A Polynomial-Time Algorithm for Reachability in Branching VASS in Dimension One

Stefan Göller[*1], Christoph Haase[*1], Ranko Lazić[†2], and Patrick Totzke[†2]

1 **LSV, CNRS & ENS Cachan**
Université Paris-Saclay, France
{goeller,haase}@lsv.ens-cachan.fr
2 **DIMAP, Department of Computer Science**
University of Warwick, United Kingdom
{r.s.lazic,p.totzke}@warwick.ac.uk

Abstract

Branching VASS (BVASS) generalise vector addition systems with states by allowing for special branching transitions that can non-deterministically distribute a counter value between two control states. A run of a BVASS consequently becomes a tree, and reachability is to decide whether a given configuration is the root of a reachability tree. This paper shows P-completeness of reachability in BVASS in dimension one, the first decidability result for reachability in a subclass of BVASS known so far. Moreover, we show that coverability and boundedness in BVASS in dimension one are P-complete as well.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases branching vector addition systems, reachability, coverability, boundedness

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Vector addition systems with states (VASS), equivalently known as Petri nets, are a fundamental model of computation which comprise a finite-state controller with a finite number of counters ranging over the naturals. The number of counters is usually refereed to as the dimension of the VASS. A configuration $q(\boldsymbol{n})$ of a VASS in dimension d consists of a control state q and a valuation $\boldsymbol{n} \in \mathbb{N}^d$ of the counters. A transition of a VASS can increment and decrement counters and is enabled in a configuration whenever the resulting counter values are all non-negative, otherwise the transition is disabled. Consequently, VASS induce an infinite transition system. Three of the most fundamental decision problems for VASS are reachability, coverability and boundedness. Given a target configuration $q(\boldsymbol{n})$ and some initial configuration, reachability is to decide whether starting in the initial configuration there exists a path ending in $q(\boldsymbol{n})$ in the induced infinite transition system. Coverability asks whether some configuration $q(\boldsymbol{n}')$ can be reached for some $\boldsymbol{n}' \geq \boldsymbol{n}$, where $\geq$ is defined component-wise. Boundedness is the problem to decide whether there are infinitely many different configurations reachable from a given starting configuration. Those decision problems find a plethora of applications, for instance in the verification of concurrent programs.

* Supported by Labex Digicosme, Univ. Paris-Saclay, project VERICONISS.
† Supported by the EPSRC, grants EP/M011801/1 and EP/M027651/1.

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–17

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Coverability can, for example, be used in order to validate mutual exclusion properties of shared-memory concurrent programs [6]; reachability is a key underlying decision problem in the verification of liveness properties of finite-data asynchronous programs [5]. Even though the complexity of coverability and boundedness are well-understood and known to be EXPSPACE-complete [12, 14], the precise complexity of reachability remains a major unsolved problem; a non-primitive recursive upper bound ($\mathbf{F}_{\omega^3}$) has only recently been established [11] and the best known lower bound is EXPSPACE [12].

The situation is even more dissatisfying when considering *branching extensions* of VASS. Such *branching VASS (BVASS)* are additionally equipped with special branching transitions of the form (q, p, p'). When in a configuration $q(\boldsymbol{n})$, a BVASS can simultaneously non-deterministically branch into configurations $p(\boldsymbol{m})$ and $p'(\boldsymbol{m}')$ such that $\boldsymbol{n} = \boldsymbol{m} + \boldsymbol{m}'$. Reachability of a configuration $q(\boldsymbol{n})$ then is to decide whether there exists a proof tree whose root is labelled with $q(\boldsymbol{n})$ and whose leaves are all labelled with designated target control states in which all counters have value zero; coverability and boundedness are defined analogously as above. While coverability and boundedness are known to be 2-EXPTIME-complete [3], reachability in BVASS is not known to be decidable, not even in any fixed dimension. Recently, non-elementary lower bounds for reachability in BVASS have been obtained [10]. Reachability in BVASS is closely related and in fact equivalent to decidability of the multiplicative-exponential fragment of linear logic [2], and also an underlying decision problem in various other applications for instance in computational linguistics, cryptographic protocol verification, data logics and concurrent program verification; see [10] for more details.

The primary contribution of this paper is to provide a polynomial-time algorithm for reachability in BVASS in dimension one (BVASS_1) and to show that reachability is in fact P-complete. To the best of our knowledge, we give the first decidability result for reachability in a fragment of BVASS. Let us remark that a decidability result with such low complexity is actually quite surprising. On the one hand, due to the infinite state space of BVASS_1 it is not immediate that reachability is decidable. In particular, the emptiness problem for conjunctive grammars over a unary alphabet, which can be seen as a slight generalisation of BVASS_1 with special alternating transitions that can simultaneously branch into two control states while retaining the same counter value (known as ABVASS_1), is undecidable [9]. On the other hand, if we disallow branching rules in ABVASS_1 and thus obtain AVASS_1 then reachability is PSPACE-complete [15, 8].

It is in fact not too difficult to show that if a configuration is reachable in a BVASS_1 then there exists a so-called reachability tree of exponential size. What causes a main challenge when establishing a polynomial-time algorithm is that this bound is optimal in the sense that there exist families of BVASS_1 whose reachability trees are inherently of exponential size and contain an exponential number of different counter values. Consequently, reachability cannot be witnessed in polynomial time by explicitly constructing a witnessing reachability tree. Instead, we show that a polynomial-time computable certificate for the reachability of a configuration suffices. These certificates have two parts: the first is a table that, for certain $d > 0$ contains those pairs of control states q and residue classes r modulo d such that $q(n)$ is reachable for some sufficiently large n with $n \equiv r \bmod d$. This is called residue reachability. The second part is a compressed collection of incomplete small reachability trees, so-called expandable partial reachability trees, whose leaves are either accepting configurations or have the special property of being pumpable. The latter means that on the path from a pumpable leaf to the root there is another node, called the anchor of the leaf, that is labelled with the same control state and the counter value between the anchor and the leaf increases

strictly. By identifying further structural properties of reachability trees, we show that the subtree between a pumping leaf and an anchor can under certain circumstances be repeated multiple times, and hence pumping leafs may achieve an arbitrarily large counter value in a certain residue class. This eventually enables us to witness the existence of a reachability tree via residue reachability.

In Section 5, we show that coverability and boundedness are P-complete for BVASS_1. For coverability, the upper bound follows easily via a reduction to reachability. For boundedness, this is not the case and we require a specifically tailored argument.

Due to space constraints, the proofs of some statements can be found in an appendix.

2 Preliminaries

We write $\mathbb{Z}$ and $\mathbb{N}$ for the sets of integers and non-negative integers, respectively, and define $[i,j] \stackrel{\text{def}}{=} \{i, i+1, \ldots, j-1, j\}$, for given integers $i < j$. For $d \geq 1$ we define $\mathbb{Z}_d \stackrel{\text{def}}{=} [0, d-1]$.

The set of finite words over alphabet A is denoted by A^* and the length of a word $w \in A^*$ is written as $|w|$. For two words $u, v \in A^*$, we say u is a *prefix* of v (written as $u \preceq v$) if $v = uw$ for some $w \in A^*$. It is a *strict prefix* ($u \prec v$) if $u \preceq v$ and $u \neq v$. We say u and v are *incomparable* if neither $u \preceq v$ nor $v \preceq u$. A set $U \subseteq A^*$ is *prefix-closed* if for all $u \in U$ and all $v \in A^*$ we have that $v \preceq u$ implies $v \in U$.

Let Σ be a set. A *Σ-labelled (finite) tree* is a mapping $T \colon U \to \Sigma$ where $U \subseteq A^*$ is a non-empty finite prefix-closed set of *nodes* for some finite set A. For $U \subseteq U$, we define $T(U) \stackrel{\text{def}}{=} \{T(u) \mid u \in U\}$. A *leaf* of T is a node $u \in U$ such that there is no $v \in U$ with $u \prec v$; every node of T that is not a leaf is called *inner node*. A node u is an *ancestor* (resp. *descendant*) of a node v if $u \preceq v$ (resp. $v \preceq u$) and a *strict ancestor* (resp. *strict descendant*) if $u \prec v$ (resp. $v \prec u$). For any node u we define the *subtree of T rooted at u* as $T^{\downarrow u} \colon u^{-1}U \to \Sigma$, where $u^{-1}U \stackrel{\text{def}}{=} \{x \in A^* \mid ux \in U\}$ and $T^{\downarrow u}(x) \stackrel{\text{def}}{=} T(ux)$. Note that $u^{-1}U$ is a prefix-closed subset of U. We define $h(u) \stackrel{\text{def}}{=} \max\{|x| \mid x \in u^{-1}U\}$ to be the *height* of the subtree rooted at u and and define $h(T) \stackrel{\text{def}}{=} h(\varepsilon)$. Note that $h(u) = 0$ if, and only if, u is a leaf. We say T is *binary* if $U \subseteq \{0,1\}^*$; in this case if for some node $u \in U$ we have that $u0 \in U$, then $u0$ the *left child* of u and if $u1 \in U$ we say that $u1$ is the *right child* of u.

2.1 Branching Vector Addition Systems

In the following, $\boldsymbol{n}$ and $\boldsymbol{z}$ will denote elements from $\mathbb{N}^k$ and $\mathbb{Z}^k$, respectively; addition on $\mathbb{Z}^k$ is defined component-wise.

▶ **Definition 1.** Let $k \geq 1$. A *k-dimensional branching vector addition system with states (BVASS_k)* is a tuple $\mathcal{B} = (Q, \Delta, F)$ where Q is a finite set of *control states*, $\Delta \subseteq Q^3 \cup (Q \times \{-1,0,1\}^k \times Q)$ is a finite set of *transitions*, and $F \subseteq Q$ is a set of *final states*. The *size* $|\mathcal{B}|$ of a BVASS is defined as $|\mathcal{B}| \stackrel{\text{def}}{=} |Q| + k \cdot |\Delta|$.

The semantics of BVASS is given in terms of reachability trees. A *partial reachability tree* of a BVASS_k $\mathcal{B}$ is a $Q \times \mathbb{N}^k$-labelled binary tree $T \colon U \to Q \times \mathbb{N}^k$, where each inner node $u \in U$ with $T(u) = (q, \boldsymbol{n})$ satisfies exactly one of the following conditions:

- branching rule (top down) $u0, u1 \in U$, and if $T(u0) = (p, \boldsymbol{n}_0)$ and $T(u1) = (p', \boldsymbol{n}_1)$, then $\boldsymbol{n} = \boldsymbol{n}_0 + \boldsymbol{n}_1$ and $(q, p, p') \in \Delta$; or u1
- $u0 \in U$, ~~$u1 \notin U$~~, and if $T(u0) = (p, \boldsymbol{n}_0)$, then $\boldsymbol{n} = \boldsymbol{n}_0 + \boldsymbol{z}$ and $(p, \boldsymbol{z}, q) \in \Delta$. bottom up

A *reachability tree* is a partial reachability tree T where $T(u) \in F \times \{0\}^k$ for all leaves u of T. We call these nodes *accepting* nodes. For each $j \in \mathbb{N}$ we say that a partial reachability tree

Figure 1 Illustration of the BVASS$_1$ $\mathcal{B}_n$. The reachability set of the control state q_n is the singleton set $\{2^n\}$, and a reachability tree for $q(0)$ contains all counter values between 0 and 2^n.

T is *j-bounded* if $T(u) \in Q \times [0,j]^k$ for all $u \in U$. We call $Q \times \mathbb{N}^k$ the set of *configurations* of $\mathcal{B}$ and for the sake of readability often write its elements $(q, \boldsymbol{n})$ as $q(\boldsymbol{n})$. We say that a configuration $q(\boldsymbol{n})$ is *reachable* if there exists a reachability tree T with $T(\varepsilon) = q(\boldsymbol{n})$. Note that in particular every configuration in $F \times \{0\}^k$ is reachable. The reachability set $reach(q)$ of a control state q is defined as $reach(q) \stackrel{\text{def}}{=} \{n \in \mathbb{N} \mid q(n) \text{ is reachable}\}$. The decision problem that we mainly focus on in this paper is *reachability*, defined as follows:

REACHABILITY IN BVASS$_k$
INPUT: A BVASS$_k$ $\mathcal{B} = (Q, \Delta, F)$, a control state q and $\boldsymbol{n} \in \mathbb{N}^k$ encoded in unary.
QUESTION: Is $q(\boldsymbol{n})$ reachable?

Our main result is that reachability is P-complete in dimension one.

▶ **Theorem 2.** *Reachability in BVASS$_1$ is* P*-complete.*

3 Lower Bounds

alternatively, from emptiness of non-deterministic tree automata. this is surprising since the counter is not used in the lower bound

As a warm-up exercise and in order to familiarise ourselves with BVASS$_1$, we begin with proving a couple of lower bounds for the reachability problem. First, it is not difficult to see that the reachability problem is P-hard via a reduction from the monotone circuit value problem (MCVP) [13]. By simulating $\vee$-gates of a Boolean by ~~by~~ non-deterministic branching and $\wedge$-gates by splitting transitions, the following statement can easily be obtained.

▶ **Proposition 3.** *Let $\mathcal{C}$ be a Boolean circuit. There exists a logspace computable BVASS$_1$ $\mathcal{B}$ with a control state q such that $q(0)$ is reachable if, and only if, $\mathcal{C}$ evaluates to true.*

A challenging aspect when providing a polynomial-time upper bound for reachability in BVASS$_1$ is that reachability trees may be of exponential size and may contain an exponential number of nodes labelled with distinct counter values. To see this, consider the family $(\mathcal{B}_n)_{n \geq 0}$ of BVASS$_1$, where $\mathcal{B}_n \stackrel{\text{def}}{=} (Q_n, \Delta_n, F)$ and where $Q_n \stackrel{\text{def}}{=} \{q, q_f\} \cup \{q_0, \ldots, q_n\}$, $\Delta_n \stackrel{\text{def}}{=} \{(q, +1, q), (q, 0, q_n)\} \cup \{(q_i, q_{i-1}, q_{i-1}) \mid 0 < i \leq n\} \cup \{(q_0, -1, q_f)\}$ and $F \stackrel{\text{def}}{=} \{q_f\}$. The construction is illustrated in Figure 1. It is easily seen that $q_i(N)$ is reachable if, and only if, $N = 2^i$. Observe that $reach(q) = \{0, \ldots, 2^n\}$ is finite and that the reachability tree of $q(0)$ contains all counter values between 0 and 2^n. In particular, this allows us to obtain the following hardness result in which the updates of the BVASS$_1$ are from $\{-1, 0, +1\}$ (i.e. encoded in unary), but the initial configuration is given in binary, via a straight-forward reduction from the NP-complete SUBSET SUM problem [13].

▶ **Proposition 4.** *Reachability in BVASS$_1$ is* NP*-hard if the initial configuration $q(n)$ is given in binary.*

It is worth mentioning that the previous lemma enables us to derive as a corollary an NP-lower bound for reachability in BVASS$_2$. This is in contrast to VASS where there is no difference between the NL-completeness of reachability in dimensions one and two [16, 4].

▶ **Corollary 5.** *Reachability in $BVASS_2$ is* NP*-hard.*

4 Reachability in BVASS_1

Here, we show that reachability in BVASS_1 is decidable in polynomial time, thereby establishing the P upper bound claimed in Theorem 2. In the first part, we consider a variation of the reachability problem in which we are only interested in reaching configurations that are sufficiently large and lie in a certain residue class. Subsequently, we will apply this intermediate result for showing that reachability can be witnessed by small partial reachability trees. Finally, we put everything together in order to obtain a polynomial-time algorithm.

4.1 The Residue Reachability Problem

A cornerstone of our algorithm for reachability in BVASS_1 is the polynomial-time decidability of the following variant of the reachability problem for BVASS_1:

(modular reachability)

RESIDUE REACHABILITY FOR BVASS_1

q0(n0)

INPUT: A BVASS_1 $\mathcal{B} = (Q, \Delta, F)$, a configuration ~~$q(n_0)$~~ and $d \geq 1$, where n_0 and d are given in unary.

QUESTION: Does there exist some $n \geq n_0$ such that $q_0(n)$ is reachable and $n \equiv n_0 \bmod d$?

The main result of this section is that residue reachability for BVASS_1 is decidable in polynomial time. Notice that setting $d = 1$ allows for checking whether there exists some $n \geq n_0$ such that $q(n)$ is reachable. We first introduce some auxiliary definitions that allow us to abstract away concrete counter values of reachability trees. A *partial d-residue tree* is a binary tree $T \colon U \to Q \times \mathbb{Z}_d$, where each inner node $u \in U$ with $T(u) = (q, n)$ satisfies precisely one of the following conditions:

(i) $u0, u1 \in U$, and if $T(u0) = (p, m_0)$ and $T(u1) = (p', m_1)$ then $n \equiv m_0 + m_1 \bmod d$ and $(q, p, p') \in \Delta$;

this allows m + z to be negative

(ii) $u0 \in U, u1 \notin U$, and if $T(u0) = (p, m)$ then $n = m + z \bmod d$ and $(p, z, q) \in \Delta$.

We call a configuration from $Q \times \mathbb{Z}_d$ a *residue configuration*. Given a set of configurations S, its *residue* is $S/\mathbb{Z}_d \stackrel{\text{def}}{=} \{(q, n \bmod d) \in Q \times \mathbb{Z}_d \mid q(n) \in S\}$. Likewise, given a partial reachability tree $T \colon U \to Q \times \mathbb{N}$, the *residue* $T/\mathbb{Z}_d$ *of* T is $T/\mathbb{Z}_d \colon U \to Q \times \mathbb{Z}_d$, where $T/\mathbb{Z}_d(u) \stackrel{\text{def}}{=} T(u)/\mathbb{Z}_d$ for all $u \in U$. Clearly, $T/\mathbb{Z}_d$ is a partial residue tree.

For the remainder of this section, fix some BVASS_1 $\mathcal{B} = (Q, \Delta, F)$, some configuration $q_0(n_0)$ and some $d \geq 1$, where n_0 and d are given in unary. In order to decide residue reachability, one might be tempted to start with an initial configuration and then to repeatedly apply transitions of $\mathcal{B}$ modulo d until the desired residue configuration is discovered. Such an approach would, however, not be sound as it may lead to residue configurations that, informally speaking, can only be obtained by forcing the counter to drop below zero. Also, the simple alternative of constructing a sufficiently large reachability tree is futile as it may be of exponential size, cf. Section 3. In order to balance between those two extremes, we introduce reachability trees in which all nodes except of the root are required to be bounded by some value $j \in \mathbb{N}$: a partial reachability tree $T \colon U \to Q \times \mathbb{N}$ is *almost j-bounded* if $T(u) \in Q \times [0, j]$ for all $u \in U \setminus \{\varepsilon\}$. Note that every j-bounded partial reachability tree is almost j-bounded. The following constant will be particularly useful:

$N \stackrel{\text{def}}{=} |Q| \cdot d.$

Moreover, by S we denote the set of configurations for which there exists an $(n_0 + N)$-bounded reachability tree and define for $i < j$:

this is a finite set of polinomial size

$$S \stackrel{\text{def}}{=} \{(q, m) \in Q \times \mathbb{N} \mid q(m) \text{ has an } (n_0 + N)\text{-bounded reachability tree}\}$$

$$S[i, j] \stackrel{\text{def}}{=} S \cap Q \times [i, j].$$

▶ **Lemma 6.** *The set S is computable in polynomial time.*

For any set of residue configurations (modulo d) $V, W \subseteq Q \times \mathbb{Z}_d$, we define the following sets that contain the result of an application of a transition of $\mathcal{B}$ modulo d:

p, z, q (bottom-up)

$$\Delta(V) \stackrel{\text{def}}{=} \{(q, r + z \bmod d) \mid (\cancel{q, z, p}) \in \Delta, (p, r) \in V\}$$

$$\Delta(V, W) \stackrel{\text{def}}{=} \{(q, r_0 + r_1 \bmod d) \mid (q, p_0, p_1) \in \Delta, (p_0, r_0) \in V, (p_1, r_1) \in W\}.$$

Next, we inductively define a sequence of sets $R_i \subseteq Q \times \mathbb{Z}_d$ for $i \geq 0$ whose fixed point will allow for deciding residue reachability. The set R_0 consists of those pairs of control states and residue classes that can be witnessed by a reachability tree that is almost $(n_0 + N)$-bounded and whose root has a counter value at least $n_0 + N$, and the R_i for $i > 0$ are obtained by application of Δ :

$$R_0 \stackrel{\text{def}}{=} \{(q, n \bmod d) \in Q \times \mathbb{Z}_d \mid n \geq n_0 + N,\ q(n) \text{ has an almost } (n_0 + N)\text{-bounded reachability tree}\}$$

$$R_{i+1} \stackrel{\text{def}}{=} R_i \cup \Delta(R_i) \cup \Delta(R_i, S/\mathbb{Z}_d) \cup \Delta(S/\mathbb{Z}_d, R_i) \cup \Delta(R_i, R_i).$$

Since the cardinality of each R_i is at most N, it is easily seen that the sequence $(R_i)_{i \geq 0}$ reaches a fixed point which can be computed in polynomial time.

▶ **Lemma 7.** *The fixed point $R \stackrel{def}{=} \bigcup_{i>0} R_i$ equals R_N and is computable in polynomial time.*

In particular, R together with S yields the whole residue reachability set.

▶ **Lemma 8.** *The set $X \stackrel{def}{=} R \cup S[n_0, n_0+N]/\mathbb{Z}_d$ is computable in polynomial time. Moreover,*

$$X = \{(q, n \bmod d) \mid q \in Q, n \in \mathit{reach}(q), n \geq n_0\}.$$

Proof (sketch). Polynomial-time computability of X follows immediately from Lemmas 6 and 7. The proof of the stated equality is quite technical though not too difficult and deferred to the appendix. The crucial part for the inclusion "$\subseteq$" is to show that for every $i \in [0, N]$ and each $(q, r) \in R_i$ there exists some $n \in \mathit{reach}(q)$ with $n \geq n_0 + N - i$ and $n \equiv r \bmod d$ by induction on i. For the converse inclusion the only interesting case is when a potential reachability tree T is not $(n_0 + N)$-bounded. One first shows that all $\prec$-maximal nodes u in T with $T(u) \notin S$ satisfy $T(u)/\mathbb{Z}_d \in R_0$ and uses the fact that $\Delta(R, R) \subseteq R$ and $\Delta(R) \subseteq R$ to conclude $T(\varepsilon)/\mathbb{Z}_d \in R$. ◀

The main result of this section now follows directly from Lemma 8.

▶ **Theorem 9.** *Residue reachability for $BVASS_1$ is decidable in polynomial time.*

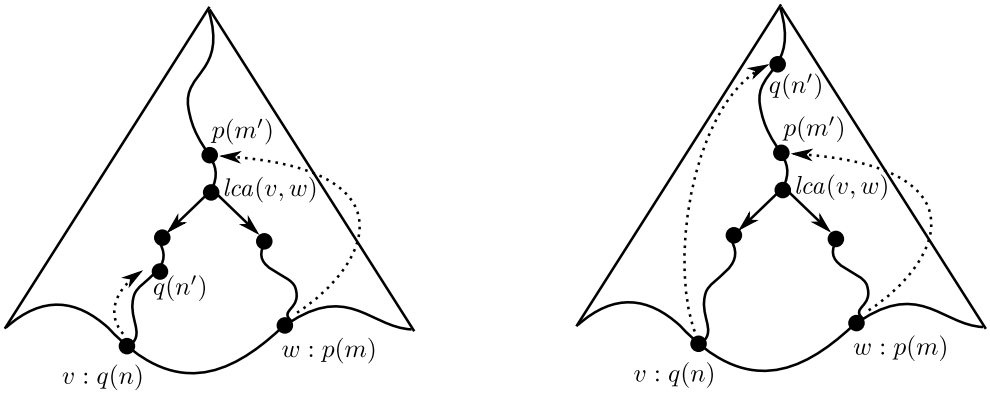

Figure 2 Illustration an exclusive (left) and a non-exclusive (right) partial reachability tree. Here, v and w are pumping nodes and anchor relationships are depicted as dashed arrows.

4.2 Expandable Partial Reachability Trees

We now employ our result on residue reachability to show that small partial reachability trees suffice in order to witness reachability. The key idea is to identify branches of partial reachability trees that end in a leaf and which could, informally speaking, be copied or pumped an arbitrary number of times, thus achieving a counter value in the leaf that is large enough and lies in a certain residue class of some modulus. Residue reachability then witnesses that such a leaf could be completed in order to yield a reachability tree. For the remainder of this section, fix some BVASS$_1$ $\mathcal{B} = (Q, \Delta, F)$.

Let us first introduce a couple of auxiliary definitions. Given a partial reachability tree $T\colon U \to Q \times \mathbb{N}$ and $v, w \in U$, the *lowest common ancestor* of $v, w \in U$ is defined as

$$lca(v,w) \stackrel{\text{def}}{=} \max\{u \in U \mid u \preceq v \text{ and } u \preceq w\},$$

where the maximum is taken with respect to $\preceq$. Let $T(u) = q(n)$, we define functions $state(u) \stackrel{\text{def}}{=} q$ and $counter(u) \stackrel{\text{def}}{=} n$ that allow us to access the control state and the counter value at u, respectively.

▶ **Definition 10.** A node $v \in U$ is a *pumping node* if there is a proper ancestor $u \prec v$ such that $state(u) = state(v)$ and $counter(u) < counter(v)$; the maximal such u is called the *anchor* of v. We say that T *exclusive* if

- every leaf v of T is either accepting or a pumping leaf; and
- the least common ancestor of any two distinct pumping leaves is a proper ancestor of at least one of their anchors.

Exclusive and non-exclusive partial reachability trees are illustrated in Figure 2. Observe that an accepting leaf can never be a pumping leaf and conversely a pumping leaf can never be accepting. A node u is said to be *exclusive* if $T^{\downarrow u}$ is exclusive. are anchors of pumping leaves allowed to be in T?

The next lemma states a useful fact that directly follows from the pigeon-hole principle: whenever the counter increases on a branch by a certain amount then the branch contains a pumping node and its anchor.

▶ **Lemma 11.** *Let u and v be nodes of a partial reachability tree such that $u \prec v$ and $counter(u) + |Q| \leq counter(v)$. Then there exists a pumping node v' with anchor u' such that $u \preceq u' \prec v' \preceq v$.*

Finally, we call T *expandable* if T is exclusive and every pumping leaf v with anchor u such that $T(v) = q(n)$ and $T(u) = q(m)$ induces a valid instance of the residue reachability problem, i.e., $q(l)$ is reachable for some $l \geq n$ and $l \equiv n \bmod (n-m)$. The following lemma shows that every reachability tree gives rise to an expandable reachability tree whose nodes have counter values bounded polynomially in $|\mathcal{B}|$.

completeness

▶ **Lemma 12.** *Suppose $q(n)$ is reachable an let $B \stackrel{\text{def}}{=} 2 \cdot |Q| + n$. Then there exists an expandable B-bounded partial reachability tree with root $q(n)$.*

Proof. Let T be a reachability tree with $T(\varepsilon) = q(n)$. We call a node w of T *large* if $\mathit{counter}(w) = B$. We obtain a partial reachability tree T' from T as follows. By Lemma 11, every large node w gives rise to at least one pair of nodes (u, v) such that $u \prec v \preceq w$ and v is a pumping node with anchor u. For every large node w that is minimal with respect to $\preceq$, we assign the maximal such pair $\mathrm{pair}(w) \stackrel{\text{def}}{=} (u, v)$ with respect to the lexicographical ordering on nodes (more precisely, $(u, v) \preceq (u', v')$ if either, $u \prec u'$, or $u = u'$ and $v \preceq v'$). Let $T' : U' \to Q \times \mathbb{N}$ denote the tree that one obtains from T by replacing all subtrees of T that are rooted at some node v such that $\mathrm{pair}(w) = (u, v)$ for some minimal (with respect to $\prec$) large node w in T by $\{v\}$ itself, i.e. such nodes v become leaves. It remains to prove that T' is B-bounded and expandable:

- T' is B-bounded since the w above are chosen minimal with respect to $\preceq$ and hence $\mathit{counter}(u) \leq B$ for all nodes $u \in U'$.
- T' is exclusive, which can be seen as follows. Striving for a contradiction, suppose that T' is not exclusive. Then there are distinct pumping nodes v, v' with anchors u, u' such that $u, u' \preceq w \stackrel{\text{def}}{=} \mathit{lca}(v, v')$. Since $\mathit{counter}(w) = \mathit{counter}(w0) + \mathit{counter}(w1) \leq B$, either $\mathit{counter}(w0) \leq B/2$ or $\mathit{counter}(w1) \leq B/2$, and assume without loss of generality that $\mathit{counter}(w0) \leq B/2$. Since $B - B/2 \geq |Q|$, by Lemma 11 there is another pumping node v'' with anchor u'' such that $w0 \preceq u'' \prec v''$, contradicting the assumed maximality of (u, v).
- It remains to show that T' is expandable. Since T is a reachability tree, we have that $T(u)$ is reachable and thus $T'(u)$ is reachable for all $u \in U'$, in particular is every pumping leaf u of T' a positive instance of the induced residue reachability problem (even modulo any modulus: if $T(u) = q(n)$, then simply choose $m \stackrel{\text{def}}{=} n$). ◀

We now turn towards the converse direction and show that every expandable tree witnesses reachability. We first state an auxiliary lemma about structural properties of nodes in exclusive trees whose proof can be found in the appendix.

▶ **Lemma 13.** *If u is a node of an exclusive partial reachability tree then the following hold:*

(i) If u is the anchor of a pumping leaf v then u is exclusive and all nodes w such that $u \prec w \preceq v$ are not exclusive.
(ii) u has at most one child that is not exclusive.

The previous lemma enables us to show that an expandable partial reachability tree implies the existence of a reachability tree.

soundness

▶ **Lemma 14.** *Let $T : U \to Q \times \mathbb{N}$ be an expandable partial reachability tree. Then for all $u \in U$, $T(u)$ is reachable or u is not exclusive.*

Proof. We prove the lemma by induction on $h(u)$. For the induction base, assume $h(u) = 0$, hence u is a leaf. Then u is either accepting and thus $T(u)$ is reachable, or u is not accepting and therefore a pumping leaf and thus not exclusive by Lemma 13(i).

For the induction step, suppose u is exclusive. We distinguish two cases:

- *All children of u are exclusive.* We only treat the case when u has two children, the case when u has one child can be proven analogously. If the children $u0$ and $u1$ of u are exclusive then by induction hypothesis there are reachability trees $T_0 \colon U_0 \to Q \times \mathbb{N}$ and $T_1 \colon U_1 \to Q \times \mathbb{N}$ with $T_0(\varepsilon) = T(u0)$ and $T_1(\varepsilon) = T(u1)$. We define the following tree $T_u \colon V \to Q \times \mathbb{N}$, where $V \stackrel{\text{def}}{=} \{0\}U_0 \cup \{1\}U_1 \cup \{\varepsilon\}$, $T_u(\varepsilon) \stackrel{\text{def}}{=} T(u)$ and $T_u(iv) \stackrel{\text{def}}{=} T_i(v)$ for all $i \in \{0,1\}$. Now T_u is a reachability tree, hence $T_u(\varepsilon) = T(u)$ is reachable.
- *Some child of u is not exclusive.* By Lemma 13(ii) there is at most one such child. Moreover, since u is exclusive but one child of u is not exclusive it must hold that u is the anchor of some unique pumping leaf v. Let $W \stackrel{\text{def}}{=} \{w \in U \mid u \preceq w \prec v\}$ be the set all nodes in T on the path from u to v excluding v. Let us assume without loss of generality that $W = \{u0^i \mid i \in [0,\ell]\}$ for some $\ell \in \mathbb{N}$, i.e. v is reachable from u by walking down along left children. Let $X \stackrel{\text{def}}{=} \{w1 \in U \mid w \in W\}$ and observe that $u^{-1}U = u^{-1}W \uplus u^{-1}X \uplus u^{-1}\{v\}$. By Lemma 13(i), all nodes in $W \setminus \{u\}$ are not exclusive and consequently, Lemma 13(ii) implies that all nodes $x \in X$ are exclusive. Hence by induction hypothesis, for every $x \in X$ there is a reachability tree T_x such that $T_x(\varepsilon) = T(x)$. Moreover, since T is expandable there exists some $m \geq \mathit{counter}(v)$ such that $q(m)$ is reachable and $m \equiv \mathit{counter}(v) \bmod d$, where $d \stackrel{\text{def}}{=} \mathit{counter}(v) - \mathit{counter}(u) \geq 1$ and $q \stackrel{\text{def}}{=} \mathit{state}(u) = \mathit{state}(v)$. It remains to show that $T(u) = q(\mathit{counter}(u))$ is reachable. We show how the reachability of $q(m)$ implies the reachability of $q(m - j \cdot d)$ for all $j \in \mathbb{N}$ satisfying $m - j \cdot d \geq \mathit{counter}(u)$; from which can conclude that $T(u)$ is reachable since $m \equiv \mathit{counter}(v) \equiv \mathit{counter}(u) \bmod d$. We only show this for $j = 1$, i.e. we prove that the reachability of $q(m)$ implies the reachability of $q(m-d)$: indeed, one can iterate the construction below to show that $q(m-2d), \ldots, q(\mathit{counter}(v)), q(\mathit{counter}(u))$ is reachable. For this, let $\widehat{T} \colon Z \to Q \times \mathbb{N}$ be a reachability tree for $q(m)$, i.e. $\widehat{T}(\varepsilon) = q(m)$. Let $\delta \stackrel{\text{def}}{=} m - \mathit{counter}(v) \geq 0$. Let the tree $T' \colon u^{-1}W \uplus u^{-1}X \uplus u^{-1}(vZ)$ be defined as follows:
  - $T'(u^{-1}w) \stackrel{\text{def}}{=} p(m+\delta)$ in case $T(w) = p(m)$, for all $w \in W$,
  - $T'(u^{-1}x) \stackrel{\text{def}}{=} T_x(\varepsilon)$, and
  - $T'(u^{-1}vz) \stackrel{\text{def}}{=} \widehat{T}(z)$ for all $z \in Z$.

  In other words T' is obtained from $T^{\downarrow u}$ by (i) adding to the configuration of all nodes in $u^{-1}W$ the non-negative value δ and by replacing the leaf that corresponds to leaf v (i.e. $u^{-1}v$) by the tree $\widehat{T}$. It is clear that the resulting tree is in fact a reachability tree and we have $T'(\varepsilon) = \mathit{counter}(u) + \delta = \mathit{counter}(u) + m - \mathit{counter}(v) = m - d$. Thus $q(m-d)$ is reachable. ◀

A consequence of the previous lemma is that in particular $T(\varepsilon)$ is reachable for every expandable partial reachability tree T. By combining Lemmas 12 and 14, we obtain the following characterisation of reachability in BVASS_1.

▶ **Proposition 15.** *A node $q(n)$ is reachable if, and only if, there exists an expandable B-bounded partial reachability tree T with $T(\varepsilon) = q(n)$, where $B \stackrel{\text{def}}{=} 2 \cdot |Q| + n$.*

4.3 The Algorithm

In this section, we provide an alternating logspace procedure for reachability in BVASS_1. This shows that reachability in BVASS_1 is decidable in deterministic polynomial time since alternating logspace equals deterministic polynomial time [1]. We employ the characterisation of reachability in BVASS_1 in terms of expandable B-bounded partial reachability of

Algorithm 1 An alternating logspace procedure for reachability in BVASS_1.

```
1: procedure Reach(q(n))
2:     if n > B then return false
3:     if q(n) ∈ F × {0} then return true
4:     else non-deterministically guess t ∈ Δ ∩ ({q} × Q × Q ∪ {q} × {−1, 0, 1} × Q)
5:         if t = (q, p, p′) ∈ Q³ then
6:             non-deterministically guess m, m′ ∈ [0, B] s.t. n = m + m′
7:             return (Reach(p(m)) and Reach(p′(m′)))
8:                 or (AnchorReach(q(n), p(m)) and Reach(p′(m′)))
9:                 or (AnchorReach(q(n), p′(m′)) and Reach(p(m)))
10:        else let t = (q, z, p) ∈ Q × {−1, 0, 1} × Q
11:            return Reach(q′(n + z)) or AnchorReach(q(n), p(n + z))

12: procedure AnchorReach(p(m), q(n))
13:    if max{m, n} > B then return false
14:    if p = q and n > m and ResidueReach(q(n), n − m) then return true
15:    else non-deterministically guess t ∈ Δ ∩ ({q} × Q × Q ∪ {q} × {−1, 0, 1} × Q)
16:        if t = (q, p′, p″) ∈ Q³ then
17:            non-deterministically guess m′, m″ ∈ [0, B] s.t. n = m′ + m″
18:            return AnchorReach(p(m), p′(m′)) and Reach(p″(m″))
19:                or AnchorReach(p(m), p″(m″)) and Reach(p′(m′))
20:        else let t = (q, z, p′) ∈ Q × {−1, 0, 1} × Q
21:            return AnchorReach(p(m), p′(n + z))
```

Proposition 15. First, by Theorem 9 we may assume the existence of an alternating logspace procedure for residue reachability in BVASS_1, i.e., an alternating logspace procedure $\textsc{ResidueReach}(q(n_0), d)$ that has an accepting computation if, and only if, $q(n)$ is reachable for some $n \geq n_0$ and $n \equiv n_0 \bmod d$. By application of this procedure, we show that one can construct an alternating logspace procedure $\textsc{Reach}(q(n))$ that takes a configuration $q(n)$ as input and that has an accepting computation if, and only if, there exists an expandable B-bounded partial reachability tree T with $T(\varepsilon) = q(n)$.

The idea is to simply to guess an expandable B-bounded partial reachability tree T in a top-down manner. The procedure Reach is defined above in Algorithm 1. First in Line 2, Reach rejects whenever the counter value n exceeds B and accepts if $q(n)$ is an accepting configuration (Line 3). Thus, subsequently we may assume that $n \leq B$. In Line 4, we non-deterministically choose a transition $t \in \Delta$. If $t = (q, p, p') \in Q^3$ is a branching rule, we non-deterministically guess how n can be decomposed as $n = m + m'$. Moreover, we non-deterministically guess whether the currently processed inner node of T labelled by $q(n)$ will be an anchor of some pumping leaf "below". If not then we simply recursively call $\textsc{Reach}(p(m))$ and $\textsc{Reach}(p'(m'))$ (Line 7). Otherwise, $q(n)$ will be the anchor of some pumping leaf that is either in the subtree "rooted at" $p(m)$ (Line 8) or in the subtree "rooted at" $p'(m')$ (Line 9). Speaking in terms of Lemma 13, either the inner node corresponding to configuration $p(m)$ is not exclusive or the one for $p'(m')$ is not exclusive. Suppose $p(m)$ is not exclusive, we then call a procedure $\textsc{AnchorReach}(p(m), q(n))$ that takes two configurations as arguments and tacitly assumes the first argument $p(m)$ "is" the anchor and the second argument $q(n)$ corresponds to some inner node that lies between the anchor and the pumping leaf it will eventually correspond to.

In more detail, analogously to Reach the procedure AnchorReach first checks whether the counter value of its input does not exceed B (Line 13). If so it checks whether $q(n)$ corresponds to a valid pumping leaf of $p(m)$, i.e., it induces a positive instance of the residue reachability problem by invoking $\textsc{ResidueReach}(q(m), m - n)$ (Line 14). If not then a rule $t \in \Delta$ is non-deterministically chosen (Line 15), and in case t is a branching rule, it is non-deterministically chosen which "child" of $q(n)$ is not exclusive, the other child is simply

checked for reachability by invoking procedure Reach (Lines 18 and 19).

Obviously, Reach and AnchorReach can be implemented in alternating logspace since the involved counter values are always bounded by $B+1$ and can hence be stored using a logarithmic number of bits.

5 Coverability and Boundedness

In this section, we show that the coverability and boundedness problem for BVASS_1 are also P-complete. The two problems are defined as follows:

Coverability and Boundedness in BVASS_1

INPUT: A BVASS_1 $\mathcal{B}=(Q,\Delta,F)$, a control state q and $n\in\mathbb{N}$ encoded in unary.
QUESTION: *Coverability:* Is there $m\geq n$ such that $q(m)$ is reachable?
Boundedness: Is $\mathit{reach}(q)$ finite?

If $q(n)$ is a positive instance of coverability then we call the configuration $q(n)$ *coverable*. A state q is *unbounded* whenever $\mathit{reach}(q)$ is unbounded (i.e. infinite).

Hardness for P is in both cases easily seen and similar to the P-hardness reduction from MCVP in Proposition 3. Moreover, the P upper bound for coverability follows easily from the P upper bound for residue reachability since $q(n)$ is coverable if, and only if, the pair $(q(n),1)$ is a positive instance of the residue reachability problem.

▶ **Theorem 16.** *Coverability in BVASS_1 is* P*-complete.*

The P upper bound for boundedness can, however, not immediately be derived. In particular, as discussed in Section 3, there exists a family of BVASS_1 $(\mathcal{B}_n)_{n\geq 0}$ with some control state q such that $\mathit{reach}(q)$ is finite but of cardinality 2^n.

For the remainder of this section, fix some BVASS_1 $\mathcal{B}=(Q,\Delta,F)$, we first provide sufficient and necessary criteria that witness that a control state is unbounded. Call a node v in a reachability tree *decreasing* if there is an ancestor $u\prec v$ with $\mathit{state}(u)=\mathit{state}(v)$ and $\mathit{counter}(u)>\mathit{counter}(v)$. The following lemma, whose proof is deferred to the appendix, shows that a reachability tree that contains some decreasing node witnesses that the control state at its root is unbounded.

▶ **Lemma 17.** *If a reachability tree T with $T(\varepsilon)=q(n)$ contains a decreasing node then q is unbounded.*

Conversely, the next lemma shows that a reachability tree whose root is labelled with a configuration with a sufficiently large counter value gives rise to a reachability tree which contains a decreasing node, informally speaking, shortly after its root.

▶ **Lemma 18.** *Suppose $n>2^{|Q|}$ with $n\in\mathit{reach}(q)$. There exists a reachability tree $T\colon U\to Q\times\mathbb{N}$ for $q(n')$ where $n'\geq n$, and which contains a decreasing node v with $|v|\leq|Q|$.*

A consequence of the two previous lemmas is that q is unbounded if, and only if, $\mathit{reach}(q)$ contains some $n>2^{|Q|}$. Even though the reachability trees in Lemma 18 are sufficient witnesses for unboundedness, they still contain much more information than necessary and are potentially of exponential size. In order to verify the existence of such a tree, exact counter values and in fact the subtrees rooted in v as well as all incomparable nodes can be abstracted away, as shown in the lemma below.

We first introduce some auxiliary notation. Write $\mathit{src}(t)\stackrel{\text{def}}{=}q$, $\mathit{trg}(t)\stackrel{\text{def}}{=}\{p,p'\}$ and $\mathit{eff}(t)\stackrel{\text{def}}{=}0$, for the *source* and *target states* and the *effect* of a branching transition $t=(q,p,p')\in Q^3$, respectively. Similarly, for $t=(q,d,p)$ define $\mathit{src}(t)\stackrel{\text{def}}{=}q$, $\mathit{trg}(t)\stackrel{\text{def}}{=}\{p\}$ and $\mathit{eff}(t)\stackrel{\text{def}}{=}d$.

▶ **Lemma 19.** *A control state p_0 is unbounded if, and only if, there is a sequence of control states and transitions $p_0t_1p_1t_2\cdots t_kp_k$ with $k \leq |Q|$ such that*

(i) $p_{i-1} = src(t_i)$ *and* $p_i \in trg(t_i)$ *for all* $1 \leq i < k$;
(ii) $p_k = p_j$ *for some* $j < k$ *and* $p_i \neq p_j$ *for all* $i < j$;
(iii) $p(0)$ *is coverable for every* $p \in \bigcup_{i=1}^{k-1} trg(t_i)$; *and*
(iv) for every $j < i \leq k$, *there exists* $n_i \leq |Q|$ *such that* $n_i = 0$ *if* $t_i \notin Q^3$; $p_i'(n_i)$ *is coverable for* $p_i' \in trg(t_i) \setminus p_i$; *and* $\sum_{j<i\leq k} n_i > \sum_{j<i\leq k} eff(t_i)$.

it's a branch in the tree,
first repetition,

Proof. If q is unbounded, then by Lemma 18 we can take a reachability tree containing a short decreasing node v that provides a sequence as above. Notice that since v is decreasing, the ~~the~~ combined effect of all transitions used on the branch from the root to v is bounded by $|v| \leq |Q|$, and must be less than the sum of the counter values of the direct siblings $ub \not\preceq v$, where $u \preceq v$. This guarantees the existence of suitable n_i and thus ensures Condition (iv).

For the converse direction, assume a sequence as above. Conditions (i)-(iii) imply the existence of some reachability tree for some $p_0(n)$. Moreover, Condition (iv) ensures that there is such a tree with decreasing node. We conclude by Lemma 17. ◀

It is easily seen that Lemma 17 provides a template that directly translates into an AL-algorithm, analogue to Algorithm 1, which yields the P upper bound for boundedness. In particular, observe that the witnessing sequence satisfying Conditions (i) and (ii), as well as the numbers $n_i \leq |Q|$ can be guessed non-deterministically in logarithmic space. Moreover, Conditions (iii) and (iv) are decidable in polynomial time by Theorem 16.

▶ **Theorem 20.** *Boundedness in $BVASS_1$ is* P*-complete.*

6 Conclusion

We showed that reachability, coverability and boundedness in BVASS_1 are all P-complete and thereby established the first decidability result for reachability in a subclass of BVASS. This low complexity is quite surprising since the general reachability problem for BVASS is at least non-elementary [10] and there exist families of instances of BVASS_1-reachability problems whose reachability trees contain an exponential number of distinct counter values, cf. Section 3. The approach developed in this paper shows that it is not necessary to explicitly construct a full reachability tree in order to witness reachability. In fact, we showed in Section 4 that the existence of so-called residue and expandable reachability trees suffices in order to decide reachability and can be witnessed in polynomial time.

Our approach is quite specific to having only one counter available in BVASS_1 and does not seem to immediately generalise to higher dimensions. Nevertheless, we believe that this paper spreads some optimism and provides sufficient evidence that obtaining results for reachability in general BVASS is not impossible.

References

1 A.K. Chandra, D. Kozen, and L.J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.
2 Ph. de Groote, B. Guillaume, and S. Salvati. Vector addition tree automata. In *Logic in Computer Science, LICS*, pages 64–73. IEEE Computer Society, 2004.
3 S. Demri, M. Jurdziński, O. Lachish, and R. Lazić. The covering and boundedness problems for branching vector addition systems. *J. Comput. Syst. Sci.*, 79(1):23–38, 2013.
4 M. Englert, R. Lazić, and P. Totzke. Reachability in two-dimensional unary vector addition systems with states is NL-complete. Unpublished manuscript, 2016.
5 P. Ganty and R. Majumdar. Algorithmic verification of asynchronous programs. *ACM Trans. Program. Lang. Syst.*, 34(1):6, 2012.
6 S.M. German and A.P. Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992.
7 R. Greenlaw, H.J. Hoover, and W.L. Ruzzo. *Limits to parallel computation: P-completeness theory*. Oxford University Press, 1995.
8 P. Jančar and Z. Sawa. A note on emptiness for alternating finite automata with a one-letter alphabet. *Inf. Process. Lett.*, 104(5):164–167, 2007.
9 A. Jez and A. Okhotin. Conjunctive grammars over a unary alphabet: Undecidability and unbounded growth. *Theory Comput. Syst.*, 46(1):27–58, 2010.
10 R. Lazić and S. Schmitz. Nonelementary complexities for branching VASS, MELL, and Extensions. *ACM Trans. Comput. Log.*, 16(3):20, 2015.
11 J. Leroux and S. Schmitz. Demystifying reachability in vector addition systems. In *Logic in Computer Science, LICS*, pages 56–67. IEEE, 2015.
12 R.J. Lipton. The reachability problem requires exponential space. *Yale University*, Technical Report 62, 1976.
13 C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
14 C. Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6:223–231, 1978.
15 O. Serre. Parity games played on transition graphs of one-counter processes. In *Foundations of Software Science and Computation Structures, FOSSACS*, pages 337–351, 2006.
16 L.G. Valiant and M. Paterson. Deterministic one-counter automata. *J. Comput. Syst. Sci.*, 10(3):340–350, 1975.

A Missing Proofs

A.1 Missing Proofs from Section 3

An instance of MCVP is a Boolean circuit $\mathcal{C}$ consisting of n gates $g_1, \dots, g_n$ such that for all $k \in [1, n]$ either $g_k = \top$, $g_k = \bot$ or there are $1 \leq i, j < k$ such that $g_k = g_i \vee g_j$ or $g_k = g_i \wedge g_j$. MCVP is to decide whether $\mathcal{C}$ evaluates to true, i.e. if g_n evaluates to true. We note that MCVP is the canonical P-complete problem [7]. The following proposition gives the lower bound for Theorem 2.

▶ **Proposition 3.** *Let $\mathcal{C}$ be a Boolean circuit. There exists a logspace computable BVASS$_1$ $\mathcal{B}$ with a control state q such that $q(0)$ is reachable if, and only if, $\mathcal{C}$ evaluates to true.*

Proof. From $\mathcal{C}$ we derive a BVASS$_1$ $\mathcal{B} \stackrel{\text{def}}{=} (Q, \Delta, F)$, where $Q \stackrel{\text{def}}{=} \{q_1, \dots, q_n\}$, $F \stackrel{\text{def}}{=} \{q_i \mid g_i = \top\}$ and $\Delta \stackrel{\text{def}}{=} \{(q_k, q_i, q_j) \mid g_k = g_i \wedge g_j\} \cup \{(g_k, 0, g_i), (g_k, 0, g_j) \mid g_k = g_i \vee g_j\}$. Hence, $\wedge$-gates are simulated by splits and $\vee$-gates by non-deterministic branching. It is easily seen that g_n evaluates to true if, and only if, $q_k(0)$ is reachable in $\mathcal{B}$. ◀

▶ **Proposition 4.** *Reachability in BVASS$_1$ is* NP*-hard if the initial configuration $q(n)$ is given in binary.*

Proof. We first show that for any $m \in \mathbb{N}$ given in binary, we can in logarithmic space extend $\mathcal{B}_n$ constructed above with a control state q_m such that $\mathit{reach}(q_m) = \{m\}$. Let $m = \sum_{0 \leq i \leq n} b_i \cdot 2^i$ with $b_i \in \{0, 1\}$ be the binary representation of m. We introduce additional fresh control states q^i_m, $0 \leq i \leq n$, transitions $(q_m, 0, q^n_m)$ and (q^0_m, q_0, q_f), and for every $1 \leq i \leq n$ transitions $(q^i_m, 0, q^{i-1}_m)$ if $b_i = 0$ and (q^i_m, q_i, q^{i-1}_m) if $b_i = 1$. It is easily checked that $\mathit{reach}(q_m) = \{m\}$.

In order to show hardness for NP, we reduce from the problem Subset Sum. Given a finite set $S = \{m_1, \dots, m_k\} \subseteq \mathbb{N}$ and $t \in \mathbb{N}$ with all numbers encoded in binary, Subset Sum is the problem to decide whether there are $c_1, \dots, c_k \in \{0, 1\}$ such that $t = \sum_{1 \leq i \leq k} c_i \cdot m_i$ and is known to be NP-complete [13]. As shown above, we can construct a BVASS $\mathcal{B}$ with control states $q_{m_1}, \dots, q_{m_k}$ such that $\mathit{reach}(q_{m_i}) = \{m_i\}$. We introduce additional fresh control states $q_{c_1}, \dots, q_{c_k}$ that allow us to non-deterministically make a choice for every c_i by introducing for every $1 \leq i < k$ transitions $(q_{c_i}, 0, q_{c_{i+1}})$ and $(q_{c_i}, q_{m_i}, q_{c_{i+1}})$. It is now easily seen that the instance (S, t) of Subset Sum is valid if, and only if, $q_{c_1}(t)$ is reachable. ◀

▶ **Corollary 21.** *Reachability in BVASS$_2$ is* NP*-hard.*

Proof (sketch). The statement follows from an easy adaption of the proof of Proposition 4. It suffices to show how to construct a BVASS$_2$ that reaches the control state q_{c_1} from Proposition 4 with counter values $(t, 0)$. But this can easily be achieved by first adding a non-deterministic number of times $(1, 1)$ to the counter and then by branching into the control states q_{c_1} and q_t, where q_t is suitably adjusted such that $\mathit{reach}(q_t) = \{(0, t)\}$. ◀

A.2 Missing Proofs from Section 4.1

▶ **Lemma 6.** *The set S is computable in polynomial time.*

Proof. We note that N is polynomially bounded in $|\mathcal{B}| + |d|$. Moreover, $S \subseteq Q \times [0, n + N]$ and S can be computed in polynomial time by using a dynamic programming approach. ◀

▶ **Lemma 7.** *The fixed point* $R \stackrel{def}{=} \bigcup_{i\geq 0} R_i$ *equals* R_N *and is computable in polynomial time.*

Proof. Analogously to the computation of S in Lemma 6, one shows that R_0 is computable in polynomial time. To see that $R = R_N$, note that by definition we have $R_i \subseteq R_{i+1}$ for all $i \in \mathbb{N}$. If $R_i \subset R_{i+1}$, there is at least one pair from $Q \times \mathbb{Z}_d$ that is in R_{i+1} and not in R_i. Since there are at most N such pairs, the sequence stabilises after at most N steps at R_N. Since N is polynomial in $|\mathcal{B}|+d$, consequently R_N can also be computed in polynomial time. ◀

▶ **Lemma 8.** *The set* $X \stackrel{def}{=} R \cup S[n_0, n_0+N]/\mathbb{Z}_d$ *is computable in polynomial time. Moreover,*

$$X = \{(q, n \bmod d) \mid q \in Q, n \in \mathit{reach}(q), n \geq n_0\}.$$

Proof. Polynomial-time computability of X follows immediately from the polynomial time computability of S (Lemma 6) and of R (Lemma 7). It thus remains to prove that $X = \{(q, n \bmod d) \mid q \in Q, n \in \mathit{reach}(q), n \geq n_0\}$.

("$\subseteq$") Trivially, $S[n_0, n_0+N]/\mathbb{Z}_d \subseteq \{(q, n \bmod d) \mid q \in Q, n \in \mathit{reach}(q), n \geq n_0\}$ since $S \subseteq \{(q, n) \mid q \in Q, n \in \mathit{reach}(q)\}$. Hence it remains to show that R is contained in $\{(q, n \bmod d) \mid q \in Q, n \in \mathit{reach}(q), n \geq n_0\}$.

To prove this, we show that for every $i \in [0, N]$ and each $(q, r) \in R_i$ there exists some $n \in \mathit{reach}(q)$ with $n \geq n_0 + N - i$ and $n \equiv r \bmod d$ by induction on i. We note that this is sufficient to prove since $R = R_N$ and thus for each $(q, r) \in R$ there exists some $n \in \mathit{reach}(q)$ with $n \geq n_0$ and $n \equiv r \bmod d$.

For the induction base, i.e. $i = 0$, we recall that for each $(q, r) \in R_0$ there exists some $n \geq n_0 + N = n_0 + N - i$ such that $n \equiv r \bmod d$ and there is some almost $(n_0 + N)$-bounded reachability tree whose root is labelled with $q(n)$ by definition of R_0; in particular $n \in \mathit{reach}(q)$.

For the induction step, let $i+1 \leq N$ and let us assume $(q, r) \in R_{i+1}$. If already $(q, r) \in R_i$ then (q, r) satisfies the desired property immediately by induction hypothesis. Otherwise, if $(q, r) \in \Delta(R_i)$ then $r \equiv r' + z \bmod d$ for some $(q', r') \in R_i$ and some $(q, z, q') \in \Delta$. By induction hypothesis there exists some $n' \in \mathit{reach}(q')$ with $n' \geq n_0 + N - i$ and $n' \equiv r' \bmod d$. For $n = n' + z$, we have $n \equiv r' + z \equiv r \bmod d$ and since $n \geq n' - |z| \geq n_0 + N - i - |z| \geq n_0 + N - (i+1) \geq n_0 \geq 0$ it follows $n \in \mathit{reach}(q)$.

It remains to consider the case when $(q, r) \in \Delta(R_i, S/\mathbb{Z}_d) \cup \Delta(S/\mathbb{Z}_d, R_i) \cup \Delta(R_i, R_i)$. We only treat the case $(q, r) \in \Delta(R_i, S/\mathbb{Z}_d)$, the other cases can be proven analogously. In this case we have $r \equiv r' + n'' \bmod d$ for some $(p', r') \in R_i$ and some $(p'', n'') \in S$, where $(q, p', p'') \in \Delta$. Clearly, $n'' \in \mathit{reach}(p'')$ by definition of S. By the induction hypothesis, there exists some $n' \geq n_0 + N - i$ such that $n' \in \mathit{reach}(p')$ and $n' \equiv n \bmod d$. Let $n = n' + n''$. Hence $n \in \mathit{reach}(q)$, since $n' \in \mathit{reach}(p')$ and $n'' \in \mathit{reach}(p'')$. Obviously $n \equiv r \bmod d$ and, finally, $n \geq n' \geq n_0 + N - i \geq n_0 + N - (i+1)$.

("$\supseteq$") Assume some $q(n)$ is reachable for some $n \geq n_0$. We prove that $(q, n \bmod d) \in X$. To this end, let us fix some reachability tree $T \colon U \to Q \times \mathbb{N}$ for $q(n)$. If T is (n_0+N)-bounded it follows that $(q, n) \in S$ and we are done since $n_0 \leq n \leq n_0 + N$.

Consequently, let us assume that T is not $(n_0 + N)$-bounded. First, observe that $T(u) \in F \times \{0\} \subseteq S$ for all leaves $u \in U$. In addition, the set of nodes $V \stackrel{\text{def}}{=} \{u \in U \mid T(u) \notin S\}$ is non-empty for otherwise T would be $(n_0 + N)$-bounded. Moreover, V is prefix-closed and note that every $\preceq$-maximal node v in V satisfies $T(v)/\mathbb{Z}_d \in R_0$ by the choice of V. For every node $v \in V$ that is not $\preceq$-maximal, we either have $T(v)/\mathbb{Z}_d \in \Delta(T(v0)/\mathbb{Z}_d)$ (if $v0$ is the only child of v) or $T(v)/\mathbb{Z}_d \in \Delta(T(v0)/\mathbb{Z}_d, T(v1)/\mathbb{Z}_d)$ (if v has two children $v0$ and $v1$).

Moreover, note that $\Delta(R) \subseteq R$ and $\Delta(R, R) \subseteq R$. This shows that $T/\mathbb{Z}_d(V)$ is contained in R, in particular $T/\mathbb{Z}_d(\varepsilon) \in R$ which proves $(q, n \bmod d) \in R \subseteq X$. ◀

A.3 Missing Proofs from Section 4.2

▶ **Lemma 11.** *Let u and v be nodes of a partial reachability tree such that $u \prec v$ and $counter(u) + |Q| \leq counter(v)$. Then there exists a pumping node v' with anchor u' such that $u \preceq u' \prec v' \preceq v$.*

Proof. The counter value of a node exceeds that of its parent by at most one. Consequently, for every $counter(u) \leq i \leq counter(v)$ there is a node $u \preceq u_i \preceq v$ with $counter(c_i) = i$ and $u_j \prec u_k$ for all $counter(u) \leq j < k \leq counter(v)$. Since $counter(v) - counter(u) \geq |Q|$ by assumption, by the pigeonhole principle there exist some $counter(u) \leq j < k \leq counter(v)$ such that $state(u_j) = state(u_k)$; these yield the required u' and v'. ◀

▶ **Lemma 13.** *If u is a node of an exclusive partial reachability tree then the following hold:*

(i) If u is the anchor of a pumping leaf v then u is exclusive and all nodes w such that $u \prec w \preceq v$ are not exclusive.
(ii) u has at most one child that is not exclusive.

Proof. Regarding (i), every anchor u' of every other pumping leaf v' of $T^{\downarrow u}$ is a strict descendant of u, since otherwise $u, u' \preceq lca(v, v')$, contradicting T being exclusive. Consequently, T being exclusive implies u being exclusive. Moreover, for w such that $u \prec w \preceq v$, due to an anchor being maximal, v has no anchor on the subtree rooted at any such w and hence w is not exclusive.

Regarding (ii), suppose $u0$ and $u1$ are both not exclusive. If all pumping leaves of $T^{\downarrow u0}$ had their anchors in $T^{\downarrow u0}$, then $u0$ would be exclusive. Hence there is some pumping leaf v_0 in $T^{\downarrow u0}$ with anchor u_0 such that $u_0 \preceq u$. Likewise, we find a pumping leaf v_1 with anchor u_1 such that $u_1 \preceq u$ in $T^{\downarrow u1}$. But then $u_0, u_1 \preceq lca(v_0, v_1)$ and hence T is not exclusive, a contradiction. ◀

A.4 Missing Proofs from Section 5

▶ **Lemma 17.** *If a reachability tree T with $T(\varepsilon) = q(n)$ contains a decreasing node then q is unbounded.*

Proof. It suffices to observe that one can unfold the cyclic suffix of a decreasing node v by replacing the subtree rooted in v by that one rooted in $u \prec v$. This construction is analogous to the construction in the proof of Lemma 14, with the only difference that the effect of the cycle is negative here. The result of such an operation is a reachability tree whose root is labelled with a configuration that has the same control state and whose counter value is strictly increased. Moreover, this reachability tree still contains a decreasing node. Such an unfolding can therefore be repeated arbitrarily often, from which the claim follows. ◀

▶ **Lemma 18.** *Suppose $n > 2^{|Q|}$ with $n \in reach(q)$. There exists a reachability tree $T \colon U \to Q \times \mathbb{N}$ for $q(n')$ where $n' \geq n$, and which contains a decreasing node v with $|v| \leq |Q|$.*

Proof. In any reachability tree it is possible to collapse the part between two nodes $u \prec v$ if $state(u) = state(v)$ and $counter(u) \leq counter(v)$, that is, to replace the subtree rooted in u by the one rooted in v. The result of this is a reachability tree with fewer nodes and where the root has the same state and a counter value at least as large as in the original tree.

Thus, we may assume with no loss of generality a reachability tree T with root $T(\varepsilon) = q(n)$ for $n \geq 2^{|Q|}$ and such that for any two nodes $u \prec v$ with $\mathit{state}(u) = \mathit{state}(v)$, it holds that $\mathit{counter}(u) > \mathit{counter}(v)$.

In order to find a decreasing node, we move from the root downwards, always choosing the successor with the largest counter value. This way, the counter value of a chosen node is at least half as large as the counter of its parent. Since the value in the root is greater or equal to $2^{|Q|}$, this means that the produced sequence is longer than $|Q|$. In particular, the prefix of length $|Q|$ must contain a decreasing node. ◀