HAL
open science

The Ideal View on Rackoff's Coverability Technique

Ranko Lazić, Sylvain Schmitz

To cite this version:

Ranko Lazić, Sylvain Schmitz. The Ideal View on Rackoff's Coverability Technique. RP 2015 - 9th International Workshop on Reachability Problems, Sep 2015, Warsaw, Poland. pp.1–13, 10.1007/978-3-319-24537-9_8 . hal-01176755v2

HAL Id: hal-01176755

https://inria.hal.science/hal-01176755v2

Submitted on 2 Feb 2016 (v2), last revised 21 Jun 2017 (v3)

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

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

Copyright

THE IDEAL VIEW ON RACKOFF'S COVERABILITY TECHNIQUE

RANKO LAZIĆ AND SYLVAIN SCHMITZ

Abstract. Rackoff's small witness property for the coverability problem is the standard means to prove tight upper bounds in vector addition systems (VAS) and many extensions. We show how to derive the same bounds directly on the computations of the VAS instantiation of the generic backward coverability algorithm. This relies on a dual view of the algorithm using ideal decompositions of downwards-closed sets, which exhibits a key structural invariant in the VAS case. This reasoning generalises uniformly to several VAS extensions: we derive optimal upper bounds for coverability in branching and alternating VAS, matching the previously known results from the literature.

1. Introduction

Checking safety properties in infinite transition systems can often be reduced to *coverability* checks. The coverability problem asks, given a transition system and two configurations x and y and a quasi-ordering $\leq$ over configurations, whether x might *cover* y, i.e. reach some configuration $y' \geq y$ in finitely many steps. The problem is decidable for the large class of (effective) *well-structured transition systems* (WSTS) where $\leq$ is a *well-quasi-ordering* (wqo) compatible with the transition relation [1, 13]. The algorithm to that end is a generic *backward coverability* procedure, which computes successively the sets of configurations that can cover y in at most 0, 1, 2, ... steps. Those sets are upwards-closed and since $\leq$ is a wqo they can be represented through their finitely many minimal elements.

Nevertheless, the naive complexity upper bounds one can extract directly from the termination argument of the backward coverability algorithm—which also relies on $\leq$ being a wqo—are sometimes very far from the optimal ones. A striking illustration is provided by vector addition systems (VAS): the complexity bounds offered e.g. by [11] are in Ackermann, whereas coverability in VAS has long been known to be ExpSpace-complete thanks to a lower bound by Lipton [20] and an upper bound by Rackoff [22].

Rackoff's Technique is essentially combinatorial in nature: he shows by induction on the dimension of the VAS that, if x can reach one such $y' \geq y$, then there exists a small (doubly-exponential) run in the VAS witnessing this fact. A non-deterministic algorithm can then simply look for such a

An extended abstract of this work was published in the Proceedings of the 9th International Workshop on Reachability Problems, Lecture Notes in Computer Science vol. 9328 pp. 76–88 (M. Bojańczyk, S. Lasota, and I. Potapov, Eds.), Springer, 2015.
Work funded in part by the Leverhulme Trust Visiting Professorship VP1-2014-041 and the EPSRC grant EP/M011801/1.

witness using only exponential space. The same general technique has since been extended to prove tight complexity upper bounds for coverability in numerous extensions of VASs [10, 6, 8, 17, 16]. It is however less clear how to adapt the technique for more general systems, where for instance the notion of dimension is absent or more involved.

Remarkably, Bozzelli and Ganty [7] showed that Rackoff's small witness property can be applied to the backward coverability algorithm for VAS to obtain a 2ExpTime upper bound.[1] However, their proof uses Rackoff's analysis as a black box, and does not work directly with the structures manipulated by the backward coverability algorithm. As such, it is again unclear how this result could be translated to further classes of well-structured transition systems.

Contributions. In this paper, we revisit the backward coverability algorithm for VAS, and extract directly a 2ExpTime upper bound for its running time. We take for this in Section 3 a dual view on the backward coverability algorithm, by considering successively the sets of configurations that do *not* cover y in 0, 1, 2, ... or fewer steps. Such sets are downwards-closed, and enjoy a (usually effective) canonical representation as finite unions of *ideals* [5, 12, 15]. We show in Section 4 that, in the case of VAS, this dual view exhibits an additional structural property of *ω-monotonicity*, which allows to derive the desired doubly-exponential bound.

Our purpose is above all pedagogical, as we hope to see this type of reasoning applied more broadly where the simple proof argument of Rackoff fails. As illustrations of the versatility of the framework, we consider in Section 5 and Section 6 the top-down and bottom-up coverability problems in *alternating branching* VAS. In each case, we provide an instance of the generic backward algorithm that solves the problem, and show that its running time matches the known optimal complexities [10, 8, 17]. In contrast to the upper bound proofs in the literature, which employ ad-hoc adaptations of Rackoff's technique, the proofs we obtain are uniform extensions from the basic case of VAS, so they proceed by analysing the downwards-closed sets that the backward algorithms produce. For top-down coverability, the ideals involved are the same as for VAS (vector ideals), whereas bottom-up coverability requires us to work one level higher (vector set ideals).

We start with some preliminaries on WSTS and ideals in Section 2.

2. Preliminaries

We first recall the necessary background on well-quasi-orders, well-structured transition systems, and ideal decompositions, while illustrating systematically the definitions on VAS and reset VAS.

2.1. **Well-Structured Transition Systems.** A *well-quasi-order* (wqo) $(X, \leq)$ is a set X equipped with a transitive reflexive relation $\leq$ such that, along any infinite sequence $x_0, x_1, \ldots$ of elements from X, one can find two indices $i < j$ such that $x_i \leq x_j$. A finite or infinite sequence without such

[1] In the same spirit, Majumdar and Wang [21] show that the 'expand, enlarge, and check' algorithm for bottom-up coverability in branching VASs runs in 2ExpTime, using the combinatorial analysis of Demri et al. [10].

pair of indices is *bad*, and necessarily finite over a wqo. See for instance [26] for more background on wqos.

Example 2.1 (Dickson's Lemma)**.** The set $\mathbb{N}^d$ of d-dimensional vectors of natural numbers forms a wqo when endowed with the product ordering $\sqsubseteq$, defined by $\boldsymbol{u} \sqsubseteq \boldsymbol{v}$ if $\boldsymbol{u}(i) \leq \boldsymbol{v}(i)$ for all $1 \leq i \leq d$.

A *well-structured transition system* (WSTS) [1, 13] is a triple $(X, \rightarrow, \leq)$ where X is a set of configurations, $\rightarrow \subseteq X \times X$ is a transition relation, and $(X, \leq)$ is a wqo with the following *compatibility* condition: if $x \leq x'$ and $x \rightarrow y$, then there exists $y' \geq y$ with $x' \rightarrow y'$. In other words, $\leq$ is a simulation relation on the transition system $(X, \rightarrow)$. We write as usual $\rightarrow^{\leq 0} \stackrel{\text{def}}{=} \{(x, x) \mid x \in X\}$ and $\rightarrow^{\leq k+1} \stackrel{\text{def}}{=} \rightarrow^{\leq k} \cup \{(x, y) \mid \exists z \in X \,.\, x \rightarrow z \rightarrow^{\leq k} y\}$ for the reachability relation in at most k steps, and $\rightarrow^* \stackrel{\text{def}}{=} \bigcup_k \rightarrow^{\leq k}$ for the reflexive transitive closure of $\rightarrow$.

Example 2.2 (VAS are WSTS)**.** A d-dimensional *vector addition system* (VAS) is a finite set $\boldsymbol{A}$ of vectors in $\mathbb{Z}^d$. It defines a WSTS $(\mathbb{N}^d, \rightarrow, \sqsubseteq)$ with configurations space $\mathbb{N}^d$ and $\boldsymbol{u} \rightarrow \boldsymbol{u} + \boldsymbol{a}$ for all $\boldsymbol{u}$ in $\mathbb{N}^d$ and $\boldsymbol{a}$ in $\boldsymbol{A}$ such that $\boldsymbol{u} + \boldsymbol{a}$ is in $\mathbb{N}^d$.

For instance, the 2-dimensional VAS $\boldsymbol{A}_{\div 2} = \{(-2, 1)\}$ can be seen as weakly computing the halving function: from any configuration $(n, 0)$, it can reach $(n \bmod 2, \lfloor n/2 \rfloor)$ and all its reachable configurations (n', m) satisfy $m \leq n/2$.

Example 2.3 (Reset VAS are WSTS)**.** A d-dimensional *reset VAS* is a finite subset $\boldsymbol{A}$ of $\mathbb{Z}^d \times \mathbb{P}(\{1, \ldots, d\})$. Given $R \subseteq \{1, \ldots, d\}$ and a vector $\boldsymbol{u}$, we define the vector $R(\boldsymbol{u})$ by $R(\boldsymbol{u})(i) = 0$ if $i \in R$, and $R(\boldsymbol{u})(i) = \boldsymbol{u}(i)$ otherwise. A reset VAS defines a WSTS $(\mathbb{N}^d, \rightarrow, \sqsubseteq)$ where $\boldsymbol{u} \rightarrow R(\boldsymbol{u} + \boldsymbol{a})$ if there exists $(\boldsymbol{a}, R)$ in $\boldsymbol{A}$ such that $\boldsymbol{u} + \boldsymbol{a}$ is in $\mathbb{N}^d$.

For instance, the 5-dimensional reset VAS

$$\boldsymbol{A}_{\log} = \left\{ \begin{array}{c} (0, 0, -2, 1, 0, \emptyset), (0, 0, 1, -1, 0, \emptyset), \\ (-1, 1, -2, 1, 0, \{3\}), (1, -1, 1, -1, 1, \{4\}) \end{array} \right\}$$

is a weak computer for the logarithm function: from any configuration of the form $(1, 0, 2^n, 0, 0)$, it can reach $(1, 0, 1, 0, n)$, and all its reachable configurations of the form $(1, 0, n', m, \ell)$ satisfy $\ell \leq n$.

2.2. **Ideal Decompositions.** The *downward-closure* of a subset $S \subseteq X$ over a wqo $(X, \leq)$ is $\downarrow X \stackrel{\text{def}}{=} \{x \in X \mid \exists s \in S \,.\, x \leq s\}$. A subset $D \subseteq X$ is *downwards-closed* if $\downarrow D = D$. We write $\downarrow x$ for the downward-closure of the singleton set $\{x\}$. Upward-closures and upwards-closed sets $U = \uparrow U$ are defined similarly.

Well-quasi-orders can also be characterised by the *descending chain condition*: a quasi-order $(X, \leq)$ is a wqo if and only if every descending sequence $D_0 \supsetneq D_1 \supsetneq D_2 \supsetneq \cdots$ of downwards-closed subsets $D_i \subseteq X$ is finite.

An *ideal* of X is a non-empty downwards-closed subset $I \subseteq X$, which is *directed*: if x, x' are two elements of I, then there exists y in I with $x \leq y$ and $x' \leq y$. Over a wqo $(X, \leq)$, any downwards-closed set $D \subseteq X$ has a unique *decomposition* as a finite union of ideals $D = I_1 \cup \cdots \cup I_n$, where the I_j's are mutually incomparable for inclusion [5, 12]. Alternatively, ideals are

characterised as *irreducible* downwards-closed sets: an ideal is a non-empty downwards-closed set I with the property that, if $I \subseteq D_1 \cup D_2$ for two downwards-closed sets D_1 and D_2, then $I \subseteq D_1$ or $I \subseteq D_2$.

Example 2.4 (Vector Ideals)**.** Over $(\mathbb{N}^d, \sqsubseteq)$, observe that $\downarrow\boldsymbol{u}$ is an ideal for every $\boldsymbol{u}$ in $\mathbb{N}^d$. Those are however not the only ideals, e.g. $I \stackrel{\text{def}}{=} \{(0,n,0) \mid n \in \mathbb{N}\}$ is also an ideal of $\mathbb{N}^3$. Write $\mathbb{N}_\omega \stackrel{\text{def}}{=} \mathbb{N} \uplus \{\omega\}$ where ω is a new top element; the product ordering $\sqsubseteq$ extends naturally to $\mathbb{N}_\omega^d$. Then the ideals of $(\mathbb{N}^d, \sqsubseteq)$ are exactly the downward-closures $\downarrow\boldsymbol{u}$ inside $\mathbb{N}^d$ of vectors $\boldsymbol{u}$ from $\mathbb{N}_\omega^d$. For the previous example in $\mathbb{N}^3$, $\downarrow(0,\omega,0) = I$.

Although ideals provide finite representations for manipulating downwards-closed sets, some additional effectiveness assumptions are necessary to employ them in algorithms. In this paper, we will say that a wqo $(X, \leq)$ has *effective* ideal representations [see 12, 15, for more stringent requisites] if every ideal can be represented, and there are algorithms on those representations:

- **(CI):** to check $I \subseteq I'$ for two ideals I and I',
- **(II):** to compute the ideal decomposition of $I \cap I'$ for two ideals I and I',
- **(CU'):** to compute the ideal decomposition of the residual $X \setminus \uparrow x = \{x' \in X \mid x \not\leq x'\}$ for any x in X.

Example 2.5 (Effective Representations of Vector Ideals)**.** We shall use vectors in $\mathbb{N}_\omega^d$ as representations. For (CI), given two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{N}_\omega^d$, $\downarrow\boldsymbol{u} \subseteq \downarrow\boldsymbol{v}$ if and only if $\boldsymbol{u} \sqsubseteq \boldsymbol{v}$. Furthermore, for (II), $\downarrow\boldsymbol{u} \cap \downarrow\boldsymbol{v} = \downarrow\boldsymbol{w}$ where $\boldsymbol{w}(i) \stackrel{\text{def}}{=} \min_{\leq}(\boldsymbol{u}(i), \boldsymbol{v}(i))$ for all $1 \leq i \leq d$. Finally, for (CU'), if $\boldsymbol{u}$ is in $\mathbb{N}^d$, then $\mathbb{N}^d \setminus \uparrow\boldsymbol{u} = \bigcup_{1 \leq j \leq d \,|\, \boldsymbol{u}(j) > 0} \downarrow\boldsymbol{u}_{/j}$ where $\boldsymbol{u}_{/j}(i) = \omega$ if $i \neq j$ and $\boldsymbol{u}_{/j}(j) \stackrel{\text{def}}{=} \boldsymbol{u}(j) - 1$ otherwise.

Crucially for the applicability of our approach, effective ideal representations exist for most wqos of interest [12, 15].

3. Backward Coverability

Let us recall in this section the generic backward coverability algorithm for well-structured transition systems [1, 13]—the first published instance we know of this algorithm was incidentally for the case of reset VASS [3].

We take a dual view on this algorithm, by considering downwards-closed sets represented through their ideal decompositions, instead of the usual view using upwards-closed sets represented through their minimal elements. We present the generic algorithm, but will illustrate all the notions using the case of VAS and reset VAS in Section 3.2, and derive naive upper bounds for both in Section 3.3—which will turn out to be optimal for reset VAS.

3.1. Generic Algorithm.

Consider a WSTS $(X, \rightarrow, \leq)$ and a target configuration y from X to be covered. Define $D_* \stackrel{\text{def}}{=} \{x \in X \mid \forall y' \geq y \,.\, x \not\rightarrow^* y'\}$ as the set of configurations that do not cover y. The purpose of the backward coverability algorithm is to compute D_*; solving a coverability instance with source configuration x_0 then amounts to checking whether x_0 belongs to D_*.

The idea of the algorithm is to compute successively for every k the set D_k of configurations that do *not* cover y in k steps or fewer:

$$D_* = \bigcap_k D_k \ , \qquad\qquad D_k \stackrel{\text{def}}{=} \{x \in X \mid \forall y' \geq y \,.\, x \not\to^{\leq k} y'\} \ . \tag{1}$$

These over-approximations D_k can be computed inductively on k by

$$D_0 = X/y \ , \qquad\qquad D_{k+1} = D_k \cap \mathrm{Pre}_\forall(D_k) \ , \tag{2}$$

where for any set $S \subseteq X$,

$$\mathrm{Pre}_\forall(S) \stackrel{\text{def}}{=} \{x \in X \mid \forall z \in X \,.\, (x \to z \implies z \in S)\} \ . \tag{3}$$

The algorithm terminates as soon as $D_k \subseteq D_{k+1}$ (and thus $D_{k+j} = D_k = D_*$ for all j). This is guaranteed to arise eventually by the descending chain condition, since otherwise we would have an infinite descending chain of downwards-closed sets $D_0 \supsetneq D_1 \supsetneq D_2 \supsetneq \cdots$.

3.1.1. *Correctness.* The correctness of the algorithm hinges on the following claim:

Claim 3.1 (Correctness). Equations (1) and (2) define the same D_k.

Proof. By induction on k. For the base case, $x \not\to^{\leq 0} y'$ for all $y' \geq y$, if and only if $x \not\geq y$, i.e. if and only if x is in X/y. For the induction step and for all $y' \geq y$, $x \not\to^{\leq k+1} y'$ if and only if $x \not\to^{\leq k} y'$ and there does not exist any z with $x \to z$ and $z \to^{\leq k} y'$. The former is equivalent to x belonging to D_k by induction hypothesis. The latter occurs if and only if for all z in X, if $x \to z$ then $z \not\to^{\leq k} y'$, i.e. if and only if x belongs to $\mathrm{Pre}_\forall(D_k)$ by induction hypothesis. □

3.1.2. *Effective Ideal Representations.* The algorithm as presented above relies on the effectiveness of Eq. (2). We are going to use effective representations of the ideal decompositions of the D_k to this end. Let us first check that we are indeed dealing with downwards-closed sets:

Claim 3.2 (Downward-closure). For all k, D_k is downwards-closed.

Proof. By induction on k. For the base case, $D_0 = X/y$ is downwards-closed. For the induction step, first observe that, if D is downwards-closed, then $\mathrm{Pre}_\forall(D)$ is also downwards-closed. Indeed, let $x \leq x'$ for some x' in $\mathrm{Pre}_\forall(D)$. Consider any z such that $x \to z$. Then by WSTS compatibility, there exists $z' \geq z$ such that $x' \to z'$. Since x' belongs to $\mathrm{Pre}_\forall(D)$, z' belongs to D. Because D is downwards-closed, z also belongs to D. This shows x in $\mathrm{Pre}_\forall(D)$ as desired. We conclude by noting that downwards-closed sets are closed under intersection, hence $D_{k+1} = D_k \cap \mathrm{Pre}_\forall(D_k)$ is downwards-closed by applying the induction hypothesis to D_k. □

The only additional effectiveness assumption we make is that:

(Pre): the ideal decomposition of $\mathrm{Pre}_\forall(D)$ is computable for all downwards-closed D.

This is sufficient to compute the ideal decompositions of all the D_k. Indeed, initially D_0 is computed using (CU'). Later, $\mathrm{Pre}_\forall(D_k)$ is computable by (Pre), and its intersection with D_k is also computable by (II). Finally, recall that, by ideal irreducibility, $I_1 \cup \cdots \cup I_n \subseteq J_1 \cup \cdots \cup J_m$ for ideals $I_1, \ldots, I_n$ and downwards-closed $J_1, \ldots, J_m$ if and only if for all $1 \leq i \leq n$ there exists $1 \leq j \leq m$ such that $I_i \subseteq J_j$. Therefore, the termination check $D_k \subseteq D_{k+1}$ is effective by (CI).

Remark 3.3 (Existential Predecessors). The classical presentation of the backward coverability algorithm works with upwards-closed sets U, which are represented through their finitely many minimal elements in $\min U$. The assumption (Pre) is therefore replaced by the computability of a set of minimal elements for

$$\mathrm{Pre}_\exists(U) \stackrel{\text{def}}{=} \{x \in X \mid \exists z \in U \,.\, x \to z\} \tag{4}$$

for an upwards-closed U.

It turns out that, under the usual effectiveness assumptions on $(X, \leq_X)$ plus the ideal effectiveness assumptions, the two assumptions on the computability of $\mathrm{Pre}_\forall$ and $\mathrm{Pre}_\exists$ are equivalent. Indeed, we have the duality

$$\mathrm{Pre}_\forall(D) = X \setminus \big(\mathrm{Pre}_\exists(X \setminus D)\big) \,, \tag{5}$$

where an ideal decomposition of $X \setminus U$ for an upwards-closed $U = \bigcup_{x \in \min U} \uparrow x$ can be computed since $X \setminus U = \bigcap_{x \in \min U} X \setminus \uparrow x$, while the minimal elements of $X \setminus D$ can be computed using the *Generalised Valk-Jantzen Lemma* [14].[2]

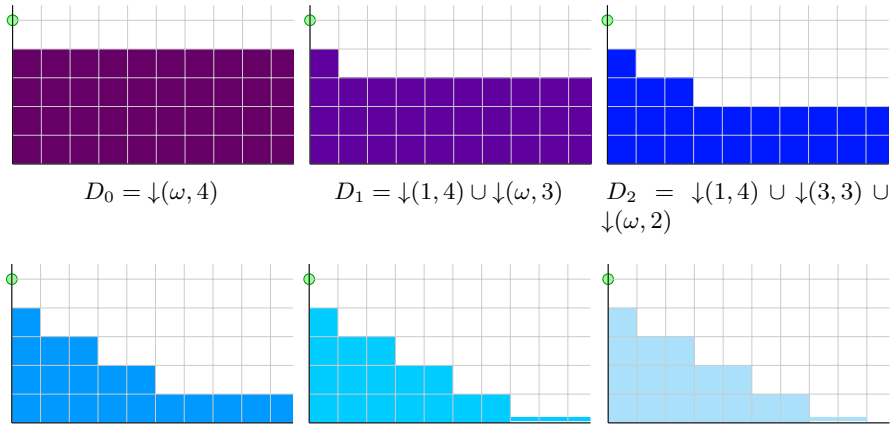
Hence, assuming effective ideal representations, our view of the backward coverability algorithm using downwards-closed sets is effective in the same cases as the usual upwards-closed version of the algorithm.

3.2. **Coverability for VAS and Reset VAS.** In order to instantiate the backward coverability algorithm for VAS and reset VAS, we need to prove that they satisfy the various effectiveness assumptions. Example 2.5 dealt with (CI), (II), and (CU'). By Remark 3.3, (Pre) is also effective since existential predecessors are well-known to be computable [1, 13].

We nevertheless provide here a direct computation of universal predecessors. Thus, given a downwards-closed $D = \downarrow\boldsymbol{u}_1 \cup \cdots \cup \downarrow\boldsymbol{u}_m$ for some $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ in $\mathbb{N}_\omega^d$, we want to compute a finite set of vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ from $\mathbb{N}_\omega^d$ such that $\mathrm{Pre}_\forall(D) = \downarrow\boldsymbol{v}_1 \cup \cdots \cup \downarrow\boldsymbol{v}_n$. Using (CI) we can then select the maximal such $\boldsymbol{v}_j$ to obtain incomparable ideals.

3.2.1. *Universal Predecessors in VAS.* Thanks to (II) and the fact that $\boldsymbol{A}$ is finite (VAS are finitely branching), we start by reducing our computation to that of predecessors along a specific action $\boldsymbol{a}$ from $\boldsymbol{A}$: $\mathrm{Pre}_\forall(D) =$

[2]This last result is quite general, but it might be simpler in practice to implement this computation directly; for instance, in the case of $X = \mathbb{N}^d$, $\mathbb{N}^d \setminus \downarrow\boldsymbol{u}$ for $\boldsymbol{u} \in \mathbb{N}_\omega^d$ is decomposed as $\bigcup_{1 \leq j \leq d \mid \boldsymbol{u}(j) < \omega} \uparrow\boldsymbol{u}_{\setminus j}$ where $\boldsymbol{u}_{\setminus j}(i) = 0$ if $i \neq j$ and $\boldsymbol{u}_{\setminus j}(j) = \boldsymbol{u}(j) + 1$ otherwise, and $\uparrow\boldsymbol{u} \cap \uparrow\boldsymbol{v} = \uparrow\boldsymbol{w}$ where $\boldsymbol{w}(i) = \max_\leq(\boldsymbol{u}(i), \boldsymbol{v}(i))$ for all $1 \leq i \leq d$.



$D_0 = \downarrow(\omega, 4)$ $D_1 = \downarrow(1,4) \cup \downarrow(\omega, 3)$ $D_2 = \downarrow(1,4) \cup \downarrow(3,3) \cup \downarrow(\omega, 2)$

$D_3 = \downarrow(1,4) \cup \downarrow(3,3) \cup \downarrow(5,2) \cup \downarrow(\omega, 1)$ $D_4 = \downarrow(1,4) \cup \downarrow(3,3) \cup \downarrow(5,2) \cup \downarrow(7,1) \cup \downarrow(\omega, 0)$ $D_* = D_5 = \downarrow(1,4) \cup \downarrow(3,3) \cup \downarrow(5,2) \cup \downarrow(7,1) \cup \downarrow(9,0)$

FIGURE 1. The successive D_k for $\boldsymbol{A}_{\div 2}$ with target $\boldsymbol{t} = (0, 5)$.

$\bigcap_{\boldsymbol{a} \in \boldsymbol{A}} \mathrm{Pre}_\forall^{\boldsymbol{a}}(D)$ where

$$\mathrm{Pre}_\forall^{\boldsymbol{a}}(D) \stackrel{\text{def}}{=} \{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in \mathbb{N}^d \implies \boldsymbol{v} + \boldsymbol{a} \in D\} \tag{6}$$

$$= \{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \notin \mathbb{N}^d\} \cup \{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in D\} \tag{7}$$

$$= \mathbb{N}^d \setminus \uparrow\theta(\boldsymbol{a}) \cup \{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in D\}\ , \tag{8}$$

where $\theta(\boldsymbol{a}) \stackrel{\text{def}}{=} \min_\sqsubseteq\{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in \mathbb{N}^d\}$ is called the *threshold* of $\boldsymbol{a}$ and can be computed for all $1 \leq i \leq d$ by

$$\theta(\boldsymbol{a})(i) = \begin{cases} 0 & \text{if } \boldsymbol{a}(i) \geq 0 \\ -\boldsymbol{a}(i) & \text{otherwise.} \end{cases} \tag{9}$$

Thus by (CU') it only remains to compute a representation for the decomposition of $\{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in D\} = \bigcup_{1 \leq j \leq m}\{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in \downarrow\boldsymbol{u}_j\}$. For each ideal $\downarrow\boldsymbol{u}_j$ in the decomposition of D, $\{\boldsymbol{v} \in \mathbb{N}^d \mid \boldsymbol{v} + \boldsymbol{a} \in \downarrow\boldsymbol{u}_j\}$ is either the empty set if $\boldsymbol{u}_j \not\sqsupseteq \theta(-\boldsymbol{a})$, or $\downarrow(\boldsymbol{u} - \boldsymbol{a})$ otherwise, where addition is extended with $\omega + z = \omega$ for all z in $\mathbb{Z}$.

Example 3.4. Recall the VAS $\boldsymbol{A}_{\div 2} = \{(-2, 1)\}$ from Example 2.2, and consider the target configuration $\boldsymbol{t} = (0, 5)$. This yields $D_0 = \downarrow(\omega, 4)$, and the backward coverability algorithm computes the set of all configurations from which $\boldsymbol{A}_{\div 2}$ cannot compute at least 5 in its second component; see Figure 1.

3.2.2. *Universal Predecessors in Reset VAS.* The same reasoning holds for reset VAS as for VAS:

$$\mathrm{Pre}_\forall(D) = \bigcap_{(\boldsymbol{a}, R) \in \boldsymbol{A}} \left(\mathbb{N}^d \setminus \uparrow\theta(\boldsymbol{a}) \cup \bigcup_{1 \leq j \leq m} \{\boldsymbol{v} \in \mathbb{N}^d \mid R(\boldsymbol{v} + \boldsymbol{a}) \in \downarrow\boldsymbol{u}_j\} \right)\ , \tag{10}$$

(where $R(\boldsymbol{v} + \boldsymbol{a})$ is defined as in Example 2.4). In order to compute a representation for this last set, given a vector $\boldsymbol{v}$ in $\mathbb{N}_\omega^d$ and $R \subseteq \{1, \ldots, d\}$,

define $\overline{\boldsymbol{v}}^R$ as the *R-closure* of $\boldsymbol{v}$, which replaces the components in R by ω's:

$$\overline{\boldsymbol{v}}^R(i) \stackrel{\text{def}}{=} \begin{cases} \omega & \text{if } i \in R \\ \boldsymbol{v}(i) & \text{otherwise.} \end{cases} \tag{11}$$

Then $\{\boldsymbol{v} \in \mathbb{N}^d \mid R(\boldsymbol{v}+\boldsymbol{a}) \in {\downarrow}\boldsymbol{u}_j\}$ is either the empty set if $\overline{\boldsymbol{u}_j}^R \not\supseteq \theta(-\boldsymbol{a})$, or ${\downarrow}\big(\overline{\boldsymbol{u}_j}^R - \boldsymbol{a}\big)$ otherwise.

Example 3.5. Recall the reset VAS $\boldsymbol{A}_{\log}$ from Example 2.3, in which the first two vector components are used to encode two control states. Setting

$$D_0 = {\downarrow}(1,0,\omega,\omega,1) \cup {\downarrow}(0,1,\omega,\omega,0)\ ,$$

the backward coverability algorithm computes as follows the set of all configurations from which $\boldsymbol{A}_{\log}$ cannot compute in its last component either at least 2 in state $(1,0)$ or at least 1 in state $(0,1)$.

$$\begin{aligned}
D_1 &= {\downarrow}(0,0,\omega,\omega,1) \cup {\downarrow}(1,0,1,\omega,1) \cup {\downarrow}(1,0,\omega,\omega,0) \cup {\downarrow}(0,1,\omega,\omega,0)\ ,\\
D_2 &= {\downarrow}(0,0,\omega,\omega,1) \cup {\downarrow}(1,0,1,0,1) \cup {\downarrow}(1,0,0,\omega,1) \cup {\downarrow}(1,0,\omega,\omega,0)\\
&\quad \cup {\downarrow}(0,1,\omega,0,0) \cup {\downarrow}(0,1,0,\omega,0)\ ,\\
D_3 &= {\downarrow}(0,0,\omega,\omega,1) \cup {\downarrow}(1,0,1,0,1) \cup {\downarrow}(1,0,0,1,1) \cup {\downarrow}(1,0,\omega,\omega,0)\\
&\quad \cup {\downarrow}(0,1,2,0,0) \cup {\downarrow}(0,1,0,1,0)\ ,\\
D_4 &= {\downarrow}(0,0,\omega,\omega,1) \cup {\downarrow}(1,0,1,0,1) \cup {\downarrow}(1,0,0,1,1) \cup {\downarrow}(1,0,1,\omega,0)\\
&\quad \cup {\downarrow}(1,0,\omega,0,0) \cup {\downarrow}(0,1,2,0,0) \cup {\downarrow}(0,1,0,1,0)\ ,\\
D_5 &= {\downarrow}(0,0,\omega,\omega,1) \cup {\downarrow}(1,0,1,0,1) \cup {\downarrow}(1,0,0,1,1) \cup {\downarrow}(1,0,0,\omega,0)\\
&\quad \cup {\downarrow}(1,0,1,1,0) \cup {\downarrow}(1,0,3,0,0) \cup {\downarrow}(0,1,2,0,0) \cup {\downarrow}(0,1,0,1,0)\ ,\\
D_* = D_6 &= {\downarrow}(0,0,\omega,\omega,1) \cup {\downarrow}(1,0,1,0,1) \cup {\downarrow}(1,0,0,1,1) \cup {\downarrow}(1,0,0,2,0)\\
&\quad \cup {\downarrow}(1,0,1,1,0) \cup {\downarrow}(1,0,3,0,0) \cup {\downarrow}(0,1,2,0,0) \cup {\downarrow}(0,1,0,1,0)\ .
\end{aligned}$$

3.3. **Ackermann Upper Bounds.** Let us finally show how to bound the running time of the backward coverability algorithm on VAS and reset VAS. The main ingredient to that end is a combinatorial statement on the length of *controlled* descending chains of downwards-closed sets.

3.3.1. *Controlled Descending Chains.* Consider some set X with a norm $\|.\|\colon X \to \mathbb{N}$. Given a monotone *control* function $g\colon \mathbb{N} \to \mathbb{N}$ and an *initial norm* $n \in \mathbb{N}$, we say that a sequence $x_0, x_1, \dots$ of elements from X is (g,n)-*controlled* if $\|x_i\| \leq g^i(n)$ the ith iterate of g applied to n. In particular, $\|x_0\| \leq n$ initially.

This notion can be applied to the descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ constructed by the backward coverability algorithm for a d-dimensional VAS or reset VAS $\boldsymbol{A}$ and target vector $\boldsymbol{t} \in \mathbb{N}^d$. We define for this $\|.\|$ as the infinity norm on elements and finite subsets of $\mathbb{Z}_\omega^d \stackrel{\text{def}}{=} (\mathbb{Z} \uplus \{\omega\})^d$, i.e. the maximum absolute value of any occurring integer. For instance, $\|\{(1,\omega,5),(0,\omega,\omega)\}\| = 5$, and in Example 2.2 $\|\boldsymbol{A}_{\div 2}\| = 2$. When considering a downwards-closed set D with decomposition ${\downarrow}\boldsymbol{u}_1 \cup \cdots \cup {\downarrow}\boldsymbol{u}_m$, we define $\|D\| \stackrel{\text{def}}{=} \|\{\boldsymbol{u}_1,\dots,\boldsymbol{u}_m\}\|$. Hence what is controlled in a descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ is actually its ideal representation.

Claim 3.6 (Control for VAS and Reset VAS). The descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ is (g, n)-controlled for $g(x) \stackrel{\text{def}}{=} x + \|\boldsymbol{A}\|$ and $n \stackrel{\text{def}}{=} \|\boldsymbol{t}\|$.

Proof. The fact that $\|D_0\| \leq \|\boldsymbol{t}\|$ follows from analysing (CU') in Example 2.5. Regarding the control function g, observe that taking unions and intersections of ideals using (II) cannot increase the norm. Hence it suffices to show that $\|\mathrm{Pre}_\forall(D)\| \leq \|D\| + \|\boldsymbol{A}\|$ for all $D = {\downarrow}\boldsymbol{u}_1 \cup \cdots \cup {\downarrow}\boldsymbol{u}_m$. Note that for reset VAS, $\|\overline{\boldsymbol{u}_j}^R - \boldsymbol{a}\| \leq \|\boldsymbol{u}_j - \boldsymbol{a}\|$. Hence for both VAS and reset VAS, $\|\mathrm{Pre}_\forall(D)\| \leq \max_{\boldsymbol{a}} \max_{1 \leq j \leq m}(\|\mathbb{N}^d \setminus {\uparrow}\theta(\boldsymbol{a})\|, \|\boldsymbol{u}_j - \boldsymbol{a}\|)$. We conclude by observing that $\|\mathbb{N}^d \setminus {\uparrow}\theta(\boldsymbol{a})\| \leq \|\boldsymbol{a}\| \leq \|\boldsymbol{A}\|$ by (CU') and $\|\boldsymbol{u}_j - \boldsymbol{a}\| \leq \|\boldsymbol{u}_j\| + \|\boldsymbol{a}\| \leq \|D\| + \|\boldsymbol{A}\|$. □

3.3.2. *Proper Ideals & Upper Bound.* Consider a computation $D_0 \supsetneq D_1 \supsetneq \cdots \supsetneq D_\ell = D_{\ell+1}$ of the backward coverability algorithm for a VAS or a reset VAS. At each step $0 \leq k \leq \ell$, the cost of computing D_{k+1} from D_k and of checking for termination is polynomial in $\|A\|$ and $\|D_k\|$. The difficulty is to evaluate how large ℓ can be.

The idea here is that, at every step $0 \leq k < \ell$, there is at least one *proper* ideal I_k: an ideal appearing in the representation of D_k but not in that of D_{k+1}; then $I_k \subseteq D_k$ but $I_k \not\subseteq D_{k+1}$. Note that for all $0 \leq j < k < \ell$, $I_j \not\subseteq I_k$, since the contrary would entail $I_j \subseteq I_k \subseteq D_k \subseteq D_{j+1}$. Hence the sequence $(I_k)_{0 \leq k < \ell}$ of proper ideals is a *bad* sequence.

In the case of VAS and reset VAS, those proper ideals I_k are of the form ${\downarrow}\boldsymbol{v}_k$ for some representant $\boldsymbol{v}_k$ in $\mathbb{N}^d_\omega$, and the sequence $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\ell-1}$ is also controlled by (g, n) according to Claim 3.6. Using the combinatorial results from [26, Corollary 2.25 and Theorem 2.34] on such bad sequences, we obtain the following *length function theorem* for descending chains of downwards-closed subsets of $\mathbb{N}^d$; as the point of this paper is rather to avoid such high complexities, the reader will find all the necessary details in A.

Theorem 3.7 (Length Function Theorem for Descending Chains)**.** *Let $n > 0$. Any (g, n)-controlled descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ of downwards-closed subsets of $\mathbb{N}^d$ is of length at most $h_{\omega^{d+1}}(n \cdot d!)$, where $h(x) \stackrel{\text{def}}{=} d \cdot g(x)$.*

Here h_α for an ordinal α and base function h denotes the αth Cichoń function [26]. Each of the ℓ steps of computation can furthermore be performed in time polynomial in $g^\ell(n)$.

Since g is primitive-recursive according to Claim 3.6, the overall complexity for an instance of size n is bounded by ackermann$(p(n))$ for some primitive-recursive function p, which lies within the complexity class Ackermann [25, Corollary 4.3]. Such an upper bound is overly pessimistic for VAS, but is actually tight for reset VAS: coverability for reset VAS is indeed complete for Ackermann [27, 26].

4. Complexity for VAS

We know from Bozzelli and Ganty's 2ExpTime upper bound [7] for the backward coverability algorithm that the Ackermann upper bound from the previous section is far from tight in the case of VAS. We show in this section that the descending chains $D_0 \supsetneq D_1 \supsetneq \cdots$ computed by the backward coverability algorithm for VAS enjoy a structural invariant, which we dub

ω-monotonicity, and which is absent from the chains computed for reset VAS. In turn, we show in Theorem 4.4, that controlled decreasing chains that are ω-monotone are much shorter, allowing us to derive the desired 2ExpTime bound in Corollary 4.6.

4.1. **Transitions Between Proper Ideals.** The proof of ω-monotonicity in the case of VAS can be shown directly, but reflects a more general *proper transition sequence* property of the generic backward coverability algorithm, which we are going to show in the general setting.

4.1.1. *Transitions over Ideals.* Let us first lift the transition relation $\to$ of a WSTS $(X, \to, \leq)$ to work over ideals. Define for any ideal I of X

$$\mathrm{Post}_\exists(I) \stackrel{\text{def}}{=} \{z \in X \mid \exists x \in I \,.\, x \to z\} \,. \tag{12}$$

Then $\downarrow\mathrm{Post}_\exists(I)$ is downwards-closed with a unique decomposition into maximal ideals. We follow Blondin et al. [4] and write '$I \to J$' if J is an ideal from the decomposition of $\downarrow\mathrm{Post}_\exists(I)$.

Example 4.1 (Transitions over Vector Ideals)**.** In the case of a VAS $\boldsymbol{A}$, observe that, if $\boldsymbol{v}$ is a vector from $\mathbb{N}_\omega^d$, then $\mathrm{Post}_\exists(\downarrow\boldsymbol{v}) = \bigcup_{\boldsymbol{a}\in\boldsymbol{A}} \downarrow(\boldsymbol{v}+\boldsymbol{a})$. Each such $\downarrow(\boldsymbol{v}+\boldsymbol{a})$ is already an ideal. In the case of a reset VAS $\boldsymbol{A}$, we have similarly $\mathrm{Post}_\exists(\downarrow\boldsymbol{v}) = \bigcup_{(\boldsymbol{a},R)\in\boldsymbol{A}} \downarrow R(\boldsymbol{v}+\boldsymbol{a})$.

4.1.2. *Proper Transition Sequences.* We can now state the result that motivates this subsection:

Claim 4.2 (Proper Transition Sequence)*.* If I_{k+1} is a proper ideal of D_{k+1}, then there exist an ideal J and a proper ideal I_k of D_k such that $I_{k+1} \to J \subseteq I_k$.

Proof. Consider first a computation $D_0 \supsetneq D_1 \supsetneq \cdots \supsetneq D_\ell = D_{\ell+1}$ of the generic backward coverability algorithm, where each of the D_k is represented as a finite union of ideals. At each refinement step $D_k \supsetneq D_{k+1}$ of the algorithm, some of the ideals from the decomposition of D_k—namely the *proper* ones—might be removed, while others might remain untouched in the decomposition of D_{k+1}. Thus, an ideal is proper in D_k if and only if it intersects the set of elements *excluded* between steps k and $k+1$: by basic set operations, first observe that (2) is equivalent to

$$D_{k+1} = D_k \setminus \{x \in D_k \mid \exists z \notin D_k \,.\, x \to z\} \,. \tag{13}$$

Moreover, noting $D_{-1} \stackrel{\text{def}}{=} X$, z in (13) must belong to D_{k-1}, or x would have already been excluded before step k. We have therefore $D_{k+1} = D_k \setminus E_k$ where

$$E_{-1} \stackrel{\text{def}}{=} \{x \in X \mid x \geq y\} \,, \qquad E_k \stackrel{\text{def}}{=} \{x \in D_k \mid \exists z \in E_{k-1} \,.\, x \to z\} \,. \tag{14}$$

Consider now a proper ideal I_{k+1} of D_{k+1}: this means $I_{k+1} \cap E_{k+1} \neq \emptyset$. This implies in turn $\downarrow\mathrm{Post}_\exists(I_{k+1}) \cap E_k \neq \emptyset$ by (14), thus there exists J such that $I_{k+1} \to J$ and $J \cap E_k \neq \emptyset$.

Since $I_{k+1} \subseteq D_{k+1} \subseteq \mathrm{Pre}_\forall(D_k)$ by (2), we also know that $\mathrm{Post}_\exists(I_{k+1}) \subseteq D_k$. By ideal irreducibility, it means that $J \subseteq I_k$ for some ideal I_k from the decomposition of D_k. Observe finally that $I_k \cap E_k \supseteq J \cap E_k \neq \emptyset$, i.e. that I_k is proper. $\square$

4.2. **ω-Monotonicity.** For $\boldsymbol{u}$ in $\mathbb{N}_\omega^d$, its *ω-set* is the subset $\omega(\boldsymbol{u})$ of $\{1,\dots,d\}$ such that $\boldsymbol{u}(i)=\omega$ if and only if $i\in\omega(\boldsymbol{u})$. We say that a descending chain $D_0 \supsetneq D_1 \supsetneq \cdots \supsetneq D_\ell$ of downwards-closed subsets of $\mathbb{N}^d$ is *ω-monotone* if for all $0\le k<\ell-1$ and all proper ideals $\downarrow\boldsymbol{v}_{k+1}$ in the decomposition of D_{k+1}, there exists a proper ideal $\downarrow\boldsymbol{v}_k$ in the decomposition of D_k such that $\omega(\boldsymbol{v}_{k+1})\subseteq\omega(\boldsymbol{v}_k)$. Note that, in VAS and reset VAS alike, because $D_k\supsetneq D_{k+1}$ and by ideal irreducibility, for any $\boldsymbol{v}_{k+1}$ from the decomposition of D_{k+1} there exists some $\boldsymbol{v}_k$ from that of D_k with $\downarrow\boldsymbol{v}_{k+1}\subseteq\downarrow\boldsymbol{v}_k$ and thus $\omega(\boldsymbol{v}_{k+1})\subseteq\omega(\boldsymbol{v}_k)$, but that $\boldsymbol{v}_k$ may not be proper (then $\boldsymbol{v}_{k+1}=\boldsymbol{v}_k$).

Claim 4.3 (VAS Descending Chains are ω-Monotone)*.* The descending chains computed by the backward coverability algorithm for VAS are ω-monotone.

Proof. Let $D_0\supsetneq D_1\supsetneq\cdots\supsetneq D_\ell$ be the descending chain computed for a VAS $\boldsymbol{A}$. Suppose $0\le k<\ell-1$ and $\downarrow\boldsymbol{v}_{k+1}$ is a proper ideal in the decomposition of D_{k+1}. By Claim 4.2, there exists a proper ideal $\downarrow\boldsymbol{v}_k$ in the decomposition of D_k such that $\boldsymbol{v}_{k+1}+\boldsymbol{a}\sqsubseteq\boldsymbol{v}_k$. We conclude that $\omega(\boldsymbol{v}_{k+1})\subseteq\omega(\boldsymbol{v}_k)$. □

As we can see with Example 3.5 however, the descending chains computed for reset VAS are in general *not* ω-monotone: $(1,0,\omega,\omega,0)$ is proper in D_3 and has a proper transition to $(0,1,0,\omega,0)$ in D_2 using $(-1,1,-2,1,0,\{3\})$ from $\boldsymbol{A}_{\log}$, but no ideal with $\{3,4\}$ as ω-set is proper in D_2.

4.3. **Upper Bound.** We are now in position to state a refinement of Theorem 3.7 for ω-monotone controlled descending chains. For a control function $g\colon\mathbb{N}\to\mathbb{N}$, define the function $\widetilde{g}\colon\mathbb{N}^2\to\mathbb{N}$ by induction on its first argument:

$$\widetilde{g}(0,n)\stackrel{\text{def}}{=}1\ ,\qquad \widetilde{g}(m+1,n)\stackrel{\text{def}}{=}\widetilde{g}(m,n)+(g^{\widetilde{g}(m,n)}(n)+1)^{m+1}\ . \tag{15}$$

Theorem 4.4 (Length Function Theorem for ω-Monotone Descending Chains)**.** *Any (g,n)-controlled ω-monotone descending chain $D_0\supsetneq D_1\supsetneq\cdots$ of downwards-closed subsets of $\mathbb{N}^d$ is of length at most $\widetilde{g}(d,n)$.*

Proof. Note that D_ℓ the last element of the chain has the distinction of not having any proper ideal, hence it suffices to bound the index k of the last set D_k with a proper ideal $\downarrow\boldsymbol{v}_k$, and add one to get a bound on ℓ. We are going to establish by induction on $d-|I|$ that if $\downarrow\boldsymbol{v}_k$ is a proper ideal from the decomposition of D_k and its ω-set is I, then $k<\widetilde{g}(d-|I|,n)$, which by monotonicity of $\widetilde{g}$ in its first argument entails $k<\widetilde{g}(d,n)$ as desired.

For the base case, $|I|=d$ implies that $\boldsymbol{v}_k$ is the vector with ω's in every coordinate, which can only occur in D_0. The inductive step is handled by the following claim, when setting $k<\widetilde{g}(d-|I|-1,n)$ by induction hypothesis for the last index with a proper ideal whose ω-set strictly includes I:

Claim 4.5. Let I and $k<k'$ be such that:

(i) for all $j\in\{k+1,\dots,k'-1\}$, the decomposition of D_j does not contain a proper ideal whose ω-set strictly includes I;
(ii) the decomposition of $D_{k'}$ contains a proper ideal whose ω-set is I.

Then we have $k'-k\le(\|D_{k+1}\|+1)^{(d-|I|)}$.

For a proof, from assumption (ii), by applying the ω-monotonicity for $j=k'-1,k'-2,\dots,k+1$ and due to assumption (i), there exists a proper ideal $\downarrow\boldsymbol{v}_j$ in the decomposition of D_j and such that $\omega(\boldsymbol{v}_j)=I$ for all

$j \in \{k+1, \ldots, k'\}$. Since they are proper, those $k' - k$ vectors are mutually distinct.

Consider any such $\boldsymbol{v}_j$. Since $D_{k+1} \supseteq D_j$, by ideal irreducibility there exists a vector $\boldsymbol{u}_j$ in the decomposition of D_{k+1} such that $\boldsymbol{v}_j \sqsubseteq \boldsymbol{u}_j$. We have that $\omega(\boldsymbol{u}_j) = I$, since otherwise $\boldsymbol{u}_j$ would be proper at $D_{j'}$ for some $j' \in \{k+1, \ldots, j-1\}$, which would contradict assumption (i). Hence $\|\boldsymbol{v}_j\| \leq \|\boldsymbol{u}_j\| \leq \|D_{k+1}\|$.

To conclude, note that there can be at most $(\|D_{k+1}\| + 1)^{(d-|I|)}$ mutually distinct vectors in $\mathbb{N}^d_\omega$ with I as ω-set and norm bounded by $\|D_{k+1}\|$. □

Finally, putting together Claim 3.6 (control for VAS), Claim 4.3 (ω-monotonicity), and Theorem 4.4 (lengths of controlled ω-monotone descending chains), we obtain that the backward coverability algorithm for VAS runs in 2ExpTime, and in pseudo-polynomial time if the dimension d is fixed.

Corollary 4.6. *For any d-dimensional VAS $\boldsymbol{A}$ and target vector $\boldsymbol{t}$, the backward coverability algorithm terminates after at most $((\|\boldsymbol{A}\| + 1)(\|\boldsymbol{t}\| + 2))^{(d+1)!}$ steps.*

Proof. Let $h(m,n) = \widetilde{g}(m,n)(\|\boldsymbol{A}\| + 1)(n+2)$ where $g(x) = x + \|\boldsymbol{A}\|$. We have $h(m+1,n) \leq (h(m,n))^{m+2}$, so $\widetilde{g}(m,n) \leq h(m,n) \leq ((\|\boldsymbol{A}\| + 1)(n + 2))^{(m+1)!}$. □

As mentioned in the introduction, this 2ExpTime upper bound is not tight for the coverability problem for VAS—which is ExpSpace-complete [20, 22]. However, the bound is tight for the backward coverability algorithm: as shown by Bozzelli and Ganty [7] using Lipton's construction, there is a uniform family of VAS $(\boldsymbol{A}_n)_{n \in \mathbb{N}}$ each of size $O(n)$ on which the algorithm requires 2^{2^n} iterations.

5. Top-Down Tree Coverability

We turn to demonstrating how easily our new proof of the doubly-exponential bound for the backward coverability algorithm on VAS can be extended to derive (the known) optimal bounds for top-down alternating branching VAS: Tower in general [17] and 2ExpTime with alternation only [8]. While this complexity analysis does not yield any new results, we believe it brings some additional insights into our dual view of the backward coverability algorithm. Also, unlike with VAS, the upper bounds we obtain are tight.

5.1. Alternating Branching VAS.

Recall that an *alternating branching vector addition system (ABVAS)* of dimension $d \in \mathbb{N}$ consists of: a finite set of *unary* rules $\boldsymbol{A} \subseteq \mathbb{Z}^d$, a finite set of *fork* rules $\boldsymbol{B}_\wedge \subseteq \mathbb{Z}^d \times \mathbb{Z}^d$, and a finite set of *split* rules $\boldsymbol{B}_+ \subseteq \mathbb{Z}^d$. A computation is a tree whose nodes are labelled by vectors in $\mathbb{N}^d$, and such that every non-leaf node ν is obtained by applying some rule (we write $\boldsymbol{v}(\cdot)$ for node labels):

- either ν has one child ν' and $\boldsymbol{v}(\nu) + \boldsymbol{a} = \boldsymbol{v}(\nu')$ for some unary rule $\boldsymbol{a}$ from $\boldsymbol{A}$,
- or ν has two children ν_1, ν_2 and $\boldsymbol{v}(\nu) + \boldsymbol{b_1} = \boldsymbol{v}(\nu_1)$ and $\boldsymbol{v}(\nu) + \boldsymbol{b}_2 = \boldsymbol{v}(\nu_2)$ for some fork rule $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ from $\boldsymbol{B}_\wedge$,

- or ν has two children ν_1, ν_2 and $\boldsymbol{v}(\nu) + \boldsymbol{b} = \boldsymbol{v}(\nu_1) + \boldsymbol{v}(\nu_2)$ for some split rule $\boldsymbol{b}$ from $\boldsymbol{B}_+$.

Thus a VAS is an ABVAS with empty $\boldsymbol{B}_\wedge$ and $\boldsymbol{B}_+$; an ABVAS with $\boldsymbol{B}_\wedge = \emptyset$ is called a *branching* VAS (BVAS), and one with $\boldsymbol{B}_+ = \emptyset$ is called an *alternating* VAS (AVAS). Such systems appear in a variety of contexts in relation with fragments of propositional linear logic, simulation relations, computational linguistics, etc. [9, 24, 8, 17].

The *top-down coverability* problem (also called the *leaf coverability* problem), given a d-dimensional top-down ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$, and root and target vectors $\boldsymbol{r}, \boldsymbol{t} \in \mathbb{N}^d$, asks whether there exists a computation whose root label is $\boldsymbol{r}$ and whose every leaf label is $\sqsupseteq \boldsymbol{t}$. We remark that the problem is equivalent to the *join* variant in which the fork rules are applied by taking pointwise maxima (i.e. $\boldsymbol{v}(\nu) = \max\{\boldsymbol{v}(\nu_1) - \boldsymbol{b}_1, \boldsymbol{v}(\nu_2) - \boldsymbol{b}_2\}$), and also to the reachability problem for *lossy* ABVAS [17].

5.2. **Backward Coverability.** The theory of well-structured transition systems can be lifted in a very natural way to top-down tree computations, although we are not aware of any literature on the subject. One can nevertheless find enlightening parallels with *monotonic games* [2]. A top-down coverability instance for an ABVAS defines indeed a two-players game over an infinite arena with positions in $\mathbb{N}^d$. Starting from $\boldsymbol{r}$, Player 1 attempts to cover $\boldsymbol{t}$, while Player 2 attempts to prevent it. At each round, in a configuration $\boldsymbol{v} \in \mathbb{N}^d$,

(1) Player 1 chooses an applicable rule $\boldsymbol{a}$ from $\boldsymbol{A}$ with $\boldsymbol{v} + \boldsymbol{a} \sqsupseteq \boldsymbol{0}$, or $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ from $\boldsymbol{B}_\wedge$ with $\boldsymbol{v} + \boldsymbol{b}_1 \sqsupseteq \boldsymbol{0}$ and $\boldsymbol{v} + \boldsymbol{b}_2 \sqsupseteq \boldsymbol{0}$, or $\boldsymbol{b}$ from $\boldsymbol{B}_+$ with $\boldsymbol{v} + \boldsymbol{b} \sqsupseteq \boldsymbol{0}$ and in that case further chooses two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ from $\mathbb{N}^d$ such that $\boldsymbol{v} + \boldsymbol{b} = \boldsymbol{v}_1 + \boldsymbol{v}_2$.
(2) Player 2 then chooses a branch, i.e. $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{a}$ in case of a unary rule $\boldsymbol{a}$ from $\boldsymbol{A}$, or $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{b}_i$ for $i \in \{1, 2\}$ in case of a fork rule $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ in $\boldsymbol{B}_\wedge$, or $\boldsymbol{v}' = \boldsymbol{v}_i + \boldsymbol{b}$ for $i \in \{1, 2\}$ in case of a split rule $\boldsymbol{b}$ from $\boldsymbol{B}_+$, and the game proceeds to the next round from $\boldsymbol{v}'$.

Player 1 wins the game if a configuration $\boldsymbol{v} \sqsupseteq \boldsymbol{t}$ is eventually reached. One can see that ABVAS computation trees represent strategies for Player 1, and such a tree has all its leaves $\sqsupseteq \boldsymbol{t}$ if and only if it is a winning strategy.

5.2.1. *Winning Regions.* Observe that, using this game-theoretic viewpoint, the *winning region* for Player 1 is an upwards-closed set of configurations, and conversely—the game is determined—the winning region for Player 2 is downwards-closed. In our dual view of the backward coverability algorithm using downwards-closed sets, we actually compute the winning region D_* for Player 2 by successive refinements of the winning region D_k for the game limited to at most k rounds. A configuration $\boldsymbol{v}$ then belongs to D_{k+1}, if for all choices of rules by Player 1, there exists a branch with $\boldsymbol{v}'$ in the winning region D_k of Player 2. Using this insight, the generic algorithm from Section 3 can be adapted to top-down coverability by replacing the $\mathrm{Pre}_\forall$ operator with

$$\mathrm{Pre}_{\forall\exists}(S) \stackrel{\text{def}}{=} \mathrm{Pre}_\forall^{\boldsymbol{A}}(S) \cap \mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(S) \cap \mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_+}(S) \tag{16}$$

where

$$\mathrm{Pre}_{\forall}^{\boldsymbol{A}}(S) \stackrel{\text{def}}{=} \{\boldsymbol{v} \in \mathbb{N}^d \,|\, \forall \boldsymbol{a} \in \boldsymbol{A} \,.\, \boldsymbol{v} + \boldsymbol{a} \in \mathbb{N}^d \Rightarrow \boldsymbol{v} + \boldsymbol{a} \in S\}\,, \tag{17}$$

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(S) \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{N}^d \;\middle|\; \begin{array}{c} \forall (\boldsymbol{b}_1, \boldsymbol{b}_2) \in \boldsymbol{B}_\wedge \,.\, \boldsymbol{v} + \boldsymbol{b}_1 \in \mathbb{N}^d \wedge \boldsymbol{v} + \boldsymbol{b}_2 \in \mathbb{N}^d \\ \Rightarrow \boldsymbol{v} + \boldsymbol{b}_1 \in S \vee \boldsymbol{v} + \boldsymbol{b}_2 \in S \end{array} \right\}, \tag{18}$$

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_+}(S) \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{N}^d \;\middle|\; \begin{array}{c} \forall \boldsymbol{b} \in \boldsymbol{B}_+ . \forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{N}^d \,.\, \boldsymbol{v} + \boldsymbol{b} = \boldsymbol{v}_1 + \boldsymbol{v}_2 \\ \Rightarrow \boldsymbol{v}_1 \in S \vee \boldsymbol{v}_2 \in S \end{array} \right\}. \tag{19}$$

5.2.2. *Downward-Closure and Correctness.* It is straightforward to check that $\mathrm{Pre}_{\forall\exists}$ preserves the property of downward closure as in Claim 3.2. Given an initial downwards-closed $D_0 \subseteq \mathbb{N}^d$, as in Section 3, we write D_* for the last set in the longest (necessarily finite) descending chain of downwards-closed sets $D_0 \supsetneq D_1 \supsetneq \cdots$ defined by $D_{k+1} = D_k \cap \mathrm{Pre}_{\forall\exists}(D_k)$. From the definition of $\mathrm{Pre}_{\forall\exists}$, we have the required correctness property, matching Claim 3.1:

Claim 5.1 (Correctness). For all k, D_k consists of all vectors $v \in \mathbb{N}^d$ such that all computation trees of $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ whose root label is v and whose height is at most k have some leaf label in D_0.

Thus, the set D_* consists of all vectors $v \in \mathbb{N}^d$ such that all computations of $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ whose root label is v have some leaf label in D_0.

5.2.3. *Effectiveness and Control.* Similarly to the cases of VAS and reset VAS in Sections 3.2 and 3.3, we shall give an explicit computation for predecessors, allowing us to provide bounds on the norms of ideals in the course of the backward coverability algorithm. The counterpart to Claim 3.6 is then:

Claim 5.2 (Control for ABVAS).

(a) The ideal decomposition of $\mathrm{Pre}_{\forall\exists}(D)$ is computable for all downwards-closed $D \subseteq \mathbb{N}^d$.
(b) The descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ is (g, n)-controlled for

$$g(x) \stackrel{\text{def}}{=} \max\{x + \|\boldsymbol{A}\|, x + \|\boldsymbol{B}_\wedge\|, 2x + \|\boldsymbol{B}_+\| + 1\}$$

and $n \stackrel{\text{def}}{=} \|\boldsymbol{t}\|$. For AVAS, i.e. when $\boldsymbol{B}_+$ is empty, the term $2x + \|\boldsymbol{B}_+\| + 1$ disappears.

Proof. Let us examine the three cases in (16); by (II) the norm of the ideals in $\mathrm{Pre}_{\forall\exists}(D)$ will be bounded by the maximal norm over all three cases. For (17),

$$\mathrm{Pre}_{\forall}^{\boldsymbol{A}}(D) = \bigcap_{\boldsymbol{a} \in \boldsymbol{A}} \mathrm{Pre}_{\forall}^{\boldsymbol{a}}(D) \tag{20}$$

as defined in (6), entailing $\|\mathrm{Pre}_{\forall}^{\boldsymbol{A}}(D)\| \leq \|D\| + \|\boldsymbol{A}\|$. For (18),

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(D) = \bigcap_{(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \boldsymbol{B}_\wedge} (\mathrm{Pre}_{\forall}^{\boldsymbol{b}_1}(D) \cup \mathrm{Pre}_{\forall}^{\boldsymbol{b}_2}(D)) \tag{21}$$

and thus $\|\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(D)\| \leq \|D\| + \|\boldsymbol{B}_\wedge\|$. Finally, regarding (19), we are here in a situation where Remark 3.3 and footnote 2 lead to a much simpler analysis:

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_+}(D) = \mathbb{N}^d \setminus (\mathrm{Pre}_{\exists\forall}^{\boldsymbol{B}_+}(\mathbb{N}^d \setminus D)) \tag{22}$$

where $\|\mathbb{N}^d \setminus D\| \leq \|D\| + 1$ when considering the norms of the minimal elements of $\mathbb{N}^d \setminus D$, and

$$\mathrm{Pre}^{\boldsymbol{B}_+}_{\exists\forall}(\mathbb{N}^d \setminus D) = \bigcup_{\boldsymbol{b} \in \boldsymbol{B}_+} \{\boldsymbol{v}_1 + \boldsymbol{v}_2 - \boldsymbol{b} \in \mathbb{N}^d \mid \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{N}^d \setminus D\} . \tag{23}$$

This last expression yields $\|\mathrm{Pre}^{\boldsymbol{B}_+}_{\exists\forall}(\mathbb{N}^d \setminus D)\| \leq 2(\|D\| + 1) + \|B_+\|$, and finally $\|\mathrm{Pre}^{\boldsymbol{B}_+}_{\forall\exists}(D)\| \leq 2\|D\| + \|B_+\| + 1$ as desired. □

5.3. **Complexity Upper Bounds.**

5.3.1. *ω-Monotonicity.* The property that allows us to deduce that the backward coverability algorithm is optimal for top-down ABVAS, and also when restricted to AVAS, is again ω-monotonicity of the downwards-closed sets that it computes. We extend for this claims 4.2 and 4.3.

Claim 5.3. The descending chains computed by the backward coverability algorithm for top-down ABVAS are ω-monotone.

Proof. Let $D_0 \supsetneq D_1 \supsetneq \cdots \supsetneq D_\ell$ be the descending chain computed for a top-down ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$. Suppose $0 \leq k < \ell - 1$ and $\boldsymbol{v}_{k+1}$ is a proper vector in the decomposition of D_{k+1}. Then $\downarrow\boldsymbol{v}_{k+1} \not\subseteq \mathrm{Pre}_{\forall\exists}(D_{k+1})$, so there is a case for each of the three types of rule.

Split: Here some $\boldsymbol{b} \in \boldsymbol{B}_+$ and $\boldsymbol{v}', \boldsymbol{v}'' \in \mathbb{N}^d_\omega$ are such that $\boldsymbol{v}_{k+1} + \boldsymbol{b} = \boldsymbol{v}' + \boldsymbol{v}''$ and $\downarrow\boldsymbol{v}', \downarrow\boldsymbol{v}'' \not\subseteq D_{k+1}$. Without loss of generality, we may assume that the ω-sets of $\boldsymbol{v}$, $\boldsymbol{v}'$ and $\boldsymbol{v}''$ are the same. Since $\downarrow\boldsymbol{v}_{k+1} \subseteq \mathrm{Pre}_{\forall\exists}(D_k)$, we have that either $\downarrow\boldsymbol{v}' \subseteq D_k$ or $\downarrow\boldsymbol{v}'' \subseteq D_k$, say the former. Let $\boldsymbol{v}_k$ be any vector in the decomposition of D_k such that $\boldsymbol{v}' \sqsubseteq \boldsymbol{v}_k$. We conclude that $\boldsymbol{v}_k$ is proper and that $\omega(\boldsymbol{v}_{k+1}) \subseteq \omega(\boldsymbol{v}_k)$.

Fork: This case is similar but easier as we may assume $\boldsymbol{v}_{k+1} + \boldsymbol{b}_1 = \boldsymbol{v}'$ and $\boldsymbol{v}_{k+1} + \boldsymbol{b}_2 = \boldsymbol{v}''$.

Unary: This case is as for VAS, cf. the proof of Claim 4.3. □

5.3.2. *Upper Bounds.* We are now equipped to establish, by applying the length function theorem for ω-monotone descending chains (Theorem 4.4) that the backward coverability algorithm for top-down ABVAS runs in Tower in general and 2ExpTime with alternation only. Since the ideal decomposition of each D_{k+1} is computable in time polynomial in the bound on the norm of D_k, it suffices to bound the number of iterations of the main loop.

Recall Equation 15, and let $h(m,n) \stackrel{\text{def}}{=} 32(n+L)(\widetilde{g^\dagger}(m,n))^2$ where

$$L \stackrel{\text{def}}{=} \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|, \|\boldsymbol{B}_+\| + 1\} \tag{24}$$

$$g^\dagger(x) \stackrel{\text{def}}{=} 2x + L. \tag{25}$$

We have

$$\widetilde{g^\dagger}(m{+}1,n) \leq (2^{\widetilde{g^\dagger}(m,n)}(n{+}L{+}1))^{m+1} \leq 2^{(\widetilde{g^\dagger}(m,n)+1)(n+L+1)(m+1)} \leq 2^{h(m,n)/4}$$

and hence

$$h(m+1,n) \leq 32(n+L)2^{h(m,n)/2} \leq 2^{5+n+L+h(m,n)/2} \leq 2^{h(m,n)},$$

so $h(d,n) \leq \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{32(n+L)}}}}}}_{d}$.

Let also $h'(m,n) = \widetilde{g^{\ddagger}}(m,n)(L'+1)(n+2)$ where $L' = \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|\}$ and $g^{\ddagger}(x) = x + L'$. We have $h'(m+1,n) \leq (h'(m,n))^{m+2}$, so $h'(d,n) \leq ((L'+1)(n+2))^{(d+1)!}$.

Theorem 4.4 gives us:

Corollary 5.4. *For any d-dimensional top-down ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ and initial downwards-closed $D_0 \subseteq \mathbb{N}^d$, the backward coverability algorithm terminates after at most $\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{32(\|D_0\|+L)}}}}}}_{d}$ steps, where $L = \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|, \|\boldsymbol{B}_+\| + 1\}$. For AVAS, i.e. when $\boldsymbol{B}_+$ is empty, the algorithm terminates after at most $((L'+1)(\|D_0\|+2))^{(d+1)!}$ steps, where $L' = \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|\}$.*

These upper bounds are optimal: top-down coverability is indeed Tower-hard for ABVAS [17] and 2ExpTime-hard for AVAS [8].

6. Bottom-Up Tree Coverability

As the last case study in this paper of the ideal view of Rackoff's technique, we consider the coverability problem for bottom-up coverability in *meet* ABVAS. We discover a backward coverability algorithm that, instead of configurations that are vectors of natural numbers as in Sections 4 and 5, works with downwards-closed sets of such vectors. The ideals involved are therefore one level higher than the vector ideals, and we recall some of their properties from the literature. We follow the pattern from the preceding sections and derive the optimal complexity bounds: Ackermann in general [17] and 2ExpTime for BVAS [10].

6.1. **Vector Set Ideals.** We first recall some properties of the downwards-closed powerset of $\mathbb{N}^d$. We write $\mathbb{D}(X)$ for the set of all subsets of X that are downwards-closed with respect to a quasi-ordering $\leq$. The subset relation is a quasi-ordering on $\mathbb{D}(X)$. Because $(\mathbb{N}^d, \sqsubseteq)$ is an ω^2-wqo (it is even a *better-quasi-order*), $(\mathbb{D}(\mathbb{N}^d), \subseteq)$ is a wqo.

Fact 6.1 ([12, 15])**.** *For any $d \in \mathbb{N}$, $(\mathbb{D}(\mathbb{N}^d), \subseteq)$ is a wqo with effective ideal representations.*

Indeed, the ideals of $(\mathbb{D}(\mathbb{N}^d), \subseteq)$ are all $\downarrow D = \{D' \in \mathbb{D}(\mathbb{N}^d) \mid D' \subseteq D\}$ for downwards-closed $D \subseteq \mathbb{N}^d$, and can thus be represented as finite sets of vectors from $\mathbb{N}^d_\omega$.

6.2. **Meet ABVAS** are defined as ABVAS (cf. Section 5), except that applications of fork rules take component-wise minima. Thus, for every non-leaf node ν in a computation:

- either ν has one child ν' and $\boldsymbol{v}(\nu) + \boldsymbol{a} = \boldsymbol{v}(\nu')$ for some unary rule $\boldsymbol{a}$ from $\boldsymbol{A}$,
- or ν has two children ν_1, ν_2 and $\boldsymbol{v}(\nu) = \min\{\boldsymbol{v}(\nu_1) - \boldsymbol{b}_1, \boldsymbol{v}(\nu_2) - \boldsymbol{b}_2\}$ for some fork rule $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ from $\boldsymbol{B}_\wedge$,

- or ν has two children ν_1, ν_2 and $\boldsymbol{v}(\nu) + \boldsymbol{b} = \boldsymbol{v}(\nu_1) + \boldsymbol{v}(\nu_2)$ for some split rule $\boldsymbol{b}$ from $\boldsymbol{B}_+$.

The *bottom-up coverability* problem (also known as the *root coverability* problem), given a d-dimensional meet ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$, and leaf and target vectors $\boldsymbol{\ell}, \boldsymbol{t} \in \mathbb{N}^d$, asks whether there exists a computation whose every leaf label is $\boldsymbol{\ell}$ and whose root label is $\sqsupseteq \boldsymbol{t}$. We remark that the problem is equivalent to the reachability problem for *gainy* ABVAS [17] but that the variant of bottom-up coverability in which fork rules are applied with exact alternation is undecidable [19].

To solve the coverability problem, we first define an operator on sets of vectors:

$$\begin{aligned}\mathrm{Post}(S) \stackrel{\text{def}}{=} &\{\boldsymbol{v} - \boldsymbol{a} \in \mathbb{N}^d \mid \boldsymbol{v} \in S \wedge \boldsymbol{a} \in \boldsymbol{A}\}\\ &\cup \{\min\{\boldsymbol{v}_1 - \boldsymbol{b}_1, \boldsymbol{v}_2 - \boldsymbol{b}_2\} \in \mathbb{N}^d \mid \boldsymbol{v}_1, \boldsymbol{v}_2 \in S \wedge (\boldsymbol{b}_1, \boldsymbol{b}_2) \in \boldsymbol{B}_\wedge\}\\ &\cup \{\boldsymbol{v}_1 + \boldsymbol{v}_2 - \boldsymbol{b} \in \mathbb{N}^d \mid \boldsymbol{v}_1, \boldsymbol{v}_2 \in S \wedge \boldsymbol{b} \in \boldsymbol{B}_+\},\end{aligned}$$

and then instantiate the generic backward algorithm from Section 3 by the following operator on sets of downward-closed sets of vectors:

$$\mathcal{PRE}(\mathcal{S}) \stackrel{\text{def}}{=} \{D \in \mathbb{D}(\mathbb{N}^d) \mid D \cup {\downarrow}\mathrm{Post}(D) \in \mathcal{S}\}\,. \tag{26}$$

6.3. **Downward-Closure and Correctness.** It is straightforward to check that $\mathcal{PRE}$ preserves the property of downward closure. Given an initial downwards-closed $\mathcal{D}_0 \subseteq \mathbb{D}(\mathbb{N}^d)$, as in Section 3, we write $\mathcal{D}_*$ for the last set in the longest (necessarily finite) descending chain of downwards-closed sets $\mathcal{D}_0 \supsetneq \mathcal{D}_1 \supsetneq \cdots$ defined by $\mathcal{D}_{k+1} = \mathcal{D}_k \cap \mathcal{PRE}(\mathcal{D}_k)$. Observing that $\mathcal{PRE}(\mathcal{D}) \subseteq \mathcal{D}$ whenever $\mathcal{D}$ is downwards-closed, we actually have $\mathcal{D}_{k+1} = \mathcal{PRE}(\mathcal{D}_k)$. From the definitions of Post and $\mathcal{PRE}$, we have the required correctness property:

Claim 6.2 (Correctness).

(a) Any $\mathcal{D}_k$ consists of all downwards-closed sets of vectors D for which there exists $D' \in \mathcal{D}_0$ such that all computations of $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ whose leaves are in D and whose height is at most k have their root in D'.

(b) The set $\mathcal{D}_*$ consists of all downwards-closed sets of vectors D for which there exists $D' \in \mathcal{D}_0$ such that all computations of $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ whose leaves are in D have their root in D'.

6.4. **Effectiveness and Control.** Let the norm of a downwards-closed subset of $\mathbb{D}(\mathbb{N}^d)$ whose ideal decomposition is ${\downarrow}D_1 \cup \cdots \cup {\downarrow}D_n$ be the maximum of the norms of $D_1, \ldots, D_n$.

Similarly to the cases of VAS and reset VAS in Sections 3.2 and 3.3, we obtain:

Claim 6.3 (Control for meet ABVAS).

(a) The ideal decomposition of $\mathcal{PRE}(\mathcal{D})$ is computable for all downwards-closed $\mathcal{D} \subseteq \mathbb{D}(\mathbb{N}^d)$.

(b) The descending chain $\mathcal{D}_0 \supsetneq \mathcal{D}_1 \supsetneq \cdots$ is (g, n)-controlled for

$$g(x) \stackrel{\text{def}}{=} x + \max\{||\boldsymbol{A}||, ||\boldsymbol{B}_\wedge||, ||\boldsymbol{B}_+||\}$$

and $n \stackrel{\text{def}}{=} \|\boldsymbol{t}\|$.

6.5. **ω-Monotonicity.** The distinction between bottom-up ABVAS and BVAS appears here. Inside the descending chain of downwards-closed subsets of $\mathbb{D}(\mathbb{N}^d)$ computed by the backward coverability algorithm, we can always find a descending chain of downwards-closed subsets of $\mathbb{N}^d$, whose length is bounded by the Ackermann function. However, for bottom-up BVAS, the latter descending chain is ω-monotone, enabling us to infer that its length is then at most doubly exponential.

Claim 6.4. Suppose $\mathcal{D}_0 \supsetneq \cdots \supsetneq \mathcal{D}_\ell$ is the descending chain computed by the backward coverability algorithm for a d-dimensional bottom-up ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ and an initial downwards-closed $\mathcal{D}_0 \subseteq \mathbb{D}(\mathbb{N}^d)$.

(a) There exist proper downwards-closed D_k in the ideal decomposition of $\mathcal{D}_k$ for all $k \in \{0, \dots, \ell-1\}$ such that

$$D_{k+1} \cup {\downarrow}\mathrm{Post}(D_{k+1}) \subseteq D_k$$

for all $k \in \{0, \dots, \ell-2\}$.

(b) For BVAS, i.e. when $\boldsymbol{B}_\wedge$ is empty, any descending chain $D_0, \dots, D_{\ell-1}$ as in (a) is ω-monotone.

Proof. For (a), pick any proper $D_{\ell-1}$ in the decomposition of $\mathcal{D}_{\ell-1}$. Assuming that D_{k+1} has been picked, we have ${\downarrow}D_{k+1} \not\subseteq \mathcal{PRE}(\mathcal{D}_{k+1})$ and thus $D_{k+1} \notin \mathcal{PRE}(\mathcal{D}_{k+1})$, so $D_{k+1} \cup {\downarrow}\mathrm{Post}(D_{k+1}) \in \mathcal{D}_k \setminus \mathcal{D}_{k+1}$. Pick any D_k in the decomposition of $\mathcal{D}_k$ such that $D_{k+1} \cup {\downarrow}\mathrm{Post}(D_{k+1}) \subseteq D_k$.

For (b), suppose $0 \leq j < \ell-2$ and $\boldsymbol{v}_{k+1}$ is a proper vector in the decomposition of D_{k+1}, i.e. ${\downarrow}\boldsymbol{v}_{k+1} \not\subseteq D_{k+2}$. Since ${\downarrow}D_{k+2}$ is in the decomposition of $\mathcal{D}_{k+2}$, we have that $D_{k+2} \cup {\downarrow}\boldsymbol{v}_{k+1} \notin \mathcal{D}_{k+2}$. But $D_{k+2} \cup {\downarrow}\boldsymbol{v}_{k+1} \subseteq D_{k+1}$, so $\mathrm{Post}(D_{k+2} \cup {\downarrow}\boldsymbol{v}_{k+1}) \not\subseteq D_{k+1}$. Recalling that $\mathrm{Post}(D_{k+2}) \subseteq D_{k+1}$, we have $\boldsymbol{w} \in \mathbb{N}^d_\omega$ and ${\downarrow}\boldsymbol{w} \not\subseteq D_{k+1}$ where:

- either $\boldsymbol{w} = \boldsymbol{v}_{k+1} - \boldsymbol{a}$ for some $\boldsymbol{a} \in \boldsymbol{A}$,
- or $\boldsymbol{w} = \boldsymbol{v}' + \boldsymbol{v}_{k+1} - \boldsymbol{b}$ for some $\boldsymbol{v}'$ in the decomposition of D_{k+2} and $\boldsymbol{b} \in \boldsymbol{B}_+$.

In both cases, ${\downarrow}\boldsymbol{w} \subseteq {\downarrow}\mathrm{Post}(D_{k+2} \cup {\downarrow}\boldsymbol{v}_{k+1}) \subseteq D_k$. Let $\boldsymbol{v}_k$ be any vector in the decomposition of D_k such that $\boldsymbol{w} \sqsubseteq \boldsymbol{v}_k$. We conclude that $\boldsymbol{v}_k$ is proper and that $\omega(\boldsymbol{v}_{k+1}) \subseteq \omega(\boldsymbol{v}_k)$. □

6.6. **Upper Bounds.** Finally, by applying the length function theorem for descending chains (Theorem 3.7), or for ω-monotone descending chains (Theorem 4.4) in the case of BVAS, we establish the Ackermann and 2ExpTime bounds. Again, the ideal decomposition of each $\mathcal{D}_{k+1}$ is computable in time polynomial in $g^k(\|\mathcal{D}_0\|)$, so it suffices to bound the number of iterations of the main loop.

Corollary 6.5. *For any d-dimensional bottom-up ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ and initial downwards-closed $\mathcal{D}_0 \subseteq \mathbb{D}(\mathbb{N}^d)$, the backward coverability algorithm terminates after at most* $\mathrm{ackermann}(p(d, L, \|\mathcal{D}_0\|))$ *steps, where p is a primitive-recursive function and $L = \max\{||\boldsymbol{A}||, ||\boldsymbol{B}_\wedge||, ||\boldsymbol{B}_+||\}$. For BVAS, i.e. when $\boldsymbol{B}_\wedge$ is empty, the algorithm terminates after at most $((L+1)(\|\mathcal{D}_0\|+2))^{(d+1)!} + 1$ steps.*

These upper bounds are tight: Ackermann-hardness was shown originally by Urquhart [28] for reachability in a variant of gainy AVAS, and also holds for bottom-up coverability for meet AVAS [17]; 2ExpTime-hardness was first shown by Demri et al. [10] for bottom-up coverability for BVAS.

7. Concluding Remarks

Rackoff's technique has successfully been employed to prove tight upper bounds for the coverability problem in VAS and extensions [10, 6, 8, 17, 16]. However, the technique does not readily generalise to more complex classes of well-structured transition systems, e.g. where configurations are not vectors of natural numbers.

We have shown that the same complexity bounds can be extracted in a principled way, by considering the ideal view of the backward coverability algorithm for VAS, and by noticing a structural invariant on its computations. Essentially the same arguments suffice to re-prove several recent upper bounds [10, 8, 17].

This paves the way for future investigations on coverability problems with large complexity gaps (where different structural invariants will need to be found). As an instance of such results, we have recently shown in [18] that the techniques presented in this paper yield new tight upper bounds for coverability in ν-Petri nets [see e.g. 23], improving on the previously known HyperAckermann upper bound.

Appendix A. Ackermann Upper Bounds

One way to obtain Ackermann upper bounds for the backward coverability algorithm on VAS and reset VAS would be to consider the dual ascending chain of *upwards-closed* sets employed in the usual description of the backward coverability algorithm. The resulting bounds would be similar (and the proof somewhat simpler), see [e.g. 26, Section 2.2.2]. Instead, we prove the bounds directly on descending chains, thanks to Theorem 3.7:

Theorem 3.7 (Length Function Theorem for Descending Chains)**.** *Let $n > 0$. Any (g, n)-controlled descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ of downwards-closed subsets of $\mathbb{N}^d$ is of length at most $h_{\omega^{d+1}}(n \cdot d!)$, where $h(x) \stackrel{def}{=} d \cdot g(x)$.*

The main tool to this end is the following statement, which combines Corollary 2.25 and Theorem 2.34 from [26]:

Theorem A.1 (Length Function Theorem for Bad Sequences)**.** *Let $n > 0$. Any (g, n)-controlled bad sequence over a polynomial normed wqo $(X, \leq, |.|_X)$ with maximal order type $o(X) < \omega^{d+1}$ is of length at most $h_{o(X)}(n \cdot d)$, where $h(x) \stackrel{def}{=} d \cdot g(x)$.*

In Section 3.3, we have already sketched how to extract a (g, n)-controlled bad sequence $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots$ of vectors from $\mathbb{N}^d_\omega$ out of a (g, n)-controlled descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ of downwards-closed subsets of $\mathbb{N}^d$. What needs to be shown in order to apply Theorem A.1 to that bad sequence is that we can use a polynomial normed wqo $(X, \leq)$ with $o(X) < \omega^{d+1}$ instead of $\mathbb{N}^d_\omega$, and derive the $h_{\omega^{d+1}}(n \cdot d!)$ bound from it. This is a routine application of the results from [26], but we shall give a detailed account for the reader's sake.

A.1. **Polynomial Normed WQOs.** Let us denote by Γ_0 the empty wqo and by Γ_1 the singleton set $\{\bullet\}$ well-quasi-ordered with equality. A *polynomial* wqo is one that can be constructed from Γ_0, Γ_1, and $\mathbb{N}$ through Cartesian products '$\times$' and disjoint unions '$+$', using respectively the product and sum orderings [26, Section 2.1.2].

A *normed* quasi-order associates a *norm* function $|.|_X \colon X \to \mathbb{N}$ to a quasi-order $(X, \leq)$, such that $X_{\leq n} \stackrel{\text{def}}{=} \{x \in X \mid |x|_X \leq n\}$ is finite for every n. For polynomial wqos, we use the zero norm on Γ_1 and the infinity norm on Cartesian products.

A.2. **Reflections.** A *normed order reflection* between two normed quasi-orders $(X, \leq_X, |.|_X)$ and $(Y, \leq_Y, |.|_Y)$ is a function $r \colon X \to Y$ such that, for all x and x' from X,

- $r(x) \leq_Y r(x')$ implies $x \leq_X x'$, and
- $|r(x)|_Y \leq |x|_X$.

If $(Y, \leq_Y, |.|_Y)$ is a normed wqo and there is a normed order reflection from $(X, \leq_X, |.|_X)$ to it, then $(X, \leq_X, |.|_X)$ is a normed wqo with (g, n)-controlled bad sequences of at most the same length [26, Proposition 2.16].

Observe that r defined by $r(\omega) \stackrel{\text{def}}{=} \bullet$ and $r(n) \stackrel{\text{def}}{=} n$ is a normed order reflection from $(\mathbb{N}_\omega, \leq, \|.\|)$ into $(\mathbb{N} + \Gamma_1, \leq, |.|)$. Hence $(\mathbb{N}_\omega^d, \sqsubseteq, \|.\|)$ reflects into

$$((\mathbb{N} + \Gamma_1)^d, \sqsubseteq, |.|) \tag{27}$$

by [26, Rem. 2.17]. This will be the polynomial normed wqo on which we will apply Theorem A.1.

A.3. **Maximal Order Types.** It remains to compute the maximal order type of $((\mathbb{N} + \Gamma_1)^d, \sqsubseteq, |.|)$. This can be done algebraically [26, Section 2.4.1] using natural sums '$\oplus$' and natural products '$\otimes$' of maximal order types:

$$o(\mathbb{N} + \Gamma_1) = \omega \oplus 1 = \omega + 1 \ , \tag{28}$$

$$o((\mathbb{N} + \Gamma_1)^d) = \underbrace{(\omega + 1) \otimes \cdots \otimes (\omega + 1)}_{d \text{ times}} = \sum_{d \geq i \geq 0} \omega^i \binom{d}{i} \ . \tag{29}$$

A.4. **Cichoń Functions.** Let us recall that, given a monotone expansive $h \colon \mathbb{N} \to \mathbb{N}$, and an ordinal α, the αth *Cichoń function* h_α is defined by induction on α by

$$h_0(x) \stackrel{\text{def}}{=} 0 \ , \qquad\qquad h_\alpha(x) \stackrel{\text{def}}{=} 1 + h_{P_x(\alpha)}(h(x)) \ , \tag{30}$$

where $P_x(\alpha) < \alpha$ denotes the predecessor ordinal at x of α, defined for $0 < \alpha < \varepsilon_0$ by:

$$P_x(\alpha + 1) \stackrel{\text{def}}{=} \alpha \ , \qquad P_x(\gamma + \omega^\beta) \stackrel{\text{def}}{=} \gamma + \omega^{P_x(\beta)} \cdot x + P_x(\omega^{P_x(\beta)}) \ . \tag{31}$$

For instance, $P_x(\omega^2) = \omega \cdot x + P_x(\omega) = \omega \cdot x + x + P_x(1) = \omega \cdot x + x$, and more generally $P_x(\omega^{d+1}) = \sum_{d \geq i \geq 0} \omega^i \cdot x$. Each Cichoń function h_α is monotone.

Proof of Theorem 3.7. By extracting proper ideals and applying Theorem A.1, we obtain an upper bound of

$$\ell = h_{o((\mathbb{N}+\Gamma_1)^d)}(n \cdot d) \tag{32}$$

on the length of (g,n)-controlled bad sequences over $\mathbb{N}^d_\omega$, and thus of $\ell+1$ on the length of (g,n)-controlled descending chains $D_0 \supsetneq D_1 \supsetneq \cdots \supsetneq D_\ell$ of downwards-closed subsets of $\mathbb{N}^d$. As (29) is quite a mouthful, we are going to over-approximate this bound with a more readable one.

Recall that any ordinal α below ω^{d+1} can be written in Cantor normal form as $\alpha = \omega^d \cdot a_d + \cdots + \omega^0 \cdot a_0$ where $a_d, \ldots, a_0$ are coefficients in $\mathbb{N}$. We can refine the *structural ordering* of [26, Eq. 2.70] for ordinals below ω^{d+1} by:

$$\omega^d \cdot a_d + \cdots + \omega^0 \cdot a_0 \sqsubseteq \omega^d \cdot b_d + \cdots + \omega^0 \cdot b_0 \text{ if } \forall 1 \leq i \leq d \,.\, a_i \leq b_i \,. \tag{33}$$

By [26, Exercise 2.11], $\alpha \sqsubseteq \beta$ ensures $h_\alpha(n) \leq h_\beta(n)$ for all n.

Observe now that (29) is such that, for all $n > 0$,

$$o((\mathbb{N}+\Gamma_1)^d) \sqsubseteq \sum_{d \geq i \geq 0} \omega^i \cdot nd! = P_{nd!}(\omega^{d+1}) \,. \tag{34}$$

Hence the result stated in Theorem 3.7, by (32) and

$$\begin{aligned} h_{o((\mathbb{N}+\Gamma_1)^d)}(nd) &\leq h_{o((\mathbb{N}+\Gamma_1)^d)}\,(h(nd!)) && \text{since } nd \leq h(nd!) \\ &\leq h_{P_{nd!}(\omega^{d+1})}\,(h(nd!)) && \text{by (34)} \\ &= h_{\omega^{d+1}}(nd!) - 1 \,. \end{aligned}$$

□

References

[1] P. A. Abdulla, K. Čerāns, B. Jonsson, and Y.-K. Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Inform. and Comput.*, 160(1/2):109–127, 2000. doi:10.1006/inco.1999.2843.

[2] P. A. Abdulla, A. Bouajjani, and J. d'Orso. Monotonic and downward closed games. *J. Logic Comput.*, 18(1):153–169, 2008. doi:10.1093/logcom/exm062.

[3] A. Arnold and M. Latteux. Récursivité et cônes rationnels fermés par intersection. *CALCOLO*, 15(4):381–394, 1978. doi:10.1007/BF02576519.

[4] M. Blondin, A. Finkel, and P. McKenzie. Handling infinitely branching WSTS. In *ICALP 2014*, volume 8573 of *LNCS*, pages 13–25, 2014. doi:10.1007/978-3-662-43951-7_2.

[5] R. Bonnet. On the cardinality of the set of initial intervals of a partially ordered set. In *Infinite and finite sets: to Paul Erdős on his 60th birthday, Vol. 1*, Coll. Math. Soc. János Bolyai, pages 189–198. North-Holland, 1975.

[6] R. Bonnet, A. Finkel, and M. Praveen. Extending the Rackoff technique to affine nets. In *FSTTCS 2012*, volume 18 of *LIPIcs*, pages 301–312. LZI, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.301.

[7] L. Bozzelli and P. Ganty. Complexity analysis of the backward coverability algorithm for VASS. In *RP 2011*, volume 6945 of *LNCS*, pages 96–109. Springer, 2011. doi:10.1007/978-3-642-24288-5_10.

[8] J.-B. Courtois and S. Schmitz. Alternating vector addition systems with states. In *MFCS 2014*, volume 8634 of *LNCS*, pages 220–231. Springer, 2014. doi:10.1007/978-3-662-44522-8_19.

[9] Ph. de Groote, B. Guillaume, and S. Salvati. Vector addition tree automata. In *LICS 2004*, pages 64–73. IEEE Computer Society, 2004. ISBN 0-7695-2192-4. doi:10.1109/LICS.2004.51.

[10] 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. doi:10.1016/j.jcss.2012.04.002.

[11] D. Figueira, S. Figueira, S. Schmitz, and Ph. Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson's Lemma. In *LICS 2011*, pages 269–278. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.39.

[12] A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In *STACS 2009*, volume 3 of *LIPIcs*, pages 433–444. LZI, 2009. doi:10.4230/LIPIcs.STACS.2009.1844.

[13] A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1–2):63–92, 2001. doi:10.1016/S0304-3975(00)00102-X.

[14] J. Goubault-Larrecq. On a generalization of a result by Valk and Jantzen. Reseach report, LSV, ENS Cachan, 2009. URL http://www.lsv.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2009-09.pdf.

[15] J. Goubault-Larrecq, P. Karandikar, K. Narayan Kumar, and Ph. Schnoebelen. The ideal approach to computing closed subsets in well-quasi-orderings. In preparation, 2016. See [14] for an earlier version.

[16] J. Kochems and C.-H. L. Ong. Decidable models of recursive asynchronous concurrency. Preprint, 2015. URL http://arxiv.org/abs/1410.8852.

[17] R. Lazić and S. Schmitz. Non-elementary complexities for branching VASS, MELL, and extensions. *ACM Trans. Comput. Logic*, 16(3:20):1–30, 2015. doi:10.1145/2733375.

[18] R. Lazić and S. Schmitz. The complexity of coverability in ν-Petri nets. Preprint, 2016. URL https://hal.inria.fr/hal-01265302.

[19] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Ann. Pure App. Logic*, 56(1–3):239–311, 1992. doi:10.1016/0168-0072(92)90075-B.

[20] R. Lipton. The reachability problem requires exponential space. Technical Report 62, Yale University, 1976.

[21] R. Majumdar and Z. Wang. Expand, enlarge, and check for branching vector addition systems. In *Concur 2013*, volume 8052 of *LNCS*, pages 152–166. Springer, 2013. doi:10.1007/978-3-642-40184-8_12.

[22] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6(2):223–231, 1978. doi:10.1016/0304-3975(78)90036-1.

[23] F. Rosa-Velardo and D. de Frutos-Escrig. Decidability and complexity of Petri nets with unordered data. *Theor. Comput. Sci.*, 412(34):4439–4451, 2011. doi:10.1016/j.tcs.2011.05.007.

[24] S. Schmitz. On the computational complexity of dominance links in grammatical formalisms. In *ACL 2010*, pages 514–524. ACL Press, 2010.

[25] S. Schmitz. Complexity hierarchies beyond Elementary. *ACM Trans. Comput. Theory*, 8(1):1–36, 2016. doi:10.1145/2858784.

[26] S. Schmitz and Ph. Schnoebelen. Algorithmic aspects of WQO theory. Lecture notes, 2012. URL http://cel.archives-ouvertes.fr/cel-00727025.

[27] Ph. Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *MFCS 2010*, volume 6281 of *LNCS*, pages 616–628. Springer, 2010. doi:10.1007/978-3-642-15155-2_54.

[28] A. Urquhart. The complexity of decision procedures in relevance logic II. *J. Symb. Log.*, 64(4):1774–1802, 1999. doi:10.2307/2586811.

DIMAP, Department of Computer Science, University of Warwick, UK

LSV, ENS Cachan & CNRS & Inria, Université Paris-Saclay, France