to be derived by a derivation which starts with the axioms and whose derived vectors are all non-negative.

Formally, a *derivation* of $\mathcal{B}$ is a labelling $\mathcal{D} : \mathcal{T} \to \mathbb{Z}^k$ such that:

- $\mathcal{T}$ is a finite binary tree;
- if n has one child in $\mathcal{T}$, then $\mathcal{D}(n) \in R_1$;
- if n has two children in $\mathcal{T}$, then $\mathcal{D}(n) \in R_2$.

The vectors that are derived at every node are obtained recursively as follows:

- if n is a leaf in $\mathcal{T}$, then $\widehat{\mathcal{D}}(n) = \mathcal{D}(n)$;
- if n has one child n' in $\mathcal{T}$, then $\widehat{\mathcal{D}}(n) = \mathcal{D}(n) + \widehat{\mathcal{D}}(n')$;
- if n has two children n' and n'' in $\mathcal{T}$, then $\widehat{\mathcal{D}}(n) = \mathcal{D}(n) + \widehat{\mathcal{D}}(n') + \widehat{\mathcal{D}}(n'')$.

Now, we say that $\mathcal{D}$:

- is *initialised* iff, for each leaf n of $\mathcal{T}$, we have $\mathcal{D}(n) \in A_0$;
- is *admissible* iff, for each node n of $\mathcal{T}$, we have $\widehat{\mathcal{D}}(n) \in \mathbb{N}^k$;
- *derives* $\widehat{\mathcal{D}}(\varepsilon)$, which is the vector derived at the root.

For $\mathbf{v} \in \mathbb{N}^k$, we say that $\mathcal{B}$ *produces* $\mathbf{v}$ iff some initialised admissible derivation of $\mathcal{B}$ derives $\mathbf{v}$.

Substitutions and contractions. For finite binary trees $\mathcal{T}$ and $\mathcal{T}'$, and a node n of $\mathcal{T}$, let $\mathcal{T}[n \leftarrow \mathcal{T}']$ denote the tree obtained by replacing with $\mathcal{T}'$ the subtree of $\mathcal{T}$ rooted at n. To extend the notation to derivations, for $\mathcal{D} : \mathcal{T} \to \mathbb{Z}^k$ and $\mathcal{D}' : \mathcal{T}' \to \mathbb{Z}^k$, and a node n of $\mathcal{T}$, let $\mathcal{D}[n \leftarrow \mathcal{D}'] : \mathcal{T}[n \leftarrow \mathcal{T}'] \to \mathbb{Z}^k$ denote the derivation obtained by replacing with $\mathcal{D}'$ the subderivation of $\mathcal{D}$ rooted at n. Observe that the vector derived at node $n^\dagger$ in $\mathcal{D}[n \leftarrow \mathcal{D}']$ is:

- $\widehat{\mathcal{D}'}(n')$, if $n^\dagger$ corresponds to the node n' of $\mathcal{D}'$;
- $\widehat{\mathcal{D}}(n^\dagger) - \widehat{\mathcal{D}}(n) + \widehat{\mathcal{D}'}(\varepsilon)$, if $n^\dagger$ is an ancestor of n;
- $\widehat{\mathcal{D}}(n^\dagger)$, otherwise.

When $\mathcal{D}'$ has only one leaf n, we write $\mathcal{D}; \mathcal{D}'$ instead of $\mathcal{D}'[n \leftarrow \mathcal{D}]$.

For a derivation $\mathcal{D}$ and its nodes n and n' such that n is an ancestor of n', we write $\mathcal{D}[n \leftarrow n']$ instead of $\mathcal{D}[n \leftarrow \mathcal{D}']$, where $\mathcal{D}'$ is the subderivation of $\mathcal{D}$ rooted at n'. We call such substitutions *contracting*. For two derivations $\mathcal{D}^\dagger$ and $\mathcal{D}^\ddagger$, we say that $\mathcal{D}^\ddagger$ is a *contraction* of $\mathcal{D}^\dagger$ iff $\mathcal{D}^\ddagger$ is obtained from $\mathcal{D}^\dagger$ by a finite sequence of contracting substitutions.

VAS. The classical vector addition systems can be defined as BVAS of the form $\mathcal{V} = \langle k, \{\mathbf{a}\}, R, \emptyset \rangle$, i.e. with one axiom and no binary rules. We may write them as just $\langle k, \mathbf{a}, R \rangle$.

All the definitions for BVAS apply to VAS, but they simplify. For each derivation $\mathcal{D} : \mathcal{T} \to \mathbb{Z}^k$, its underlying tree $\mathcal{T}$ is a sequence.

Restrictions and bounds. For k-dimensional X, and $I \subseteq [1,k]$, we write $X(I)$ for the "restriction of X to the set of places I", e.g.: $\mathbf{v}(I)$ is the vector obtained from $\mathbf{v}$ by removing the entries in places outside of I; $\langle k, \mathbf{a}, R\rangle(I)$ is the $|I|$-dimensional VAS obtained from $\langle k, \mathbf{a}, R\rangle$ by replacing $\mathbf{a}$ with $\mathbf{a}(I)$, and by replacing every rule $\mathbf{r} \in R$ with $\mathbf{r}(I)$; and $\mathcal{D}(I)$ is the derivation obtained from $\mathcal{D}$ by replacing, for every node n, the label $\mathcal{D}(n)$ of n with $\mathcal{D}(n)(I)$.

For $\mathbf{v} \in \mathbb{Z}^k$ and $B \in \mathbb{N}$, we say that $\mathbf{v}$ is *B-bounded* iff $\mathbf{v} \in [0, B-1]^k$. We regard a derivation *B-bounded* iff all the vectors derived at its nodes are B-bounded. Thus, B-boundedness implies admissibility.

For a k-dimensional vector or derivation X, and $I \subseteq [1,k]$, we say that X is *I-B-bounded* iff $X(I)$ is B-bounded.

Decision problems. We study the complexity of the following problems. As is standard, the input sizes are with respect to binary representations of integers.

Covering Given a BVAS $\mathcal{B}$ and a target non-negative vector $\mathbf{t}$ of the same dimension, does $\mathcal{B}$ produce some $\mathbf{v}$ such that $\mathbf{v} \geq \mathbf{t}$?

Boundedness Given a BVAS, is the set of all vectors that it produces finite?

THEOREM 1. *[10, 12] Covering and boundedness for VAS are* EXPSPACE*-complete.*

THEOREM 2. *[17] Covering and boundedness for BVAS are decidable.*

3 Upper bound for the covering problem

We say that a derivation $\mathcal{D}$ of a BVAS $\mathcal{B}$ is a *covering* of a vector $\mathbf{t}$ iff the vector that $\mathcal{D}$ derives is at least $\mathbf{t}$, i.e. $\widehat{\mathcal{D}}(\varepsilon) \geq \mathbf{t}$. Thus, the covering problem asks whether there exists an initialised admissible covering.

For VAS, Rackoff [12] established EXPSPACE-membership of the covering problem by showing that, if an initialised admissible covering exists, then there must exist one of at most doubly-exponential length. Such a "short" covering can be guessed and verified in non-deterministic exponential space, and determinism is regained by Savitch's Theorem.

More precisely, Rackoff proved:

LEMMA 3. *[12, Section 3] If a VAS $\langle k, \mathbf{a}, R\rangle$ has an initialised admissible covering of $\mathbf{t} \in \mathbb{N}^k$, then it has one whose length is at most $2^{(3L)^{k+1}}$, where $L = \max\{\text{size}(R), \text{size}(\mathbf{t})\}$.*

Now, the following proof scheme suggests itself for showing that, if a k-dimensional BVAS $\mathcal{B}$ has an initialised admissible covering $\mathcal{D}$ of $\mathbf{t}$, then it has one of at most doubly-exponential height:

(i) If $\mathcal{D}$ has an excessively high leaf n, let $\mathcal{V}$ be the VAS whose axiom is $\mathcal{D}(n)$ and whose rules R are all the vectors:
   - $\mathcal{D}(n')$, such that n' is on the path π from n to the root, and has one child;
   - $\mathcal{D}(n') + \widehat{\mathcal{D}}(n'')$, such that n' is on π, and n'' is a child of n' not on π.

   Hence, the sequence obtained from π by relabelling the nodes with two children as specified is a derivation $\mathcal{D}^\dagger$ of $\mathcal{V}$. The vectors derived along $\mathcal{D}^\dagger$ are the same as the vectors derived along π in $\mathcal{D}$, so $\mathcal{D}^\dagger$ is an initialised admissible covering of $\mathbf{t}$.

(ii) By Lemma 3, $\mathcal{V}$ has an initialised admissible covering $\mathcal{D}^\ddagger$ of $\mathbf{t}$ with length at most $2^{(3L)^{k+1}}$, where $L = \max\{\text{size}(R), \text{size}(\mathbf{t})\}$.

(iii) Let $\mathcal{D}'$ be a derivation of $\mathcal{B}$ obtained from $\mathcal{D}^\ddagger$ by undoing the linearisation done in (i), i.e. by unfolding each rule in $\mathcal{D}^\ddagger$ which is not a unary rule of $\mathcal{B}$ into a binary rule of $\mathcal{B}$ and a subderivation of $\mathcal{D}$. It is straightforward to check that $\mathcal{D}'$ is also an initialised admissible covering of $\mathbf{t}$. We repeat from (i) with $\mathcal{D}'$ instead of $\mathcal{D}$, until there are no excessively high leaves.

There are, unfortunately, two obstacles:

- Since the definition of R in (i) involves adding derived vectors (the ones at the nodes one edge away from the path π), we have no bound on size(R) in terms of size$(\mathcal{B})$ and size$(\mathbf{t})$, and therefore neither on L in (ii).
- Even if we manage to bound L, Lemma 3 gives us no guarantees about the shape of $\mathcal{D}^\ddagger$ in (ii) in relation to the shape of $\mathcal{D}^\dagger$. Hence, although the length of $\mathcal{D}^\ddagger$ is bounded, we are not able to deduce that after the unfolding in (iii), $\mathcal{D}'$ has fewer excessively high leaves than $\mathcal{D}$.

However, the key to overcoming both obstacles is observing that essentially Rackoff's proof of Lemma 3 shows more than is stated in that result! Firstly, any initialised admissible covering has a contraction which is a short initialised admissible covering, and secondly, the length of the latter is bounded by the sizes of the target vector and only the negative entries in the rules of the VAS. More precisely, we have:

LEMMA 4. *If a VAS $\langle k, \mathbf{a}, R\rangle$ has an initialised admissible covering $\mathcal{D}$ of $\mathbf{t} \in \mathbb{N}^k$, then it has one which is a contraction of $\mathcal{D}$ and whose length is at most* $(\max(R^-) + \max(\mathbf{t}) + 2)^{(3k)!}$.

We are now in a position to show that, indeed, if a given BVAS has an initialised admissible covering of a given vector of non-negative integers, then it has one of at most doubly-exponential height. Although that is all that is required in this article, the actual statement is stronger for the record.

LEMMA 5. *If a BVAS $\langle k, A_0, R_1, R_2\rangle$ has an initialised admissible covering $\mathcal{D}$ of $\mathbf{t} \in \mathbb{N}^k$, then it has one which is a contraction of $\mathcal{D}$ and whose height is at most* $(\max((R_1 \cup R_2)^-) + \max(\mathbf{t}) + 2)^{(3k)!}$.

Therefore, to decide the covering problem, it suffices to search for an initialised admissible covering of at most doubly-exponential height. Note, however, that the size of a binary tree of doubly-exponential height can be triply-exponential, and hence vectors derived in a derivation of doubly-exponential height may contain triply-exponential entries. In order to prove the main result of this section, i.e., that the covering problem for is in 2EXPTIME, we need to avoid having to manipulate such large numbers. That is achieved by our next result, Proposition 6, which shows that for a large enough bound B, whether a derivation is admissible and a covering can be verified accurately even if entries in the derived vectors are truncated to be at most B.

For a derivation $\mathcal{D} : \mathcal{T} \to \mathbb{Z}^k$ and $B \in \mathbb{N}$, we define the *B-truncated* derived vectors by:

- if n is a leaf in $\mathcal{T}$, then $\widehat{\mathcal{D}}^B(n) = \min(B, \mathcal{D}(n))$;
- if n has one child n' in $\mathcal{T}$, then $\widehat{\mathcal{D}}^B(n) = \min(B, \mathcal{D}(n) + \widehat{\mathcal{D}}^B(n'))$;
- if n has two children n' and n'' in $\mathcal{T}$, then $\widehat{\mathcal{D}}^B(n) = \min(B, \mathcal{D}(n) + \widehat{\mathcal{D}}^B(n') + \widehat{\mathcal{D}}^B(n''))$.

PROPOSITION 6. *Suppose $\mathcal{B} = \langle k, A_0, R_1, R_2 \rangle$ is a BVAS, $\mathbf{t} \in \mathbb{N}^k$, $\mathcal{D}$ is a derivation in $\mathcal{B}$ of height at most H, and $B \geq H \cdot \max((R_1 \cup R_2)^-) + \max(\mathbf{t})$. Then $\mathcal{D}$ is an admissible covering of $\mathbf{t}$ iff, for each node n in $\mathcal{D}$, $\widehat{\mathcal{D}}^B(n) \geq \mathbf{0}$, and $\widehat{\mathcal{D}}^B(\varepsilon) \geq \mathbf{t}$.*

THEOREM 7. *Covering for BVAS is in* 2EXPTIME.

PROOF. Let $\mathcal{B} = \langle k, A_0, R_1, R_2 \rangle$ be a BVAS and $\mathbf{t} \in \mathbb{N}^k$. Let $N = \text{size}(\mathcal{B}) + \text{size}(\mathbf{t})$. If $\ell = \max((R_1 \cup R_2)^-) + \max(\mathbf{t}) + 2$ then $\ell \leq 2^N$, and without any loss of generality we can assume that $3k \leq N$.

Lemma 5 implies that if there is an initialised admissible covering of $\mathbf{t}$ in $\mathcal{B}$ then there is one of height at most $\ell^{(3k)!} \leq (2^N)^{N!} \leq 2^{2^{\mathsf{C}_1 N \log N}}$, for some constant $\mathsf{C}_1 > 1$. If we set $H = 2^{2^{\mathsf{C}_1 N \log N}}$ and $B = H^2$, then from Proposition 6 it follows that in order to establish existence of an initialised admissible covering of $\mathbf{t}$ in $\mathcal{B}$, it suffices to:

- guess an initialised derivation $\mathcal{D}$ in $\mathcal{B}$ of height at most H;
- guess the B-truncated derived vectors at all nodes in $\mathcal{D}$, and for every node and its children, verify that they satisfy the equations defining B-truncated derived vectors, and that they are non-negative;
- verify that the B-truncated derived vector at the root covers $\mathbf{t}$.

We argue that the guessing and verification of such a structure of at most triply-exponential size can be carried out by an alternating Turing machine with exponential space, and hence the covering problem is in 2EXPTIME [3]. The alternating Turing machine starts at the root of the derivation, it uses non-deterministic states to guess the rules labelling the current node and its children, and their B-truncated derived vectors, and it uses universal states to proceed with the guessing and verification process to both children (for nodes labelled by binary rules) in parallel. All those tasks can indeed be carried out by a Turing machine with only exponential space because it can represent—in binary—and manipulate numbers of doubly-exponential magnitude. ∎

4 Upper bound for the boundedness problem

Let us say that a derivation $\mathcal{D}$ is *self-covering* iff, for some node n, the vector derived at n is less than or equal to the one at the root, and less in at least one place, i.e. $\widehat{\mathcal{D}}(n) < \widehat{\mathcal{D}}(\varepsilon)$.

The following fact tells us that boundedness is equivalent to non-existence of an initialised admissible self-covering derivation. The "if" part is easy. The "only if" part was inferred by Verma and Goubault-Larrecq, using the properties of their extension of Karp and Miller's procedure.

THEOREM 8. *[17] A BVAS produces infinitely many vectors iff it has an initialised admissible self-covering derivation.*

In the simpler setting of VAS, to conclude that boundedness is in EXPSPACE, Rackoff showed that if an initialised admissible self-covering derivation exists, then there exists one of at most doubly-exponential length:

LEMMA 9. *[12, Section 4] If a VAS $\mathcal{V} = \langle k, \mathbf{a}, R\rangle$ has an initialised admissible self-covering derivation, then it has one whose length is at most $2^{2^{\mathsf{C}_2 L \log L}}$, where $L = \mathrm{size}(R)$ and C_2 is some constant.*

Encouraged by our eventual success in Section 3, consider the following scheme for proving that, if a BVAS $\mathcal{B} = \langle k, A_0, R_1, R_2\rangle$ has an initialised admissible self-covering derivation $\mathcal{D}$, then it has one of at most doubly-exponential height:

(I) Let node n be such that $\widehat{\mathcal{D}}(n) < \widehat{\mathcal{D}}(\varepsilon)$, and pick a simple path π in $\mathcal{D}$ which is from a leaf to the root and passes through n. Let $\mathcal{V}$ be the VAS defined as in (i) in Section 3, i.e. its axiom is the label of the leaf of π and its rules R are obtained by linearising the binary rules on π. Thus, $\mathcal{V}$ has a derivation $\mathcal{D}^\dagger$ whose sequence of derived vectors is the same as the sequence of derived vectors along π in $\mathcal{D}$. In particular, $\mathcal{D}^\dagger$ is initialised, admissible and self-covering.

(II) By Lemma 9, $\mathcal{V}$ has an initialised admissible self-covering derivation $\mathcal{D}^\ddagger$ whose length is at most $2^{2^{\mathsf{C}_2 L \log L}}$, where $L = \mathrm{size}(R)$.

(III) Let $\mathcal{D}'$ be a derivation of $\mathcal{B}$ obtained from $\mathcal{D}^\ddagger$ by undoing the linearisation done in (I), as in (iii) in Section 3, and let π' be the path in $\mathcal{D}'$ that is from a leaf to the root and corresponds to $\mathcal{D}^\ddagger$. It is straightforward to check that $\mathcal{D}'$ is also initialised, admissible and self-covering.

(IV) Let H be the length of π', which equals the length of $\mathcal{D}^\ddagger$. For each node n' that is one edge away from π' in $\mathcal{D}'$ (i.e., that was attached in (III)), the subderivation of $\mathcal{D}'$ rooted at n' is an initialised admissible covering of $\min((H-1)\cdot\max(R^-)+1, \widehat{\mathcal{D}'}(n'))$. By Lemma 5, $\mathcal{B}$ has an initialised admissible covering $\mathcal{D}^*_{n'}$ of the same vector, whose height is at most

$$\left(\max((R_1\cup R_2)^-) + \max\left(\min\left((H-1)\cdot\max(R^-)+1, \widehat{\mathcal{D}'}(n')\right)\right)+2\right)^{(3k)!}$$
$$\le \left(\max((R_1\cup R_2)^-) + (H-1)\cdot\max(R^-)+3\right)^{(3k)!}$$
$$\le \left(H\cdot\max((R_1\cup R_2)^-)+3\right)^{(3k)!}.$$

Let $\mathcal{D}''$ be obtained from $\mathcal{D}'$ by performing each substitution $[n' \leftarrow \mathcal{D}^*_{n'}]$. The truncating threshold $(H-1)\cdot\max(R^-)+1$ is such that $\mathcal{D}''$ is still admissible and self-covering, certainly it is still initialised, and $H + (H\cdot\max((R_1\cup R_2)^-)+3)^{(3k)!}$ bounds its height.

Of course, we have the same problem as the first one in Section 3: we have no bound on $\mathrm{size}(R)$ in terms of $\mathrm{size}(\mathcal{B})$, and therefore neither on H in (IV). Seeking therefore a refinement of Lemma 9, we find that the key ingredient in its proof is:

LEMMA 10. *[12, Lemma 4.5] Suppose $\mathcal{V} = \langle k, \mathbf{a}, R\rangle$ is a VAS, $I \subseteq [1,k]$ and $B > 1$. If $\mathcal{V}$ has an initialised I-B-bounded self-covering derivation, then it has one whose length is at most $B^{(\mathrm{size}(R))^{\mathsf{C}_3}}$, where C_3 is some constant.*

In turn, at the centre of the proof of Lemma 10, Rackoff invokes the following theorem of Borosh and Treybig on small solutions of integer linear programming problems. Recall that the interval notations denote sets of integers.

THEOREM 11. *[2] Let* $\mathbf{A} \in (-m, m)^{k \times n}$ *and* $\mathbf{b} \in (-m, m)^k$*, where* $k, n, m \in \mathbb{N}$*. If there exists* $\mathbf{x} \in \mathbb{N}^n$ *such that* $\mathbf{Ax} \geq \mathbf{b}$*, then there exists* $\mathbf{y} \in [0, (\max\{n, m\})^{\mathsf{C}_4 k}]^n$ *such that* $\mathbf{Ay} \geq \mathbf{b}$*, where* C_4 *is some constant.*

When we examine feeding a VAS $\langle k, \mathbf{a}, R \rangle$ for which we have a bound on $\max(R^-)$ but not on $\max(R^+)$ into Rackoff's proof of Lemma 10, we discover that Theorem 11 is invoked for bounded k, unbounded n, $\mathbf{A}$ whose entries are bounded below but not above, and $\mathbf{b}$ whose entries are bounded above but not below. Surprisingly, this is where we can make progress. We now show that, if we can afford roughly one exponential more, small solutions exist for $\mathbf{A}$ and $\mathbf{b}$ which are only one-sidedly bounded by m. Moreover, the number of non-zero entries in the small solutions and their values are bounded only in terms of k and m.

THEOREM 12. *Let* $\mathbf{A} \in (-m, \infty)^{k \times n}$ *and* $\mathbf{b} \in (-\infty, m)^k$*, where* $k, n, m \in \mathbb{N}$*. If there exists* $\mathbf{x} \in \mathbb{N}^n$ *such that* $\mathbf{Ax} \geq \mathbf{b}$*, then there exists* $\mathbf{y} \in [0, L]^n$ *such that* $|\mathrm{supp}(\mathbf{y})| \leq L$ *and* $\mathbf{Ay} \geq \mathbf{b}$*, where* $L = m^{2^{\mathsf{C}_5 k^2}}$ *and* C_5 *is some constant.*

In order to reformulate Theorem 12 so that it becomes appropriate for a proof by induction on k (cf. Lemma 14), we define $F_k(m)$, for all integers $k \geq 1$ and $m \geq 2$, by:

$$F_k(m) = \begin{cases} m & \text{if } k = 1, \\ \left(F_{k-1}(2m)\right)^{4\mathsf{C}_4 k^2} & \text{if } k > 1, \end{cases}$$

where C_4 is the constant from Theorem 11, which we can assume is at least 1.

PROPOSITION 13. *For all integers* $k \geq 1$ *and* $m \geq 2$*, we have* $F_k(m) \leq m^{(4\mathsf{C}_4)^k \cdot (2k)!}$*.*

Observe that there is a constant C_5 such that, for all integers $k \geq 1$ and $m \geq 2$, we have $F_k(m) \leq m^{(4\mathsf{C}_4)^k \cdot (2k)!} \leq m^{2^{\mathsf{C}_5 k^2}}$. Hence, and since Theorem 12 is true trivially when $k = 0$ or $m \leq 1$, Theorem 12 follows from the following lemma.

LEMMA 14. *Let* $\mathbf{A} \in (-m, \infty)^{k \times n}$ *and* $\mathbf{b} \in (-\infty, m)^k$*, where* $k \geq 1$*,* n *and* $m \geq 2$ *are integers. If there exists* $\mathbf{x} \in \mathbb{N}^n$ *such that* $\mathbf{Ax} \geq \mathbf{b}$*, then there exists* $\mathbf{y} \in [0, F_k(m)]^n$ *such that* $|\mathrm{supp}(\mathbf{y})| \leq F_k(m)$ *and* $\mathbf{Ay} \geq \mathbf{b}$*.*

PROOF. We can assume without any loss of generality that, for each $j \in [1, n]$, there exists $\mathbf{x} \in \mathbb{N}^n$ such that $\mathbf{Ax} \geq \mathbf{b}$ and $\mathbf{x}(j) \geq 1$. Otherwise, consider $\mathbf{A}' = \mathbf{A}_{\bullet(-j)}$, where there exists no $\mathbf{x} \in \mathbb{N}^n$ such that $\mathbf{Ax} \geq \mathbf{b}$ and $\mathbf{x}(j) \geq 1$.

The proof is by induction on k. First we consider the base case when $k = 1$. If $\mathbf{b} \leq 0$ then $\mathbf{Ay} \geq \mathbf{b}$ for $\mathbf{y} = \mathbf{0}$. If, however, $\mathbf{b} > 0$ then the existence of $\mathbf{x} \in \mathbb{N}^n$ such that $\mathbf{Ax} \geq \mathbf{b}$ implies that there must be $i \in [1, n]$ such that $\mathbf{A}(1, i) > 0$. Then, we have $\mathbf{Ay} \geq \mathbf{b}$ for $\mathbf{y} = m \cdot \mathbf{e}_i$.

For the inductive step we consider the following three cases. Essentially, if either $\mathbf{b}$ contains a large negative entry or $\mathbf{A}$ contains a large positive entry, then we remove that row of $\mathbf{A}$ and argue by the inductive hypothesis and the largeness of the entry. Otherwise, we have a lower bound for all entries of $\mathbf{b}$ and an upper bound for all entries of $\mathbf{A}$, and we invoke Theorem 11.

Case 1. *There exists $i \in [1,k]$ such that $\mathbf{b}(i) \leq -m \cdot (F_{k-1}(m))^2$.* Let $\mathbf{A}' = \mathbf{A}_{(-i)\bullet}$ and let $\mathbf{b}' = \mathbf{b}_{-i}$. By the inductive hypothesis, there exists $\mathbf{y} \in [0, F_{k-1}(m)]^n$—and hence $\mathbf{y} \in [0, F_k(m)]^n$—such that $|\mathrm{supp}(\mathbf{y})| \leq F_{k-1}(m) < F_k(m)$ and $\mathbf{A}'\mathbf{y} \geq \mathbf{b}'$. The assumption that $\mathbf{A}(i,j) > -m$ for all $j \in [1,n]$ then implies that $\mathbf{A}_{i\bullet}\mathbf{y} > -m \cdot (F_{k-1}(m))^2 \geq \mathbf{b}(i)$, and hence we have $\mathbf{A}\mathbf{y} \geq \mathbf{b}$.

Case 2. *There exist $i \in [1,k]$ and $j \in [1,n]$ such that $\mathbf{A}(i,j) \geq 2m \cdot (F_{k-1}(2m))^2$, and there exists $\mathbf{x} \in \mathbb{N}^n$ such that $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\mathbf{x}(j) \geq 1$.* Let $\mathbf{A}' = \mathbf{A}_{(-i)\bullet}$, let $\mathbf{b}' = \mathbf{b}_{-i}$, and let $\mathbf{b}'' = \mathbf{b}' - \mathbf{A}_{(-i)j}$. Note that $\mathbf{A}'(\mathbf{x} - \mathbf{e}_j) \geq \mathbf{b}''$ and that, since $\mathbf{x}(j) \geq 1$, we have $\mathbf{x} - \mathbf{e}_j \in \mathbb{N}^n$. Observe also that $\mathbf{b}'' \in (-\infty, 2m)^{k-1}$ and hence, by the inductive hypothesis, there exists $\mathbf{y} \in [0, F_{k-1}(2m)]^n$ such that $|\mathrm{supp}(\mathbf{y})| \leq F_{k-1}(2m)$ and $\mathbf{A}'\mathbf{y} \geq \mathbf{b}''$.

Let $\mathbf{z} = \mathbf{y} + \mathbf{e}_j$. Note that then $\mathbf{z} \in [0, F_{k-1}(2m) + 1]^n \subseteq [0, F_k(m)]^n$ and $|\mathrm{supp}(\mathbf{y})| \leq F_{k-1}(2m) + 1 \leq F_k(m)$, and hence we only need to establish that $\mathbf{A}\mathbf{z} \geq \mathbf{b}$. We have:

$$(\mathbf{A}\mathbf{z})(i) = \mathbf{A}_{i\bullet}(\mathbf{y} + \mathbf{e}_j) \geq \mathbf{A}(i,j) - m \cdot (F_{k-1}(2m))^2 \geq m \cdot (F_{k-1}(2m))^2 \geq m \geq \mathbf{b}(i),$$

where the first inequality follows from $\mathbf{A} \in (-m, \infty)^{k \times n}$, from $\mathbf{y} \in [0, F_{k-1}(2m)]$, and from $|\mathrm{supp}(\mathbf{y})| \leq F_{k-1}(2m)$; and the second inequality follows from the assumption that $\mathbf{A}(i,j) \geq 2m \cdot (F_{k-1}(2m))^2$. Moreover, we have:

$$(\mathbf{A}\mathbf{z})_{-i} = \mathbf{A}'(\mathbf{y} + \mathbf{e}_j) = \mathbf{A}'\mathbf{y} + \mathbf{A}_{(-i)j} \geq \mathbf{b}'' + \mathbf{A}_{(-i)j} = \mathbf{b}' = \mathbf{b}_{-i}.$$

Case 3. *Neither Case 1 nor Case 2 applies.* Observe that, in this case, every column of $\mathbf{A}$ is in $[-m, 2m \cdot (F_{k-1}(2m))^2]^k$, and $\mathbf{b} \in [-m \cdot (F_{k-1}(m))^2, m]^k$. The number of distinct columns of $\mathbf{A}$ is therefore at most $(3m \cdot (F_{k-1}(2m))^2)^k \leq (F_{k-1}(2m))^{4k}$, and so without loss of generality we may assume $n \leq (F_{k-1}(2m))^{4k}$. By Theorem 11, there exists $\mathbf{y} \in [0, F_{k-1}(2m)^{4\mathsf{C}_4 k^2}]^n = [0, F_k(m)]^n$ such that $|\mathrm{supp}(\mathbf{y})| \leq (F_{k-1}(2m))^{4k} \leq F_k(m)$ and $\mathbf{A}\mathbf{y} \geq \mathbf{b}$. ∎

By substituting the use of Theorem 11 in Rackoff's proof of Lemma 10 by a use of Theorem 12, we obtain:

LEMMA 15. *Suppose $\mathcal{V} = \langle k, \mathbf{a}, R \rangle$ is a VAS, $I \subseteq [1,k]$ and $B > 1$. If $\mathcal{V}$ has an initialised I-B-bounded self-covering derivation, then it has one of length at most $((\max(R^-) + 1) \cdot B)^{2^{\mathsf{C}_6 k^2}}$, where C_6 is some constant.*

The final step in obtaining a revision of Lemma 9 that we can apply to VAS whose rules are bounded below but not above is to substitute in its proof uses of Lemma 10 by uses of Lemma 15. That yields the following result, which shows that we could indeed afford the extra exponential in Theorem 12. Although it filters through to Lemma 15, it gets swallowed by the steps of Rackoff's inductive proof of Lemma 9.

LEMMA 16. *If a VAS $\mathcal{V} = \langle k, \mathbf{a}, R \rangle$ has an initialised admissible self-covering derivation, then it has one of length at most $(2(\max(R^-) + 1))^{2^{\mathsf{C}_7 k^3}}$, where C_7 is some constant.*

THEOREM 17. *Boundedness for BVAS is in* 2EXPTIME.

5 Lower bounds

Let a *counter machine* consist of finite sets of states, counters and transitions. Each transition changes state, and either increments a counter, or checks that a counter is positive and decrements it, or checks that a counter is zero. We consider alternating counter machines, where the set of states is partitioned into non-deterministic and universal. Without loss of generality, we restrict to at most binary branching. A computation of such a machine is a binary tree of configurations, each of which is a state together with a non-negative integer for every counter.

To establish lower bounds for the covering and boundedness problems for BVAS, we reduce from the following problem. Its AEXPSPACE-hardness is an easy consequence of standard translations from Turing machines to counter machines (e.g., by simulating the tape by two stacks and encoding the latter by counters), and so it is 2EXPTIME-hard [3].

Doubly-exponential halting Given an alternating counter machine of size N with an initial state and a halting state, does it have an initialised 2^{2^N}-bounded halting computation, i.e. whose root is the initial state with 0 for every counter, in which every counter value is less than 2^{2^N}, and which is finite and such that the state of each leaf is halting?

We argue that, given an alternating counter machine $\mathcal{M}$ of size N, a BVAS $\mathcal{B}_\mathcal{M}$ which simulates $\mathcal{M}$ and is of size $O(N^2)$ is computable:

- For simulating the operations on counters, we employ Lipton's construction [10] (cf. the nice presentation by Esparza [5, Section 7]), in which each counter c of $\mathcal{M}$ is represented by two places p_c and $\overline{p_c}$ of $\mathcal{B}_\mathcal{M}$, and it is an invariant in all initialised admissible derivations of $\mathcal{B}_\mathcal{M}$ that the sum of p_c and $\overline{p_c}$ is 2^{2^N}. Increments and decrements of c are easy, but to simulate checking that c is zero, $\mathcal{B}_\mathcal{M}$ uses implementations of two auxiliary counters bounded by $2^{2^{N-1}}$ to decrement $\overline{p_c}$ exactly $2^{2^{N-1}} \cdot 2^{2^{N-1}} = 2^{2^N}$ times. The implementations of the two auxiliary counters in turn require two auxiliary counters bounded by $2^{2^{N-2}}$ etc.
- The simulation is performed in reverse, so that $\mathcal{B}_\mathcal{M}$ guesses and verifies an initialised 2^{2^N}-bounded halting computation of $\mathcal{M}$. To verify a universal branching, where the two child configurations of $\mathcal{M}$ are represented by two derived vectors $\mathbf{v}$ and $\mathbf{v}'$, $\mathcal{B}_\mathcal{M}$ derives $\mathbf{v}''$ from $\mathbf{v}'$ by transferring each pair of places that represents a counter of $\mathcal{M}$ to a separate pair of places which is reserved for that purpose. Then, $\mathcal{B}_\mathcal{M}$ joins $\mathbf{v}$ and $\mathbf{v}''$ by performing a binary rule, verifies that the values of each counter of $\mathcal{M}$ were the same in $\mathbf{v}$ and $\mathbf{v}'$, and empties the auxiliary places.
- Since $\mathcal{B}_\mathcal{M}$ can simulate checking that every counter of $\mathcal{M}$ is zero, it can guess and verify that the configuration that it represents is initial.

To reduce to the covering problem, we use the target vector to specify that the reverse simulation has reached the initial configuration of $\mathcal{M}$. To reduce to the boundedness problem, we amend $\mathcal{B}_\mathcal{M}$ so that upon guessing and verifying that the configuration of $\mathcal{M}$ is initial, it becomes unbounded by deriving an infinite sequence of increasing vectors.

THEOREM 18. *Covering and boundedness for BVAS are* 2EXPTIME*-hard.*

6 Concluding remarks

The extra work in this article in relation to the proofs of Lipton and Rackoff [10, 12], and the recent result that reachability for BVAS is 2EXPSPACE-hard [9] (the highest known lower bound for VAS is Lipton's), indicate that BVAS are not a trivial extension of VAS.

We would like to thank Serge Haddad (LSV, Cachan) for numerous discussions about VAS and their extensions, Sylvain Schmitz (LSV, Cachan) for pointing us to [13], and Alexander Schrijver (CWI, Amsterdam) for correspondence about integer linear programming.

References

[1] M. Bojańczyk, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. *J. ACM*, 56(3), 2009.

[2] I. Borosh and L. B. Treybig. Bounds on positive integral solutions of linear Diophantine equations. *Proc. AMS*, 55:299–304, 1976.

[3] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.

[4] P. de Groote, B. Guillaume, and S. Salvati. Vector addition tree automata. In *LICS*, pages 64–73. IEEE, 2004.

[5] J. Esparza. Decidability and complexity of Petri net problems — an introduction. In *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lect. Notes Comput. Sci.*, pages 374–428. Springer, 1998.

[6] J. Esparza and M. Nielsen. Decidability issues for Petri nets — a survey. *Bull. EATCS*, 52:244–262, 1994.

[7] P. Habermehl. On the complexity of the linear-time mu-calculus for Petri nets. In *ICATPN*, volume 1248 of *Lect. Notes Comput. Sci.*, pages 102–116. Springer, 1997.

[8] R. M. Karp and R. E. Miller. Parallel program schemata. *J. Comput. Syst. Sci.*, 3(2):147–195, 1969.

[9] R. Lazić. The branching reachability problem requires doubly-exponential space. Manuscript, 2009.

[10] R. J. Lipton. The reachability problem requires exponential space. Technical Report 62, Dep. Comput. Sci., Yale Univ., Jan. 1976.

[11] E. W. Mayr and A. R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Adv. Math.*, 46:305–329, 1982.

[12] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6(2):223–231, 1978.

[13] O. Rambow. Multiset-valued linear index grammars: imposing dominance constraints on derivations. In *ACL*, pages 263–270. Morgan Kaufmann, 1994.

[14] W. Reisig. *Petri Nets: An Introduction*, volume 4 of *Monographs in Theor. Comput. Sci. An EATCS Series*. Springer, 1985.

[15] L. Rosier and H.-C. Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *J. Comput. Syst. Sci.*, 32:105–135, 1986.

[16] R. Valk and G. Vidal-Naquet. Petri nets and regular languages. *J. Comput. Syst. Sci.*, 23(3):299–325, 1981.

[17] K. N. Verma and J. Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discr. Math. and Theor. Comput. Sci.*, 7:217–230, 2005.