Information and Computation 277 (2021) 104582

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

# The ideal view on Rackoff's coverability technique [☆], [☆☆]

Ranko Lazić [a,*], Sylvain Schmitz [b]

[a] DIMAP, Department of Computer Science, University of Warwick, UK
[b] IRIF, Université de Paris, CNRS, France

ARTICLE INFO

*Article history:*
Received 1 February 2016
Received in revised form 20 June 2017
Accepted 31 October 2017
Available online 1 April 2020

ABSTRACT

Well-structured transition systems form a large class of infinite-state systems, for which safety verification is decidable thanks to a generic backward coverability algorithm. However, for several classes of systems, the generic upper bounds one can extract from the algorithm are far from optimal. In particular, in the case of vector addition systems (VAS) and several of their extensions, the known tight upper bounds were rather derived thanks to ad-hoc arguments based on Rackoff's small witness property.

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 offers a uniform setting for all well-structured transition systems, including branching ones, and we further apply it to several VAS extensions, deriving optimal upper bounds.



## 1. Introduction

*A Generic Framework for Safety Verification* One of the key tasks in automated verification is ensuring that 'nothing bad will every occur' in the system, that is checking its *safety*. Such a sound and complete check is in general impossible for systems with an infinite configuration space, caused for instance by dynamic data structures, real-time constraints, unbounded channels, parameters, or thread creation. Nevertheless, Abdulla, Čerāns, Jonsson, and Tsay [1] and Finkel and Schnoebelen [2] have identified in the 1990s an ubiquitous class of infinite-state transition systems encompassing many of these 'sources of infinity', namely effective *well-structured transition systems* (WSTS), for which a variant of safety is decidable.

In a WSTS, the configurations can be compared through a *well-quasi-ordering* (wqo) $\leq$ compatible with the transition relation. Safety in such systems can usually be expressed as a *coverability* check, where we wish to avoid both the 'bad' configuration and any larger one. In other words, we want to know given the initial configuration $x$ and the 'bad' configuration $y$, whether $x$ can *cover* $y$, i.e. reach some configuration $y' \geq y$ in finitely many steps. The generic algorithm to solve this problem is known as the *backward coverability* procedure, and 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.

☆ 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.
* Corresponding author.
*E-mail address:* R.S.Lazic@warwick.ac.uk (R. Lazić).

https://doi.org/10.1016/j.ic.2020.104582
0890-5401/

*The Complexity of Coverability* The termination of the backward coverability algorithm relies on the underlying wqo—more precisely, on the ascending chain condition for its upwards-closed subsets. In full generality, no complexity upper bounds can be extracted from the wqo alone. Nevertheless, in most algorithmic uses of wqos, one can rely on generic combinatorial analyses to extract upper bounds [3–5]. Those bounds are typically non primitive-recursive, and depend primarily on the underlying wqo. This approach has been successfully applied to the coverability problem in several classes of WSTS, and in many cases these gigantic worst-case complexity upper bounds are really a testament to the expressiveness of the corresponding classes of WSTS, as they are matched with tight lower bounds [e.g. 6–10,5].

(Un?)fortunately, these generic upper bounds are sometimes very far from optimal. Intuitively, this occurs when the operations allowed by the WSTS at hand are not able to fully exploit the underlying wqo. A striking illustration is provided by *vector addition systems* (VAS) and *reset VAS*: they work on the same wqo of tuples of vectors of non-negative integers, and the complexity upper bounds offered e.g. by Figueira et al. [3] are in Ackermann in both cases. However, while coverability in reset VAS is indeed Ackermann-complete [6], coverability in the weaker VAS model has long been known to be ExpSpace-complete thanks to a lower bound by Lipton [11] and an upper bound by Rackoff [12].

*Rackoff's Technique* is essentially combinatorial in nature: he shows by induction on the dimension of the VAS that, if $x$ can cover $y$, then there exists a small (doubly-exponential) run in the VAS witnessing the reachability of some $y' \geq y$ from $x$. A non-deterministic algorithm can then simply look for such a witness using only exponential space. While quite ad-hoc, the technique has nevertheless proven surprisingly versatile as it has been extended to prove tight complexity upper bounds for coverability in numerous extensions of VAS [13–17]. It is however far from clear how to adapt the technique for more general systems, where for instance the notion of dimension is absent or more involved, as e.g. in data nets [18] whose configurations can be seen as multisets or sequences of vectors of integers.

Remarkably, Bozzelli and Ganty [19] showed that Rackoff's small witness property can be applied to the backward coverability algorithm for VAS to obtain a 2ExpTime upper bound. In the same spirit, Majumdar and Wang [20] applied the combinatorial analysis of Demri et al. [13] to establish that the expand, enlarge, and check (EEC) algorithm for 'bottom-up' coverability in *branching VAS* runs in 2ExpTime. However, the arguments of both Bozzelli and Ganty and Majumdar and Wang use Rackoff's analysis and its bottom-up branching extension as black boxes, avoiding to work directly with the structures manipulated by the backward and EEC algorithms. Therefore, it is again unclear how those results could be translated to further classes of well-structured transition systems.

*A Generic Approach to the Complexity of Coverability* In this paper, we show how to recast Rackoff's technique in the generic setting of the backward coverability algorithm for WSTS. 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* [21–23].

This dual view of the backward coverability algorithm is merely an alternative presentation of its usual exposition using upwards-closed sets, and terminates in exactly the same number of steps. Hence so far we have not gained anything, and in fact have rather added new effectiveness assumptions on ideal representations and made the complexity analysis using wqos slightly more tedious (see Appendix A for the case of VAS and reset VAS).

Crucially, we show in Section 4 that, in the case of VAS, this dual view exhibits an additional invariant of $\omega$*-monotonicity*, which allows us to obtain the optimal doubly-exponential bound of Bozzelli and Ganty [19]. This step is not fully generic, as other invariants might need to be invented for other classes of WSTS; note however that since the original publication of this work at RP 2015, this has been achieved for $\nu$-Petri nets in [24] and invertible polynomial automata in [25].

*Generic Extensions to Branching Systems* As further proof of the versatility of the framework, we consider in the second part of the paper the top-down (Section 5) and bottom-up (Section 6) coverability problems in *branching* transition systems, whose executions are trees rather than paths, and which provide a natural setup for games and deduction systems [26,27,15,16]. *Bottom-up* coverability is one of the two natural extensions of the coverability problem to branching systems: one regards the execution trees as bottom-up and asks whether a target configuration is coverable at the root, whereas the other regards them as *top-down* and asks whether a target configuration is coverable at all leaves. While both coverability problems have been studied in *alternating branching VAS* (ABVAS) [27,13,15,16], the arguments for the corresponding complexity upper bounds were rather specialised, and we were missing a generic formulation of the backward coverability algorithm for such branching systems.

In each case, we provide a generic backward algorithm that solves the problem, and show that its running time matches the known optimal complexities in the case of ABVAS [13,15,16]. 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 and rely on the same $\omega$-monotonicity invariant. For top-down coverability in ABVAS, 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).

Our purpose in this paper is above all pedagogical, as we hope to see this type of reasoning applied more broadly where the simple proof argument of Rackoff fails. We do not prove any new upper bounds here, and rather illustrate the approach throughout sections 3 and 4 in the well-understood cases of VAS and reset VAS, and show that it scales to the

**Table 1**
The complexity of coverability problems in VAS extensions; all these upper bounds are optimal.

| Model | Complexity | Reference | In this Paper |
|---|---|---|---|
| VAS | ExpSpace | [12,19] | Corollary 4.6 |
| reset VAS | Ackermann | [3] | Corollary 3.11 |
| top-down ABVAS | Tower | [16] | Corollary 5.6 |
| top-down AVAS | 2ExpTime | [15] | Corollary 5.6 |
| bottom-up meet ABVAS | Ackermann | [16] | Corollary 6.9 |
| bottom-up BVAS | 2ExpTime | [13] | Corollary 6.9 |

more involved case of ABVAS in sections 5 and 6. We sum up the known complexity upper bounds we obtain with our approach in Table 1.

We start with some preliminaries on WSTS and ideals in the upcoming 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 pair of indices is *bad*, and necessarily finite over a wqo. See for instance [28] 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) [1,2] 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 $(X, \rightarrow)$. Finkel and Schnoebelen [2] show that weaker notions of compatibility suffice for the backward coverability algorithm, but we will stick to the classical one from Abdulla et al. [1].

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 $k+1$ 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 space of configurations $\mathbb{N}^d$ 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 $\boldsymbol{A}_{\div 2} \stackrel{\text{def}}{=} \{(-2, 1)\}$ 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 $\mathbb{Z}^d \times \mathbb{P}(\{1, \ldots, d\})$, where $\mathbb{P}$ denotes the powerset operation. Given $R \subseteq \{1, \ldots, d\}$ and a vector $\boldsymbol{u}$, we define the vector $R(\boldsymbol{u})$ by $R(\boldsymbol{u})(i) \stackrel{\text{def}}{=} 0$ if $i \in R$, and $R(\boldsymbol{u})(i) \stackrel{\text{def}}{=} \boldsymbol{u}(i)$ 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

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

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 $\ell \leq n$. Such a weak computation is impossible in a VAS [29].

More specifically, the first two vector components are used to encode two control states $(1, 0)$ and $(0, 1)$, whereas the remaining components store counters for the input $n'$, temporary $m$, and result $\ell$, respectively (see Fig. 1). Starting at $(1, 0)$, the system can increment $\ell$ only by switching the control to $(0, 1)$ and then coming back to $(1, 0)$, which necessarily resets both $n'$ and $m$. It follows that the maximum values of $\ell$ are attained by executions in which $n'$ and $m$ are minimal whenever they are reset, i.e. executions that:

- at control state $(1, 0)$, subtract 2 from $n'$ and add 1 to $m$ using the top loop as many times as possible;

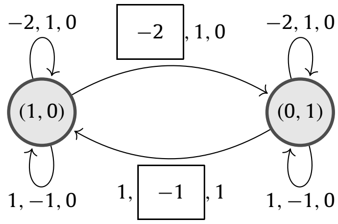


**Fig. 1.** A view of $\boldsymbol{A}_{\log}$ with its first two components seen as states. The boxed components are reset after firing the corresponding transitions.

- at control state $(0,1)$, subtract 1 from $m$ and add 1 to $n'$ using the bottom loop as many times as possible.

In particular, starting from a configuration $(1,0,2^n,0,0)$, the maximum result $\ell = n$ can be obtained by an execution in which the resets of $n'$ produce configurations of the form $(0,1,0,2^{n-\ell-1},\ell)$, and the resets of $m$ produce configurations of the form $(1,0,2^{n-\ell},0,\ell)$.

### 2.2. Ideal decompositions

The *downward-closure* of a subset $S \subseteq X$ over a wqo $(X,\leq)$ is $\downarrow S \stackrel{\text{def}}{=} \{x \in X \mid \exists s \in S\,.\, x \leq s\}$. 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 downwards-closed subset $I \subseteq X$, which is *directed*: it is non-empty, and if $x, x'$ are two elements of $I$, then there exists $y$ in $I$ with $x \leq y$ and $x' \leq y$. Alternatively, ideals are characterised as *irreducible* non-empty 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 $I \subseteq D_2$. 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 [21,22].

**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 $\omega$ 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 intersections $\downarrow\boldsymbol{u} \cap \mathbb{N}^d$ as $\boldsymbol{u}$ ranges over $\mathbb{N}_\omega^d$.

To reduce clutter, for such vectors $\boldsymbol{u}$, we shall write simply $\downarrow\boldsymbol{u}$ to denote its downward closure inside $\mathbb{N}^d$ provided that interpretation is clear from the context. For example, regarding the subset $I$ of $\mathbb{N}^3$ considered above, we have $I = \downarrow(0,\omega,0)$.

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 22,23, for more stringent requisites] if every ideal can be represented, and there are algorithms on those representations:

**(ContId)** to check $I \subseteq I'$ for two ideals $I$ and $I'$,
**(IntId)** to compute the ideal decomposition of $I \cap I'$ for two ideals $I$ and $I'$,
**(CompUp)** 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 (ContId), 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 (IntId), $\downarrow\boldsymbol{u} \cap \downarrow\boldsymbol{v} = \downarrow\boldsymbol{w}$ where $\boldsymbol{w}(i) \stackrel{\text{def}}{=} \min_{\leq}(\boldsymbol{u}(i),\boldsymbol{v}(i))$ for all $1 \leq i \leq d$. Finally, for (CompUp), if $\boldsymbol{u}$ is in $\mathbb{N}^d$, then $\mathbb{N}^d \setminus \uparrow\boldsymbol{u} = \bigcup_{1 \leq j \leq d \mid \boldsymbol{u}(j) > 0} \downarrow\boldsymbol{u}_{/j}$ where $\boldsymbol{u}_{/j}(i) = \omega$ if $i \neq j$ and $\boldsymbol{u}_{/j}(j) \stackrel{\text{def}}{=} \boldsymbol{u}(j) - 1$.

Crucially for the applicability of our approach, effective ideal representations exist for most wqos of interest [22,23].

## 3. Backward coverability

Let us recall in this section the generic backward coverability algorithm for well-structured transition systems [1,2]—the first published instance we know of this algorithm was incidentally for the case of reset VAS [30].

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:

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

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 $D_0 \supsetneq D_1 \supsetneq D_2 \supsetneq \cdots$.

We shall show that the over-approximations $D_k$ can be computed inductively on $k$ by

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

where for any set $S \subseteq X$,

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

**Remark 3.1** *(Coverability of an upwards-closed set).* It is straightforward to extend the algorithm so that it computes the set of all configurations from which a given upwards-closed set $U = \bigcup_{y \in \min U} \uparrow y$ is not reachable: simply start with $D_0 = X \setminus U = \bigcap_{y \in \min U} X \setminus \uparrow y$.

**Remark 3.2** *(Coverability witnesses).* Let us call a computation $x \rightarrow^* y'$ for $y' \geq y$ a *coverability witness*. If $x \notin D_\ell$ at some step $\ell$ of the backward coverability algorithm, then this entails that $x \rightarrow^{\leq \ell} y'$ for some $y' \geq y$ and thus that there is a coverability witness of length at most $\ell$.

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

**Claim 3.3** *(Correctness). Equations (1) and (2) define the same $D_k$.*

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

*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 3.4** *(Downward-closure). For all $k$, $D_k$ is downwards-closed.*

**Proof.** By induction on $k$. For the base case, $D_0 = X \setminus \uparrow y$ 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 $x \rightarrow z$. Then by WSTS compatibility, there exists $z' \geq z$ such that $x' \rightarrow z'$. 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)** for any downwards-closed $D$ (given by its ideal decomposition), the ideal decomposition of $\mathrm{Pre}_\forall(D)$ is computable.

This is sufficient to compute the ideal decompositions of all the $D_k$. Indeed, initially $D_0$ is computed using (CompUp). Later, $\mathrm{Pre}_\forall(D_k)$ is computable by (Pre), and its intersection with $D_k$ is also computable by (IntId). 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 (ContId).

**Remark 3.5** *(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 $U$ represented by its minimal elements.

It turns out that, under the usual effectiveness assumptions on $(X, \leq)$ 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* [31].[1]

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

We now show how the general backward algorithm can be instantiated to the coverability problems for VAS and reset VAS. Although the resulting concrete algorithms are essentially well-known, this both provides gentle examples and allows us to introduce a few operations involving vector ideals that will be useful in the sequel.

To obtain the instantiations for VAS and reset VAS, we need to prove that they satisfy the various effectiveness assumptions. Example 2.5 dealt with (ContId), (IntId), and (CompUp). By Remark 3.5, (Pre) is also effective since existential predecessors are well-known to be computable [1,2].

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 (ContId) we can then select the maximal such $\boldsymbol{v}_j$ to obtain incomparable ideals.

*Universal Predecessors in VAS* Thanks to (IntId) 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 $\boldsymbol{A}$: $\mathrm{Pre}_\forall(D) = \bigcap_{\boldsymbol{a} \in \boldsymbol{A}} \mathrm{Pre}_\forall^{\boldsymbol{a}}(D)$ where

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

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

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

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

**Example 3.6.** 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 Fig. 2.

[1] 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 $\bigcup_{1 \leq j \leq d | \boldsymbol{u}(j) < \omega} \uparrow \boldsymbol{u}_{\setminus j}$ 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 $1 \leq i \leq d$.

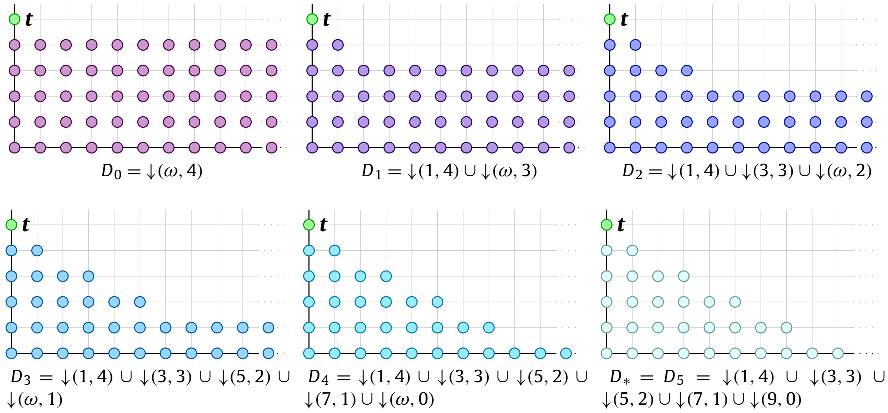


**Fig. 2.** The successive $D_k$ for $\boldsymbol{A}_{\div 2}$ with target $\boldsymbol{t} = (0, 5)$.

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

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

(where $R(\boldsymbol{v}+\boldsymbol{a})$ is defined as in Example 2.3). In order to compute a representation for this last set, given a vector $\boldsymbol{v}$ in $\mathbb{N}^d_\omega$ 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 $\omega$'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 $\downarrow\{\boldsymbol{v} \sqsupseteq \theta(\boldsymbol{a}) \mid R(\boldsymbol{v}+\boldsymbol{a}) \in \downarrow\boldsymbol{u}_j\}$ is either the empty set if $\overline{\boldsymbol{u}_j}^R \not\sqsupseteq \theta(-\boldsymbol{a})$, or $\downarrow\left(\overline{\boldsymbol{u}_j}^R - \boldsymbol{a}\right)$ otherwise.

**Example 3.7.** 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 for VAS and reset VAS*

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.

Essentially, we shall observe here that, for a suitable notion of size, the downwards-closed sets that are produced by the backward coverability algorithm for VAS and reset VAS cannot grow in an uncontrolled fashion. We shall then be able to infer bounds on the number of iterations before saturation, and thereby on the running time.

*Controlled Descending Chains* Consider some set $X$ with a size function $\|.\|: X \to \mathbb{N}$. We require that for every $n$ in $\mathbb{N}$, the set of elements of size at most $n$ in $X$ is finite ($X$ is called a *combinatorial structure*).

Given a monotone *control* function $g: \mathbb{N} \to \mathbb{N}$ and an *initial size* $n \in \mathbb{N}$, we say that a sequence $x_0, x_1, \ldots$ of elements from $X$ is $(g, n)$*-controlled* if $\|x_i\| \leq g^i(n)$ the $i$th 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, $\|(\omega, \omega)\| = 0$, $\|\{(1, \omega, 5), (0, \omega, \omega)\}\| = 5$, 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 $\|D\| \stackrel{\text{def}}{=} \|\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}\|$. Hence what is controlled in a descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ is actually its ideal representation. Similarly, for an upwards-closed set $U \subseteq \mathbb{N}^d$, we shall define its size as that of its finite basis $\|U\| \stackrel{\text{def}}{=} \max_{\boldsymbol{u} \in \min U} \|\boldsymbol{u}\|$.

**Lemma 3.8** *(Size-preserving operations). Let $\boldsymbol{u}, \boldsymbol{u}'$ be ideal representations in $\mathbb{N}_\omega^d$, $\boldsymbol{v}, \boldsymbol{v}'$ vectors in $\mathbb{N}^d$, and $\boldsymbol{a}$ a vector in $\mathbb{Z}^d$. Then*

$$\begin{aligned}
&\|{\downarrow}\boldsymbol{u} \cup {\downarrow}\boldsymbol{u}'\| \leq \max\{\|\boldsymbol{u}\|, \|\boldsymbol{u}'\|\} && \|{\uparrow}\boldsymbol{v} \cup {\uparrow}\boldsymbol{v}'\| \leq \max\{\|\boldsymbol{v}\|, \|\boldsymbol{v}'\|\}\\
&\|{\downarrow}\boldsymbol{u} \cap {\downarrow}\boldsymbol{u}'\| \leq \max\{\|\boldsymbol{u}\|, \|\boldsymbol{u}'\|\} && \|{\uparrow}\boldsymbol{v} \cap {\uparrow}\boldsymbol{v}'\| \leq \max\{\|\boldsymbol{v}\|, \|\boldsymbol{v}'\|\}\\
&\|\mathbb{N}^d \setminus {\uparrow}\boldsymbol{v}\| \leq \max\{\|\boldsymbol{v}\| - 1, 0\} && \|\mathbb{N}^d \setminus {\downarrow}\boldsymbol{u}\| \leq \|\boldsymbol{u}\| + 1\\
&\|\boldsymbol{u} + \boldsymbol{a}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{a}\| && \|\boldsymbol{v} + \boldsymbol{a}\| \leq \|\boldsymbol{v}\| + \|\boldsymbol{a}\|\,.
\end{aligned}$$

**Proof.** The inequalities for unions are immediate by definition. Regarding ideals, the inequalities for intersections and complementations follow from (IntId) and (CompUp) as detailed in Example 2.5. The remaining inequalities are obtained similarly. □

**Claim 3.9** *(Control for VAS and reset VAS). The descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ (cf. (2)) is $(g, n)$-controlled for $g(q) \stackrel{\text{def}}{=} q + \|\boldsymbol{A}\|$ and $n \stackrel{\text{def}}{=} \|\boldsymbol{t}\|$.*

**Proof.** The fact that $\|D_0\| \leq \|\boldsymbol{t}\|$ follows from ideal complementation in Lemma 3.8. Regarding the control function $g$, still by Lemma 3.8, 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, $\|\mathrm{Pre}_\forall(D)\| \leq \max_{\boldsymbol{a}} \max_{1 \leq j \leq m}(\|\mathbb{N}^d \setminus {\uparrow}\theta(\boldsymbol{a})\|, \|\boldsymbol{u}_j - \boldsymbol{a}\|)$, where Lemma 3.8 allows to conclude since $\|\theta(\boldsymbol{a})\| \leq \|\boldsymbol{a}\| \leq \|\boldsymbol{A}\|$. □

*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 $\ell$ 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 $I_k \nsubseteq D_{k+1}$. Note that for all $0 \leq j < k < \ell$, $I_j \nsubseteq I_k$, 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 for inclusion.

In the case of VAS and reset VAS, those proper ideals $I_k$ are of the form ${\downarrow}\boldsymbol{v}_k$ for some representation $\boldsymbol{v}_k$ in $\mathbb{N}_\omega^d$, and the sequence $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\ell-1}$ is also controlled by $(g, n)$ according to Claim 3.9. Using the combinatorial results from [28, 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 Appendix A.

**Theorem 3.10** *(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 $h(q) \stackrel{\text{def}}{=} d \cdot g(q)$.*

Here $h_\alpha$ for an ordinal $\alpha$ and base function $h$ denotes the $\alpha$th Cichoń function [28].

Recalling that every $D_{k+1}$ is computable from $D_k$ in polynomial time, we furthermore have that each of the $\ell$ steps of computation can be performed in time polynomial in $g^\ell(n)$. Since $g$ is primitive-recursive according to Claim 3.9, 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 [32, Corollary 4.3].

**Corollary 3.11.** *Coverability in VAS and reset VAS is in* Ackermann.

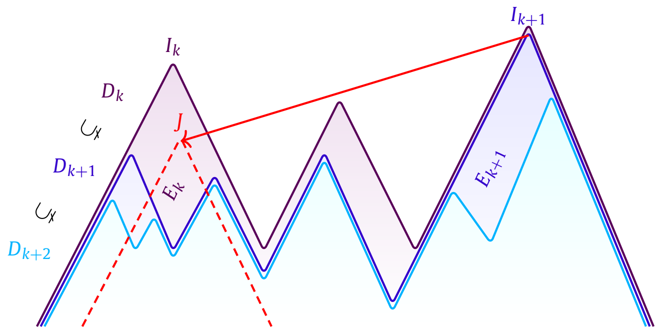


**Fig. 3.** Schematic view of the proof of Claim 4.2, where ideals are represented as cones.

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 [26,6,28].

## 4. Complexity for VAS

We know from Bozzelli and Ganty's 2ExpTime upper bound [19, Theorem 2] 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 *$\omega$-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 $\omega$-monotone are much shorter than arbitrary ones, allowing us to derive the desired 2ExpTime bound in Corollary 4.6.

### *4.1. Transitions between proper ideals*

The proof of $\omega$-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. Since the latter notion will be useful more widely, we first develop the material around it in this section, and then consider the $\omega$-monotonicity for VAS in the next.

*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. [33] 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 $\downarrow(\boldsymbol{v}+\boldsymbol{a})$, if not empty, is already an ideal. In the case of a reset VAS $\boldsymbol{A}$, we have similarly $\mathrm{Post}_\exists(\downarrow\boldsymbol{v}) = \bigcup_{(\boldsymbol{a},R)\in\boldsymbol{A}} \downarrow R(\boldsymbol{v}+\boldsymbol{a})$. Of course, to obtain decompositions, non-maximal ideals may need to be removed from those union expressions.

*Proper Transition Sequences* We can now state the result that motivates this subsection (see Fig. 3 for a picture).

**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

$$D_{k+1} = D_k \setminus \{x \in D_k \mid \exists z \notin D_k \,.\, x \to z\} \text{ for } k \geq 0. \tag{13}$$

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 $k$. Noting $E_k$ for the set of excluded elements at step $k$, we have therefore $D_{k+1} = D_k \setminus E_k$ where

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

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. □

### 4.2. $\omega$-Monotonicity

For $\boldsymbol{u}$ in $\mathbb{N}^d_\omega$, its *$\omega$-set* is the subset $\omega(\boldsymbol{u})$ of $\{1, \ldots, d\}$ 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 *$\omega$-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 $\boldsymbol{v}_{k+1} = \boldsymbol{v}_k$).

As we can see with Example 3.7 however, the descending chains computed for reset VAS are in general *not* $\omega$-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 $\omega$-set is proper in $D_2$.

**Claim 4.3** *(VAS descending chains are $\omega$-monotone). The descending chains computed by the backward coverability algorithm for VAS are $\omega$-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 (cf. Example 4.1), 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 $\omega(\boldsymbol{v}_{k+1}) \subseteq \omega(\boldsymbol{v}_k)$. □

### 4.3. Upper bound

We are now in position to state a refinement of Theorem 3.10 for $\omega$-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 $\omega$-monotone descending chains). Any $(g, n)$-controlled $\omega$-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 $D_\ell$ 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 $\ell$. We are going to establish by induction on $d - |\Omega|$ that, if $\downarrow\boldsymbol{v}_k$ is a proper ideal from the decomposition of $D_k$ and its $\omega$-set is $\Omega$, then $k < \widetilde{g}(d - |\Omega|, n)$, which by monotonicity of $\widetilde{g}$ in its first argument entails $k < \widetilde{g}(d, n)$ as desired.

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

**Claim 4.5.** *Let $\Omega \subseteq \{1, \ldots, d\}$ and $k < k'$ be such that:*

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

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

**Proof of Claim 4.5.** See Fig. 4 for a depiction of the main arguments.

From assumption (ii), there exists a proper ideal $\downarrow\boldsymbol{v}_{k'}$ in the decomposition of $D_{k'}$ with $\Omega = \omega(\boldsymbol{v}_{k'})$. By $\omega$-monotonicity, for every $j = k' - 1, k' - 2, \ldots, k + 1$ we can find a proper ideal $\downarrow\boldsymbol{v}_j$ in the decomposition of $D_j$ such that $\Omega = \omega(\boldsymbol{v}_{k'}) \subseteq \omega(\boldsymbol{v}_{k'-1}) \subseteq \cdots \subseteq \omega(\boldsymbol{v}_{k+1})$. Due to assumption (i), these inclusions cannot be strict, hence

$$\Omega = \omega(\boldsymbol{v}_{k'}) = \omega(\boldsymbol{v}_{k'-1}) = \cdots = \omega(\boldsymbol{v}_{k+1})\;. \tag{16}$$

Since they are proper, those $k' - k$ vectors are mutually distinct.

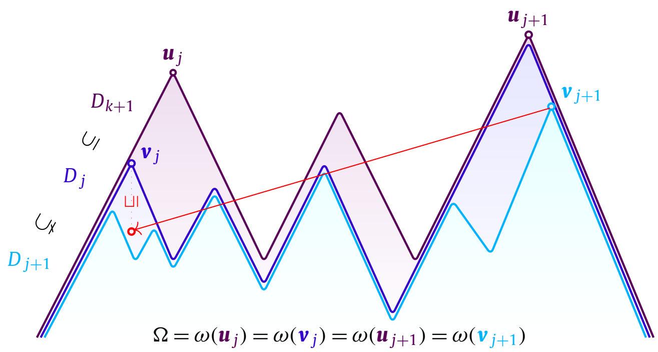


**Fig. 4.** Schematic view of the proof of Claim 4.5 for $j \in \{k+1, \ldots, k'-1\}$. The red connection witnesses $\omega$-monotonicity: $\boldsymbol{v}_{j+1} \to \boldsymbol{v}_{j+1} + \boldsymbol{a} \sqsubseteq \boldsymbol{v}_j$ for some $\boldsymbol{a} \in \boldsymbol{A}$ by Claim 4.2, and this implies $\omega(\boldsymbol{v}_{j+1}) \subseteq \omega(\boldsymbol{v}_j)$. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

Consider any such $\boldsymbol{v}_j$ with $j \in \{k+1, \ldots, k'\}$; we shall prove that

$$\|\boldsymbol{v}_j\| \le \|D_{k+1}\| \,. \tag{17}$$

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

$$\boldsymbol{v}_j \sqsubseteq \boldsymbol{u}_j \,, \tag{18}$$

which implies $\omega(\boldsymbol{v}_j) \subseteq \omega(\boldsymbol{u}_j)$. We claim that

$$\omega(\boldsymbol{u}_j) = \omega(\boldsymbol{v}_j) = \Omega \,. \tag{19}$$

Indeed, either $\boldsymbol{u}_j = \boldsymbol{v}_j$ and (19) holds, or because $\downarrow\boldsymbol{v}_j$ is proper at $D_j$, we have that $\downarrow\boldsymbol{u}_j$ is proper at $D_{j'}$ for some $j' \in \{k+1, \ldots, j-1\}$. Hence (19) holds, since otherwise we would have $\Omega = \omega(\boldsymbol{v}_j) \subsetneq \omega(\boldsymbol{u}_j)$, contradicting assumption (i) on $j'$. Therefore, by (18) and (19), $\|\boldsymbol{v}_j\| \le \|\boldsymbol{u}_j\| \le \|D_{k+1}\|$ and (17) holds.

To conclude, note that there can be at most $(\|D_{k+1}\|+1)^{(d-|\Omega|)}$ mutually distinct vectors in $\mathbb{N}_\omega^d$ with $\Omega$ as $\omega$-set (cf. (16)) and norm bounded by $\|D_{k+1}\|$ (cf. (17)). $\square$

Finally, putting together Claim 3.9 (control for VAS), Claim 4.3 ($\omega$-monotonicity), and Theorem 4.4 (lengths of controlled $\omega$-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 $h(m,n) \stackrel{\text{def}}{=} \widetilde{g}(m,n)(\|\boldsymbol{A}\|+1)(n+2)$ where $g(q) = q + \|\boldsymbol{A}\|$ as in Claim 3.9. We have

$$\begin{aligned} h(m+1,n) &= \widetilde{g}(m+1,n)(\|\boldsymbol{A}\|+1)(n+2) \\ &= \left(\widetilde{g}(m,n) + (g^{\widetilde{g}(m,n)}(n)+1)^{m+1}\right)(\|\boldsymbol{A}\|+1)(n+2) \\ &= \left(\widetilde{g}(m,n) + (\widetilde{g}(m,n)\|\boldsymbol{A}\|+n+1)^{m+1}\right)(\|\boldsymbol{A}\|+1)(n+2) \\ &\le (2\widetilde{g}(m,n)\|\boldsymbol{A}\|+n+1)^{m+1}(\|\boldsymbol{A}\|+1)(n+2) \\ &\le (h(m,n))^{m+2} \,, \end{aligned}$$

so $\widetilde{g}(m,n) \le h(m,n) \le ((\|\boldsymbol{A}\|+1)(n+2))^{(m+1)!}$, which applies to the backward coverability algorithm for $n = \|\boldsymbol{t}\|$ and $m = d$. $\square$

As mentioned in the introduction, this 2ExpTime upper bound is not tight for the coverability problem for VAS—which is ExpSpace-complete [11,12]. However, the bound is tight for the backward coverability algorithm: as shown by Bozzelli and Ganty [19, Corollary 2] 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.

By Remark 3.2, the double exponential upper bound on the length of the descending chain $D_0 \supsetneq D_1 \supsetneq \cdots$ constructed by the backward coverability algorithm for VAS translates into a bound on the length of the shortest coverability witnesses—this is a converse to the result of Bozzelli and Ganty. Then, the ExpSpace complexity upper bound of Rackoff [12] can be recovered using a nondeterministic algorithm that guesses and checks the existence of such a witness.

## 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 [16] and 2ExpTime with alternation only [15]. 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 show that the backward coverability algorithm has optimal complexity.

We start this section by extending the notion of well-structured transition systems to top-down tree computations (Section 5.1) and adapting the (dual) backward coverability algorithm to the top-down setting (Section 5.2). We then recall the definition of alternating branching VAS from [16] in Section 5.3, and show in Section 5.4 that their instantiation of the dual backward coverability algorithm enjoys the same $\omega$-monotonicity as in the case of VAS, from which we derive the known upper bounds from [15,16].

### 5.1. Top-down well-structured transition systems

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* [34], which we shall exploit next in Section 5.2.

*Branching transition systems* $(X, \to)$ allow their transitions to be branching, i.e. to map configurations to sets of configurations: $\to \subseteq X \times \mathbb{P}(X)$ where $\mathbb{P}(X)$ denotes the powerset of $X$. Intuitively, a computation in a branching transition system starting from some initial configuration $x$ is an $X$-labelled tree with root label $x$, such that every node labelled $y$ has set of child labels $S$ for some $y \to S$.

We adapt the notations of Section 2.1 by lifting $\to$ to a relation $\rightrightarrows$ between sets of configurations: $S \rightrightarrows S'$ if and only if for all $x$ in $S$ there exists $S_x \subseteq S'$ with $x \to S_x$. The notations $\rightrightarrows^{\leq k}$ and $\rightrightarrows^*$ are then defined as usual for the lifted relation $\rightrightarrows$; for instance, $S \rightrightarrows^{\leq k} S'$ if and only if, for every $x$ in $S$, there is a computation tree of height at most $k$ with root label $x$ and leaf labels included in $S'$.

*Top-Down Compatibility* In order to define top-down WSTS, the compatibility condition must be lifted to work with sets of configurations, and we use the Smyth ordering to this end. More precisely, let $(X, \leq)$ be a quasi-order; we define the *Smyth* quasi-order $(\mathbb{P}(X), \subseteq_{\mathbb{S}})$ by $S \subseteq_{\mathbb{S}} S'$ for $S, S' \in \mathbb{P}(X)$ if, for all $x' \in S'$, there exists $x \in S$ such that $x \leq x'$.

**Definition 5.1** *(Top-down WSTS).* A *top-down WSTS* is a triple $(X, \to, \leq)$ where $(X, \to)$ is a branching transition system and $(X, \leq)$ is a wqo with the following *top-down* compatibility condition: if $x \leq x'$ and $x \to S$, then there exists $S' \supseteq_{\mathbb{S}} S$ with $x' \to S'$.

Note that we do not require $\to$ to map to finite sets, nor $(\mathbb{P}(X), \subseteq_{\mathbb{S}})$ to be a wqo (which it might not be in general; see [35] and the discussion in Section 6.1).

### 5.2. Backward top-down coverability

The *top-down coverability* problem (also called the *leaf coverability* problem), given an ordered branching transition system $(X, \to, \leq)$ and root and leaf configurations $x, y \in X$, asks whether there exists a finite computation tree whose root label is $x$ and whose every leaf label is $\geq y$. Equivalently, it asks whether $\{x\} \rightrightarrows^* \uparrow y$.

*Game Viewpoint* A top-down coverability instance defines a two-player game over an infinite arena with positions in $X$. Starting from $x$, Player 1 attempts to cover $y$, while Player 2 attempts to foil it. At each round, in a configuration $z$,

1. Player 1 chooses an applicable transition $z \to S$
2. Player 2 then chooses a branch, i.e. some $z' \in S$, and the game proceeds to the next round from $z'$.

Player 1 wins the game if a configuration $z \geq y$ is eventually reached. One can see that computation trees represent strategies for Player 1, and such a tree has all its leaves $\geq y$ if and only if it is a winning strategy.

*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 winning region for Player 2 is downwards-closed. In our dual view of the backward top-down 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 $z$ then belongs to $D_{k+1}$ if, for all choices of rules by Player 1, there exists a move to a configuration $z'$ 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 from (3) with

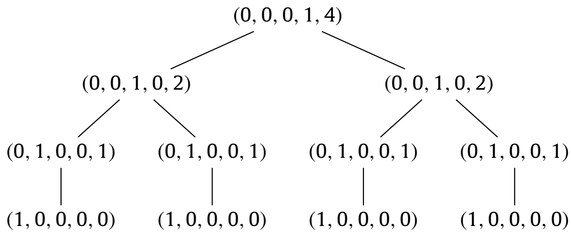


**Fig. 5.** A computation tree for the BVAS of Example 5.3.

$$\mathrm{Pre}_{\forall\exists}(S) \stackrel{\text{def}}{=} \{x \in X \mid \forall S' \subseteq X \,.\, x \to S' \implies S' \cap S \neq \emptyset\}\,. \tag{20}$$

*Downward-Closure and Correctness* It is straightforward to check that $\mathrm{Pre}_{\forall\exists}$ preserves the property of downward-closure as in Claim 3.4. Given the initial downwards-closed $D_0 \stackrel{\text{def}}{=} X \setminus \uparrow y$, 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 $D_{k+1} \stackrel{\text{def}}{=} D_k \cap \mathrm{Pre}_{\forall\exists}(D_k)$. From the definition of $\mathrm{Pre}_{\forall\exists}$, we have the required correctness property, matching Claim 3.3:

**Claim 5.2** *(Correctness). For all $k$, $D_k = \{x \in X \mid \{x\} \not\Rightarrow^{\le k} \uparrow y\}$.*

Thus, the set $D_*$ consists of all configurations $x \in X$ such that all computation trees of $(X, \to, \le)$ whose root label is $x$ have some leaf label in $D_0$. This generic procedure can be turned into an algorithm under the same effectiveness assumptions as in Section 3, where (Pre) now requires the ideal decomposition of $\mathrm{Pre}_{\forall\exists}(D)$ to be computable from the ideal decomposition of $D$.

### 5.3. Alternating branching VAS

Recall that an *alternating branching vector addition system (ABVAS)* of dimension $d \in \mathbb{N}$ is a triple $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ where $\boldsymbol{A} \subseteq \mathbb{Z}^d$ is a finite set of *unary* rules, $\boldsymbol{B}_\wedge \subseteq \mathbb{Z}^d \times \mathbb{Z}^d$ is a finite set of *fork* rules, and $\boldsymbol{B}_+ \subseteq \mathbb{Z}^d$ is a finite set of *split* rules.

An ABVAS defines a top-down WSTS $(\mathbb{N}^d, \to, \sqsubseteq)$ where the transition relation $\to$ is the union of

- unary transitions $\boldsymbol{u} \to \{\boldsymbol{u} + \boldsymbol{a}\}$ for some $\boldsymbol{a}$ in $\boldsymbol{A}$,
- fork transitions $\boldsymbol{u} \to \{\boldsymbol{u} + \boldsymbol{b}_1, \boldsymbol{u} + \boldsymbol{b}_2\}$ for some $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ in $\boldsymbol{B}_\wedge$, and
- split transitions $\boldsymbol{u} \to \{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ where $\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{u}_1 + \boldsymbol{u}_2$ for some $\boldsymbol{b}$ in $\boldsymbol{B}_+$ with $\boldsymbol{u} + \boldsymbol{b}$ in $\mathbb{N}^d$.

Recall that all the vectors $\boldsymbol{u}$, $\boldsymbol{u} + \boldsymbol{a}$, $\boldsymbol{u} + \boldsymbol{b}_1$, $\boldsymbol{u} + \boldsymbol{b}_2$, $\boldsymbol{u}_1$, and $\boldsymbol{u}_2$ must belong to $\mathbb{N}^d$. We can easily check that it satisfies top-down compatibility.

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. [36,37,15,16].

**Example 5.3.** Consider the BVAS $(\boldsymbol{A}, \emptyset, \boldsymbol{B}_+)$ defined by

$$\boldsymbol{A} \stackrel{\text{def}}{=} \{(1, -1, 0, 0, -1)\} \qquad \boldsymbol{B}_+ \stackrel{\text{def}}{=} \{(0, 0, 1, -1, 0), (0, 1, -1, 0, 0)\}\,.$$

Observe that the only way to cover $(1, 0, 0, 0, 0)$ from $(0, 0, 0, 1, n)$ is to start initially with $n \ge 4$; this corresponds to the computation tree represented in Fig. 5.

The top-down coverability problem in ABVAS is equivalent to top-down coverability of ABVAS with *join* semantics, in which the fork rules are applied by taking pointwise maxima (i.e. $\boldsymbol{u} \to \{\boldsymbol{u}_1 + \boldsymbol{b}_1, \boldsymbol{u}_2 + \boldsymbol{b}_2\}$ for $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ in $B_\wedge$ if $\boldsymbol{u} = \max\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ and $\boldsymbol{u}$, $\boldsymbol{u}_1$, $\boldsymbol{u}_1 + \boldsymbol{b}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_2 + \boldsymbol{b}_2$ are in $\mathbb{N}^d$), and also to the reachability problem for *lossy* ABVAS [16].

### 5.4. Complexity for ABVAS

We instantiate now the backward top-down coverability algorithm of Section 5.2 to the case of top-down ABVAS. Similarly to the cases of VAS and reset VAS in sections 3.2 and 3.3, we shall start by giving an explicit computation for predecessors, allowing us to provide bounds on the sizes of ideals in the course of the backward top-down coverability algorithm. The next step will be to show that $\omega$-monotonicity also holds for top-down ABVAS.

*Effectiveness and Control* Note that for every $n$, there are finitely many vectors $\boldsymbol{v}$ in $\mathbb{N}_\omega^d$ of size at most $n$. Thus the effectiveness of $\mathrm{Pre}_{\forall\exists}$ is a consequence of its control, for which we have the following counterpart to Claim 3.9.

**Claim 5.4** *(Control for top-down ABVAS). 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.** First observe from the definition of $\mathrm{Pre}_{\forall\exists}$ in (20) that

$$\mathrm{Pre}_{\forall\exists}(S) = \mathrm{Pre}_{\forall}^{\boldsymbol{A}}(S) \cap \mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(S) \cap \mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_+}(S) \tag{21}$$

where

$$\mathrm{Pre}_{\forall}^{\boldsymbol{A}}(S) \stackrel{\text{def}}{=} \{\boldsymbol{v} \in \mathbb{N}^d \mid \forall \boldsymbol{a} \in \boldsymbol{A} \,.\, \boldsymbol{v} + \boldsymbol{a} \in \mathbb{N}^d \implies \boldsymbol{v} + \boldsymbol{a} \in S\}\,, \tag{22}$$

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(S) \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{N}^d \;\middle|\; \begin{array}{c} \forall(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \boldsymbol{B}_\wedge \,.\, \boldsymbol{v} + \boldsymbol{b}_1 \in \mathbb{N}^d \wedge \boldsymbol{v} + \boldsymbol{b}_2 \in \mathbb{N}^d \\ \implies \boldsymbol{v} + \boldsymbol{b}_1 \in S \vee \boldsymbol{v} + \boldsymbol{b}_2 \in S \end{array} \right\}, \tag{23}$$

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_+}(S) \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{N}^d \;\middle|\; \begin{array}{c} \forall \boldsymbol{b} \in \boldsymbol{B}_+ . \forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{N}^d \,.\, \boldsymbol{v} + \boldsymbol{b} = \boldsymbol{v}_1 + \boldsymbol{v}_2 \\ \implies \boldsymbol{v}_1 \in S \vee \boldsymbol{v}_2 \in S \end{array} \right\}. \tag{24}$$

Let us examine the three cases in (21); by (IntId) for vector ideals, the size of the ideals in $\mathrm{Pre}_{\forall\exists}(D)$ will be bounded by the maximal size over all three cases. For (22),

$$\mathrm{Pre}_{\forall}^{\boldsymbol{A}}(D) = \bigcap_{\boldsymbol{a} \in \boldsymbol{A}} \mathrm{Pre}_{\forall}^{\boldsymbol{a}}(D) \tag{25}$$

as defined in (6), entailing $\|\mathrm{Pre}_{\forall}^{\boldsymbol{A}}(D)\| \le \|D\| + \|\boldsymbol{A}\|$ as seen in Claim 3.9. For (23),

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(D) = \bigcap_{(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \boldsymbol{B}_\wedge} (\mathrm{Pre}_{\forall}^{\boldsymbol{b}_1}(D) \cup \mathrm{Pre}_{\forall}^{\boldsymbol{b}_2}(D)) \tag{26}$$

and thus $\|\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_\wedge}(D)\| \le \|D\| + \|\boldsymbol{B}_\wedge\|$. Finally, regarding (24), we are here in a situation where Remark 3.5 and footnote 1 lead to a much simpler analysis:

$$\mathrm{Pre}_{\forall\exists}^{\boldsymbol{B}_+}(D) = \mathbb{N}^d \setminus (\mathrm{Pre}_{\exists\forall}^{\boldsymbol{B}_+}(\mathbb{N}^d \setminus D)) \tag{27}$$

where $\|\mathbb{N}^d \setminus D\| \le \|D\| + 1$ by Lemma 3.8, and

$$\mathrm{Pre}_{\exists\forall}^{\boldsymbol{B}_+}(\mathbb{N}^d \setminus D) = \bigcup_{\boldsymbol{b} \in \boldsymbol{B}_+} \{\boldsymbol{v}_1 + \boldsymbol{v}_2 - \boldsymbol{b} \in \mathbb{N}^d \mid \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{N}^d \setminus D\}\,. \tag{28}$$

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

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

**Claim 5.5** *(Top-Down ABVAS descending chains are $\omega$-monotone). The descending chains computed by the backward top-down coverability algorithm for ABVAS are $\omega$-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 $0 \le k < \ell - 1$ 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 rules in (21).

**Split** Here some $\boldsymbol{b} \in \boldsymbol{B}_+$ and $\boldsymbol{v}', \boldsymbol{v}'' \in \mathbb{N}_\omega^d$ 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 $\omega$-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. □

*Upper Bounds* We are now equipped to establish, by applying the length function theorem for $\omega$-monotone descending chains (Theorem 4.4) that the backward top-down 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 size of $D_k$, it suffices to bound the number of iterations of the main loop.

**Corollary 5.6.** *For any $d$-dimensional ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ and target vector $\boldsymbol{t}$, the backward top-down coverability algorithm terminates after at most $\underbrace{2^{\cdot^{\cdot^{2}}}}_{d}{}^{32(\|\boldsymbol{t}\|+L)}$ steps, where $L \stackrel{\text{def}}{=} \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|, \|\boldsymbol{B}_+\| + 1\}$.*

*For AVAS, i.e. when $\boldsymbol{B}_+$ is empty, the algorithm terminates after at most $((L'+1)(\|\boldsymbol{t}\|+2))^{(d+1)!}$ steps, where $L' \stackrel{\text{def}}{=} \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|\}$.*

**Proof.** As top-down ABVAS have controlled $\omega$-monotone chains, we can apply Theorem 4.4. Recall Equation (15), and let us over-approximate the control function from Claim 5.4 by $g^\dagger(q) \stackrel{\text{def}}{=} 2q + L$. Define $h(m,n) \stackrel{\text{def}}{=} 32(n+L)(\widetilde{g}^\dagger(m,n))^2$; then

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

hence

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

so

$$\widetilde{g}^\dagger(d,n) \le h(d,n) \le \underbrace{2^{\cdot^{\cdot^{2}}}}_{d}{}^{32(n+L)} .$$

Regarding top-down AVAS, let also $h'(m,n) \stackrel{\text{def}}{=} \widetilde{g}^\ddagger(m,n)(L'+1)(n+2)$ where $g^\ddagger(q) \stackrel{\text{def}}{=} q + L'$. We have $h'(m+1,n) \le (h'(m,n))^{m+2}$ as in Corollary 4.6, so $\widetilde{g}^\ddagger(d,n) \le h'(d,n) \le ((L'+1)(n+2))^{(d+1)!}$. $\square$

These upper bounds are optimal: top-down coverability is indeed Tower-hard for ABVAS [16] and 2ExpTime-hard for AVAS [15].

## 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. As in Section 5, we start this section by extending the notion of well-structured transition systems to bottom-up tree computations in Section 6.1. It turns out that bottom-up coverability in a branching transition system reduces to plain coverability in a 'powerset' transition system, and therefore that the algorithm from Section 3 can be instantiated as such.

Nevertheless, we show in Sections 6.1 and 6.2 how to transcribe in a generic manner most of the ingredients of the backward coverability algorithm on this powerset system in terms of the original branching system. In particular, the complexity analysis does not need to consider descending chains of downwards-closed sets of sets, as we can extract descending chains of downwards-closed sets (Remark 6.6).

This allows to derive in Section 6.3 the known optimal Ackermann upper bounds for bottom-up coverability in meet ABVAS [16], and even 2ExpTime upper bounds in the case of BVAS [13], as the extracted descending sequences are $\omega$-monotone in that case.

### 6.1. Bottom-up well-structured transition systems

We consider again branching transition systems $(X, \to, \le)$ as defined in Section 5.1, where $\to \subseteq X \times \mathbb{P}(X)$. When we consider a branching transition system bottom-up, it is enlightening to think of it as a deduction system with 'propositions' in $X$, where a transition $x \to S$ means that $x$ can be deduced from the set $S$. This intuition provides another way of lifting the transition relation $\to$ to operate on subsets of $X$, this time proceeding bottom-up in computation trees: we write

$$S \vdash \hat{S} \text{ if and only if } \hat{S} = S \cup \{x \in X \mid \exists S_x \subseteq S \,.\, x \to S_x\} . \tag{29}$$

Observe that this new relation is actually a function. It maps the set $S$ to the elements that can be deduced from it in one or fewer steps, i.e. $\vdash = \vdash^{\le 1}$. Thus $S \vdash^* S'$ if and only if $S'$ is exactly the set of elements that root computation trees with leaf labels in $S$.

*Quasi-Ordering Powersets* Let $(X, \le)$ be a quasi-order. We define the *Hoare* quasi-order $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$ by $S \subseteq_{\mathbb{H}} S'$ for $S, S' \in \mathbb{P}(X)$ if, for all $x \in S$, there exists $x' \in S'$ such that $x \le x'$. Observe that this ordering reduces to a simple inclusion when $S$ and $S'$ are downwards-closed; in fact, $S \subseteq_{\mathbb{H}} S'$ if and only if $\downarrow S \subseteq \downarrow S'$. This is a sort of dual of the Smyth quasi-order, as $S \subseteq_{\mathbb{H}} S'$ if and only if $X \setminus (\downarrow S) \subseteq_{\mathbb{S}} X \setminus (\downarrow S')$.

Both the Hoare and the Smyth quasi-orders might fail to be wqo when $(X, \le)$ is a wqo. In fact, they are wqos precisely when $(X, \le)$ is an $\omega^2$-*wqo* [35, Corollary 12]. Fortunately, all the wqos used in the WSTS literature, including $(\mathbb{N}^d, \le)$, are $\omega^2$-wqos.

*Bottom-Up Coverability* Also known as the *root coverability* problem, the *bottom-up* coverability problem, given an ordered branching transition system $(X, \to, \le)$ and root and leaf configurations $y, x \in X$, asks whether there exists a finite computation tree whose root label is $\ge y$ and whose every leaf label is $x$. Equivalently, it asks whether some $y' \ge y$ can be deduced from the single $x$, i.e. whether $\{x\} \vdash^* S$ where $S \cap \uparrow y \neq \emptyset$.

The Hoare quasi-order offers another view of bottom-up coverability: for a root label $y$ and a leaf label $x$, the question becomes whether there exists $S \in \mathbb{P}(X)$ such that $\{y\} \subseteq_{\mathbb{H}} S$ and $\{x\} \vdash^* S$. In other words, bottom-up coverability is an instance of plain coverability, from $\{x\}$ to $\{y\}$, in the 'powerset' transition system $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$, a deterministic, non-branching transition system.

*Bottom-Up Compatibility* In the light of the previous remark, we should define a notion of bottom-up compatibility that captures exactly the usual compatibility in $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$.

**Definition 6.1** *(Bottom-Up WSTS).* A *bottom-up WSTS* is a triple $(X, \to, \le)$ where $(X, \to)$ is a branching transition system and $(X, \le)$ is an $\omega^2$-*wqo* with the following *bottom-up* compatibility condition: if $S \subseteq_{\mathbb{H}} S'$ and $x \to S$, then there exist $x' \ge x$ such that: $x' \in S'$ or there exists $S'_{x'} \subseteq S'$ with $x' \to S'_{x'}$.

We can check that the above definition is exactly the needed one.

**Proposition 6.2.** *An ordered branching transition system $(X, \to, \le)$ is a bottom-up WSTS if and only if $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$ is a WSTS.*

**Proof.** As already mentioned, $(X, \le)$ is $\omega^2$-wqo if and only if $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$ is wqo [35, Corollary 12].

Regarding compatibility, first assume that $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$ enjoys compatibility and assume $S \subseteq_{\mathbb{H}} S'$ and $x \to S$. Let $\hat{S}$ and $\hat{S}'$ be the sets such that $S \vdash \hat{S}$ and $S' \vdash \hat{S}'$. By definition of $\hat{S}$, since $x \to S$, $x \in \hat{S}$. By compatibility of $\vdash$, $\hat{S} \subseteq_{\mathbb{H}} \hat{S}'$, therefore there exists $x' \in \hat{S}'$ such that $x \le x'$. By definition of $\hat{S}'$, this entails that $x' \in S'$, or that there exists $S'_{x'} \subseteq S'$ such that $x' \to S'_{x'}$. This shows that $(X, \to, \le)$ is bottom-up compatible.

Conversely, assume that $(X, \to, \le)$ is bottom-up compatible and assume $S \subseteq_{\mathbb{H}} S'$. Let $\hat{S}$ and $\hat{S}'$ be the sets such that $S \vdash \hat{S}$ and $S' \vdash \hat{S}'$; we want to show that $\hat{S} \subseteq_{\mathbb{H}} \hat{S}'$. Consider for this any $x \in \hat{S}$. Two cases arise by definition of $\hat{S}$. If $x \in S$, then because $S \subseteq_{\mathbb{H}} S'$ there exists $x' \ge x$ in $S' \subseteq \hat{S}'$ as desired. Otherwise, there exists $S_x \subseteq S$ such that $x \to S_x$. As $S_x \subseteq S \subseteq_{\mathbb{H}} S'$, $S_x \subseteq_{\mathbb{H}} S'$, hence bottom-up compatibility applies and there exists $x' \ge x$ with two possible cases: if $x' \in S' \subseteq \hat{S}'$ we are done; otherwise there exists $S'_{x'} \subseteq S'$ such that $x' \to S'_{x'}$ and this in turn implies that $x' \in \hat{S}'$ and concludes the proof. □

**Example 6.3** *(ABVAS are* not *Bottom-Up WSTS).* Alternating branching VAS as defined in Section 5.3 are in general not bottom-up WSTS. For instance, consider the 2-dimensional ABVAS $(\emptyset, \boldsymbol{B}_\wedge, \emptyset)$ where $\boldsymbol{B}_\wedge \stackrel{\text{def}}{=} \{((0, -2), (-1, 0))\}$. Then we have the fork transition $(1, 2) \to \{(1, 0), (0, 2)\}$, but $\{(1, 0), (0, 2)\} \subseteq_{\mathbb{H}} \{(1, 1), (0, 2)\} \not\supseteq_{\mathbb{H}} \{(1, 2)\}$, and there is no configuration $\boldsymbol{v} \in \mathbb{N}^2$ such that $\boldsymbol{v} \to S'_{\boldsymbol{v}} \subseteq \{(1, 1), (0, 2)\}$.

In fact, bottom-up coverability in ABVAS is undecidable [38].

**Example 6.4** *(Meet ABVAS are bottom-up WSTS).* A *meet ABVAS* is defined as an ABVAS (cf. Section 5.3), except that applications of fork rules take component-wise minima. Thus fork transitions are now defined as $\boldsymbol{u} \to \{\boldsymbol{u}_1 + \boldsymbol{b}_1, \boldsymbol{u}_2 + \boldsymbol{b}_2\}$ if $\boldsymbol{u} = \min\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ for some $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ in $\boldsymbol{B}_\wedge$ and $\boldsymbol{u}, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_1 + \boldsymbol{b}_1, \boldsymbol{u}_2 + \boldsymbol{b}_2$ are all in $\mathbb{N}^d$.

Regarding bottom-up compatibility, unary rules are compatible as in the case of VAS. For fork rules, observe that if $\boldsymbol{u} \to \{\boldsymbol{u}_1 + \boldsymbol{b}_1, \boldsymbol{u}_2 + \boldsymbol{b}_2\}$ as the result of a rule $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ in $\boldsymbol{B}_\wedge$ and $S' \supseteq_{\mathbb{H}} \{\boldsymbol{u}_1 + \boldsymbol{b}_1, \boldsymbol{u}_2 + \boldsymbol{b}_2\}$, then by definition of the Hoare quasi-order there exist $S'_{\boldsymbol{u}'} \stackrel{\text{def}}{=} \{\boldsymbol{u}'_1 + \boldsymbol{b}_1, \boldsymbol{u}'_2 + \boldsymbol{b}_2\} \subseteq S'$ with $\boldsymbol{u}_1 \le \boldsymbol{u}'_1$ and $\boldsymbol{u}_2 \le \boldsymbol{u}'_2$, and $\boldsymbol{u}' \stackrel{\text{def}}{=} \min\{\boldsymbol{u}'_1, \boldsymbol{u}'_2\}$ is such that $\boldsymbol{u}' \to S'_{\boldsymbol{u}'}$ and $\boldsymbol{u} \le \boldsymbol{u}'$ as required.

For split rules, if $\boldsymbol{u} \to \{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ where $\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{u}_1 + \boldsymbol{u}_2$ for some $\boldsymbol{b}$ in $\boldsymbol{B}_+$ and $S' \supseteq_{\mathbb{H}} \{\boldsymbol{u}_1, \boldsymbol{u}_2\}$, then by definition of the Hoare quasi-order there exists $S'_{\boldsymbol{u}'} \stackrel{\text{def}}{=} \{\boldsymbol{u}'_1, \boldsymbol{u}'_2\} \subseteq S'$ with $\boldsymbol{u}_1 \le \boldsymbol{u}'_1$ and $\boldsymbol{u}_2 \le \boldsymbol{u}'_2$ and $\boldsymbol{u}' \stackrel{\text{def}}{=} \boldsymbol{u}'_1 + \boldsymbol{u}'_2 - \boldsymbol{b}$ fits: $\boldsymbol{u}' \to S'_{\boldsymbol{u}'}$ and $\boldsymbol{u}' \ge \boldsymbol{u}$.

In particular, BVAS, i.e. when $\boldsymbol{B}_\wedge$ is empty, are bottom-up WSTS. We also remark that the bottom-up coverability problem in meet ABVAS is equivalent to the reachability problem for *gainy* ABVAS [16]

*6.2. Backward bottom-up coverability*

Let us consider the instantiation of the algorithm of Section 3.1 to the WSTS $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$. Our main purpose here is to provide formulations in terms of the underlying bottom-up WSTS $(X, \rightarrow, \leq)$ of the main ingredients of the backward coverability algorithm. Its effectiveness requires

1. effective ideal representations for $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$, and
2. effective universal predecessors.

We will conclude this subsection by instantiating the proper transition sequences of Claim 4.2 in the case of the WSTS $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$.

*Effective Ideal Representations* Regarding the first point, we rely on the following fact.

**Fact 6.5** *([22,39,23]).* If $(X, \leq)$ is an $\omega^2$-wqo with effective ideal representations, then $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$ is a wqo with effective ideal representations.

Indeed, the ideals of $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$ are all $\downarrow_{\mathbb{H}} D \stackrel{\text{def}}{=} \{D' \in \mathbb{D}(X) \mid D' \subseteq D\}$ for downwards-closed $D \in \mathbb{D}(X)$, where $\mathbb{D}(X)$ denotes the downwards-closed subsets of $X$ and $\downarrow_{\mathbb{H}}$ the downward closure with respect to $\subseteq_{\mathbb{H}}$. Thus the ideals of $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$ can be represented as finite sets of ideals from $X$, and the algorithms for (ContId), (IntId), and (CompUp) are straightforward to adapt. In order to distinguish between the downwards-closed subsets of $\mathbb{P}(X)$ and those of $X$, we use calligraphic $\mathcal{D}$ to denote the former.

*Effective Universal Predecessors* Regarding the second point, since $\vdash$ is a function, we can simplify the expression of $\mathrm{Pre}_\forall(\mathcal{D})$ in (3) for a downwards-closed subset $\mathcal{D} \subseteq \mathbb{P}(X)$ to

$$\mathrm{Pre}_\forall(\mathcal{D}) = \{S \in \mathbb{P}(X) \mid S \vdash \hat{S} \in \mathcal{D}\}\ . \tag{30}$$

Note that $S \in \mathrm{Pre}_\forall(\mathcal{D})$, i.e. $S \vdash \hat{S} \in \mathcal{D}$, entails $S \in \mathcal{D}$ since $S \subseteq \hat{S}$ and $\mathcal{D}$ is downwards-closed, hence Equation (2) can be simplified using

$$\mathrm{Pre}_\forall(\mathcal{D}) = \mathcal{D} \cap \mathrm{Pre}_\forall(\mathcal{D})\ . \tag{31}$$

As $\mathcal{D}$ is downwards-closed, it is a finite union of ideals, hence $\hat{S} \in \mathcal{D}$ if and only if there exists a downwards-closed $D$ from the ideal decomposition of $\mathcal{D}$ such that $\hat{S} \in \downarrow_{\mathbb{H}} D$. Equivalently, this occurs exactly when $\hat{S} \subseteq D$, i.e. in terms of the underlying branching transition relation, exactly when $S \in \mathrm{Pre}_\forall(D)$ defined by

$$\mathrm{Pre}_\forall(D) \stackrel{\text{def}}{=} \{S \subseteq D \mid \forall x\,.\,(\exists S_x \subseteq S \wedge x \rightarrow S_x \implies x \in D)\}\ . \tag{32}$$

The set $\mathrm{Pre}_\forall(D)$ is downwards-closed for $\subseteq_{\mathbb{H}}$. If $\mathcal{D}$ decomposes as $\downarrow_{\mathbb{H}} D_1 \cup \cdots \cup \downarrow_{\mathbb{H}} D_n$, then

$$\mathrm{Pre}_\forall(\mathcal{D}) = \bigcup_{1 \leq i \leq n} \mathrm{Pre}_\forall(D_i)\ . \tag{33}$$

Hence (Pre) can be restated as requiring an algorithm computing an ideal decomposition for $\mathrm{Pre}_\forall(D)$ as defined in (32) for any downwards-closed $D \subseteq X$.

*Proper Transition Sequences* Recall the definition of existential successors from Equation (12). In the case of an ideal $\downarrow_{\mathbb{H}} D$ of $(\mathbb{P}(X), \subseteq_{\mathbb{H}})$, this yields

$$\mathrm{Post}_\exists(\downarrow_{\mathbb{H}} D) = \{\hat{S} \in \mathbb{P}(X) \mid \exists S \in \downarrow_{\mathbb{H}} D\,.\, S \vdash \hat{S}\}\ . \tag{34}$$

Also recall that we are actually interested in Claim 4.2 in its downward closure $\downarrow_{\mathbb{H}} \mathrm{Post}_\exists(\downarrow_{\mathbb{H}} D)$. It turns out that the latter has a single ideal: let $D \vdash \hat{D}$, then

$$\downarrow_{\mathbb{H}} \mathrm{Post}_\exists(\downarrow_{\mathbb{H}} D) = \downarrow_{\mathbb{H}} \hat{D}\ . \tag{35}$$

Indeed, $\hat{D}$ is certainly among the $\hat{S}$ in $\mathrm{Post}_\exists(\downarrow_{\mathbb{H}} D)$, and conversely, if $\hat{S} \in \mathrm{Post}_\exists(\downarrow_{\mathbb{H}} D)$ for some $S \in \downarrow_{\mathbb{H}} D$ with $S \vdash \hat{S}$, then by compatibility $\hat{S} \subseteq_{\mathbb{H}} \hat{D}$.

**Remark 6.6** *(Transfer of length function theorems).* At this stage, we can already observe that, in the computation $\mathcal{D}_0 \supsetneq \mathcal{D}_1 \supsetneq \cdots \supsetneq \mathcal{D}_\ell$ of the backward coverability algorithm in $(\mathbb{P}(X), \vdash, \subseteq_{\mathbb{H}})$, Equation (35) and Claim 4.2 show that there is a sequence of proper ideals $\downarrow_{\mathbb{H}} D_0, \downarrow_{\mathbb{H}} D_1, \ldots, \downarrow_{\mathbb{H}} D_{\ell-1}$ with $D_{k+1} \vdash \hat{D}_{k+1} \subseteq D_k$ for all $k$. Because $D_{k+1} \subseteq \hat{D}_{k+1}$, we have the inclusion $D_{k+1} \subseteq D_k$, and it is strict since $D_k$ is proper. Therefore the sequence is actually a descending chain $D_0 \supsetneq D_1 \supsetneq \cdots \supsetneq D_{\ell-1}$ of proper ideals, and it suffices to bound the length of that chain to obtain a bound on $\ell$.

Thus, in the case of bottom-up coverability, length function theorems like Theorem 3.10 on the length of descending chains of downwards-closed subsets of $X$ can be applied to yield a bound on the length of the descending chain $\mathcal{D}_0 \supsetneq \mathcal{D}_1 \supsetneq \cdots \supsetneq \mathcal{D}_\ell$ of downwards-closed subsets of $\mathbb{P}(X)$.

### 6.3. Complexity for meet ABVAS

As in the previous complexity analyses for coverability in VAS and reset VAS and top-down coverability in ABVAS, in order to apply length function theorems, we should show that universal predecessors can be computed in meet ABVAS and extract a control function. Together with Remark 6.6 and Theorem 3.10, this will be enough to get an Ackermann upper bound for bottom-up coverability in meet ABVAS. Furthermore, we shall see that in BVAS, the descending chain exhibited in Remark 6.6 is $\omega$-monotone, hence Theorem 4.4 can be applied to prove a 2ExpTime upper bound.

*Effectiveness and Control* Let the size of a downwards-closed subset $\mathcal{D}$ of $\mathbb{P}(\mathbb{N}^d)$ whose ideal decomposition is $\downarrow_{\mathbb{H}} D_1 \cup \cdots \cup \downarrow_{\mathbb{H}} D_n$ be the maximum of the sizes of $D_1, \ldots, D_n$.

Using a similar but lengthy case analysis as in the cases of VAS and reset VAS in Sections 3.2 and 3.3, we extract a control function from (32). This also entails the effectiveness of universal predecessors since there are only finitely many $\mathcal{D}$ of a given size.

**Claim 6.7** *(Control for meet ABVAS). 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 $n \stackrel{\text{def}}{=} \|\boldsymbol{t}\|$.*

*$\omega$-Monotonicity* The distinction between bottom-up ABVAS and BVAS appears here. As observed in Remark 6.6, inside the descending chain of downwards-closed subsets of $\mathbb{P}(\mathbb{N}^d)$ computed by the backward bottom-up 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 $\omega$-monotone, enabling us to infer that its length is then at most doubly exponential.

**Claim 6.8** *(Bottom-up BVAS descending chains are $\omega$-monotone). The descending chains $D_0 \supsetneq \cdots \supsetneq D_{\ell-1}$ of proper ideals extracted in Remark 6.6 from the computation of the backward bottom-up coverability algorithm for BVAS are $\omega$-monotone.*

**Proof.** Consider a $d$-dimensional BVAS $(\boldsymbol{A}, \emptyset, \boldsymbol{B}_+)$. First note that, if $D$ is a downwards-closed subset of $\mathbb{N}^d$ with an ideal decomposition $\downarrow \boldsymbol{u}_1 \cup \cdots \cup \downarrow \boldsymbol{u}_n$ and $D \vdash \hat{D}$, then $\downarrow \hat{D}$ can be written as

$$\downarrow \hat{D} = \left( \bigcup_{\boldsymbol{a} \in \boldsymbol{A}} \bigcup_{1 \le j \le n} \downarrow (\boldsymbol{u}_j - \boldsymbol{a}) \right) \cup \left( \bigcup_{\boldsymbol{b} \in \boldsymbol{B}_+} \bigcup_{1 \le i, j \le n} \downarrow (\boldsymbol{u}_i + \boldsymbol{u}_j - \boldsymbol{b}) \right). \tag{36}$$

Suppose now $0 \le k < \ell - 2$ and $\boldsymbol{v}_{k+1}$ is a proper vector in the decomposition of $D_{k+1}$, i.e. $\downarrow \boldsymbol{v}_{k+1} \not\subseteq D_{k+2}$. Let $D \stackrel{\text{def}}{=} D_{k+2} \cup \downarrow \boldsymbol{v}_k$ and $\hat{D}$ be such that $D \vdash \hat{D}$. On the one hand, $D \notin \mathcal{D}_{k+2}$, which entails

$$(D_{k+2} \cup \downarrow \boldsymbol{v}_k) \vdash \hat{D} \not\subseteq D_{k+1}\,, \tag{37}$$

since otherwise we would have $D \in \mathrm{Pre}_\forall(\mathcal{D}_{k+1}) = \mathcal{D}_{k+2}$ according to (30) and (31). On the other hand,

$$D_{k+2} \vdash \hat{D}_{k+2} \subseteq D_{k+1} \tag{38}$$

in the descending chain from Remark 6.6. In order to reconcile (37) and (38), there must exist $\boldsymbol{w}$ in the decomposition of $\downarrow \hat{D}$ such that $\downarrow \boldsymbol{w} \not\subseteq D_{k+1}$, and where $\boldsymbol{w}$ was 'produced' using $\boldsymbol{v}_k$ in (36):

- 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, $\omega(\boldsymbol{v}_{k+1}) \subseteq \omega(\boldsymbol{w})$.[2]

2 In a meet ABVAS, there is an additional possibility: $\boldsymbol{w} = \min\{\boldsymbol{v}_k - \boldsymbol{b}_1, \boldsymbol{v}' - \boldsymbol{b}_2\}$ for some $\boldsymbol{v}'$ in the decomposition of $D_{k+2}$ and $(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \boldsymbol{B}_\wedge$ or $(\boldsymbol{b}_2, \boldsymbol{b}_1) \in \boldsymbol{B}_\wedge$. In such a case, due to the 'min' operation, it is possible that $\omega(\boldsymbol{v}_{k+1}) \not\subseteq \omega(\boldsymbol{w})$.

Finally, $\downarrow \boldsymbol{w} \subseteq \downarrow \hat{D} \subseteq \downarrow \hat{D}_{k+1} \subseteq D_k$. Hence by ideal irreducibility there exists $\boldsymbol{v}_k$ in the decomposition of $D_k$ such that $\boldsymbol{w} \sqsubseteq \boldsymbol{v}_k$. Since $\downarrow \boldsymbol{w} \not\subseteq D_{k+1}$, $\boldsymbol{v}_k$ is proper, and furthermore $\omega(\boldsymbol{v}_{k+1}) \subseteq \omega(\boldsymbol{w}) \subseteq \omega(\boldsymbol{v}_k)$. $\square$

*Upper Bounds* Finally, by applying the length function theorem for descending chains (Theorem 3.10), or for $\omega$-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 $g^k(\|\boldsymbol{t}\|)$, so it suffices to bound the number of iterations of the main loop.

**Corollary 6.9.** *For any $d$-dimensional meet ABVAS $(\boldsymbol{A}, \boldsymbol{B}_\wedge, \boldsymbol{B}_+)$ and target vector $\boldsymbol{t}$, the backward bottom-up coverability algorithm terminates after at most* $\mathrm{ackermann}(p(d, L, \|\boldsymbol{t}\|))$ *steps, where $p$ is a primitive-recursive function and $L \stackrel{\text{def}}{=} \max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}_\wedge\|, \|\boldsymbol{B}_+\|\}$.*
*For BVAS, i.e. when $\boldsymbol{B}_\wedge$ is empty, the algorithm terminates after at most $((L+1)(\|\boldsymbol{t}\|+2))^{(d+1)!} + 1$ steps.*

These upper bounds are tight: Ackermann-hardness was shown originally by Urquhart [26] for reachability in a variant of gainy AVAS, and also holds for bottom-up coverability for meet AVAS [16]; 2ExpTime-hardness was first shown by Demri et al. [13] 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 [13–17]. 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 [13,15,16] in branching extensions of VAS.

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 [24] that the techniques presented in this paper yield new tight upper bounds for coverability in $\nu$-Petri nets [see e.g. 40], improving on the previously known HyperAckermann upper bound.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## 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. 28, Section 2.2.2]. Instead, we prove the bounds directly on descending chains, thanks to Theorem 3.10:

**Theorem 3.10** *(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 $h(q) \stackrel{\text{def}}{=} d \cdot g(q)$.*

The main tool to this end is the following statement, which combines Corollary 2.25 and Theorem 2.34 from [28]:

**Theorem A.1** *(Length function theorem for bad sequences). Let $n > 0$. Any $(g, n)$-controlled bad sequence over a polynomial combinatorial 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 $h(x) \stackrel{\text{def}}{=} d \cdot g(x)$.*

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 combinatorial 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 [28], but we shall give a detailed account for the reader's sake.

*Polynomial Combinatorial WQOs* Let us denote by $\Gamma_0$ the empty wqo and by $\Gamma_1$ the singleton set $\{\bullet\}$ well-quasi-ordered with equality. A *polynomial* wqo is one that can be constructed from $\Gamma_0$, $\Gamma_1$, and $\mathbb{N}$ through Cartesian products '$\times$' and disjoint unions '$+$', using respectively the product and sum orderings [28, Section 2.1.2].

A *combinatorial* quasi-order associates a *size* 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 $n$ (this is called a normed wqo in [28]). For polynomial wqos, we use the zero size on $\Gamma_1$ and the infinity norm on Cartesian products.

*Reflections* A *shrinking order reflection* (aka a normed reflection) between two combinatorial 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 combinatorial wqo and there is a shrinking order reflection from $(X, \leq_X, |.|_X)$ to it, then $(X, \leq_X, |.|_X)$ is a combinatorial wqo with $(g, n)$-controlled bad sequences of at most the same length [28, Proposition 2.16].

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

$$((\mathbb{N} + \Gamma_1)^d, \sqsubseteq, |.|) \tag{A.1}$$

by [28, Rem. 2.17]. This will be the polynomial combinatorial wqo on which we will apply Theorem A.1.

*Maximal Order Types* It remains to compute the maximal order type of $((\mathbb{N} + \Gamma_1)^d, \sqsubseteq, |.|)$. This can be done algebraically [28, Section 2.4.1] using natural sums '$\oplus$' and natural products '$\otimes$' of maximal order types:

$$o(\mathbb{N} + \Gamma_1) = \omega \oplus 1 = \omega + 1\,, \tag{A.2}$$

$$o((\mathbb{N} + \Gamma_1)^d) = \underbrace{(\omega + 1) \otimes \cdots \otimes (\omega + 1)}_{d \text{ times}} = \sum_{d \geq i \geq 0} \omega^i \binom{d}{i}\,. \tag{A.3}$$

*Cichoń Functions* Let us recall that, given a monotone inflationary $h\colon \mathbb{N} \to \mathbb{N}$, and an ordinal $\alpha$, the $\alpha$th *Cichoń function* $h_\alpha$ is defined by induction on $\alpha$ by

$$h_0(x) \stackrel{\text{def}}{=} 0\,, \qquad h_\alpha(x) \stackrel{\text{def}}{=} 1 + h_{P_x(\alpha)}(h(x))\,, \tag{A.4}$$

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

$$P_x(\alpha + 1) \stackrel{\text{def}}{=} \alpha\,, \qquad P_x(\gamma + \omega^\beta) \stackrel{\text{def}}{=} \gamma + \omega^{P_x(\beta)} \cdot x + P_x(\omega^{P_x(\beta)})\,. \tag{A.5}$$

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 $h_\alpha$ is monotone.

**Proof of Theorem 3.10.** By extracting proper ideals and applying Theorem A.1, we obtain an upper bound of

$$\ell = h_{o((\mathbb{N}+\Gamma_1)^d)}(n \cdot d) \tag{A.6}$$

on the length of $(g, n)$-controlled bad sequences over $\mathbb{N}_\omega^d$, 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 (A.3) is quite a mouthful, we are going to over-approximate this bound with a more readable one.

Recall that any ordinal $\alpha$ below $\omega^{d+1}$ 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 [28, Eq. 2.70] for ordinals below $\omega^{d+1}$ by:

$$\omega^d \cdot a_d + \cdots + \omega^0 \cdot a_0 \sqsubseteq \omega^d \cdot b_d + \cdots + \omega^0 \cdot b_0 \text{ if } \forall 1 \leq i \leq d\,.\, a_i \leq b_i\,. \tag{A.7}$$

By [28, Exercise 2.11], $\alpha \sqsubseteq \beta$ ensures $h_\alpha(n) \leq h_\beta(n)$ for all $n$.

Observe now that (A.3) is such that, for all $n > 0$,

$$o((\mathbb{N} + \Gamma_1)^d) \sqsubseteq \sum_{d \geq i \geq 0} \omega^i \cdot nd! = P_{nd!}(\omega^{d+1})\,. \tag{A.8}$$

Hence the result stated in Theorem 3.10, by (A.6) and

$$\begin{aligned} h_{o((\mathbb{N}+\Gamma_1)^d)}(nd) &\leq h_{o((\mathbb{N}+\Gamma_1)^d)}(h(nd!)) && \text{since } nd \leq h(nd!) \\ &\leq h_{P_{nd!}(\omega^{d+1})}(h(nd!)) && \text{by (A.8)} \\ &= h_{\omega^{d+1}}(nd!) - 1\,. && \square \end{aligned}$$

## References


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

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

[3] D. Figueira, S. Figueira, S. Schmitz, Ph. Schnoebelen, Ackermannian and primitive-recursive bounds with Dickson's lemma, in: LICS 2011, IEEE Computer Society, 2011, pp. 269–278, https://doi.org/10.1109/LICS.2011.39.

[4] S. Schmitz, Ph. Schnoebelen, Multiply-recursive upper bounds with Higman's lemma, in: ICALP 2011, in: Lecture Notes in Computer Science, vol. 6756, Springer, 2011, pp. 441–452, https://doi.org/10.1007/978-3-642-22012-8_35.

[5] F. Rosa-Velardo, Ordinal recursive complexity of unordered data nets, Inf. Comput. 254 (2017) 41–58, https://doi.org/10.1016/j.ic.2017.02.002.

[6] Ph. Schnoebelen, Revisiting Ackermann-Hardness for lossy counter machines and reset Petri nets, in: MFCS 2010, in: Lecture Notes in Computer Science, vol. 6281, Springer, 2010, pp. 616–628, https://doi.org/10.1007/978-3-642-15155-2_54.

[7] S. Haddad, S. Schmitz, Ph. Schnoebelen, The ordinal recursive complexity of timed-arc Petri nets, data nets, and other enriched nets, in: LICS 2012, IEEE Press, 2012, pp. 355–364, https://doi.org/10.1109/LICS.2012.46.

[8] P. Karandikar, S. Schmitz, The parametric ordinal-recursive complexity of post embedding problems, in: FoSSaCS 2013, in: Lecture Notes in Computer Science, vol. 7794, Springer, 2013, pp. 273–288, https://doi.org/10.1007/978-3-642-37075-5_18.

[9] C. Haase, S. Schmitz, Ph. Schnoebelen, The power of priority channel systems, Log. Methods Comput. Sci. 10 (4) (2014) 4, https://doi.org/10.2168/LMCS-10(4:4)2014.

[10] N. Decker, D. Thoma, On freeze LTL with ordered attributes, in: FoSSaCS 2016, in: Lecture Notes in Computer Science, vol. 9634, Springer, 2016, pp. 269–284, https://doi.org/10.1007/978-3-662-49630-5_16.

[11] R. Lipton, The reachability problem requires exponential space, Tech. Rep. 62, Yale University, 1976.

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

[13] S. Demri, M. Jurdziński, O. Lachish, R. Lazić, The covering and boundedness problems for branching vector addition systems, J. Comput. Syst. Sci. 79 (1) (2013) 23–38, https://doi.org/10.1016/j.jcss.2012.04.002.

[14] R. Bonnet, A. Finkel, M. Praveen, Extending the Rackoff technique to affine nets, in: FSTTCS 2012, Leibniz International Proceedings in Informatics, vol. 18, LZI, 2012, pp. 301–312, https://doi.org/10.4230/LIPIcs.FSTTCS.2012.301.

[15] J.-B. Courtois, S. Schmitz, Alternating vector addition systems with states, in: MFCS 2014, in: Lecture Notes in Computer Science, vol. 8634, Springer, 2014, pp. 220–231, https://doi.org/10.1007/978-3-662-44522-8_19.

[16] R. Lazić, S. Schmitz, Non-elementary complexities for branching VASS, MELL, and extensions, ACM Trans. Comput. Log. 16 (3) (2015) 20, https://doi.org/10.1145/2733375.

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

[18] R. Lazić, T. Newcomb, J. Ouaknine, A.W. Roscoe, J. Worrell, Nets with tokens which carry data, Fundam. Inform. 88 (3) (2008) 251–274.

[19] L. Bozzelli, P. Ganty, Complexity analysis of the backward coverability algorithm for VASS, in: RP 2011, in: Lecture Notes in Computer Science, vol. 6945, Springer, 2011, pp. 96–109, https://doi.org/10.1007/978-3-642-24288-5_10.

[20] R. Majumdar, Z. Wang, Expand, enlarge, and check for branching vector addition systems, in: Concur 2013, in: Lecture Notes in Computer Science, vol. 8052, Springer, 2013, pp. 152–166, https://doi.org/10.1007/978-3-642-40184-8_12.

[21] 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, in: Coll. Math. Soc. János Bolyai, vol. 1, North-Holland, 1975, pp. 189–198.

[22] A. Finkel, J. Goubault-Larrecq, Forward analysis for WSTS. Part I: completions, in: STACS 2009, Leibniz International Proceedings in Informatics, vol. 3, LZI, 2009, pp. 433–444, https://doi.org/10.4230/LIPIcs.STACS.2009.1844.

[23] J. Goubault-Larrecq, S. Halfon, P. Karandikar, K. Narayan Kumar, Ph. Schnoebelen, The ideal approach to computing closed subsets in well-quasi-orderings, Preprint, http://arxiv.org/abs/1904.10703, 2019.

[24] R. Lazić, S. Schmitz, The complexity of coverability in $\nu$-Petri nets, in: LICS 2016, ACM Press, 2016, pp. 467–476, https://doi.org/10.1145/2933575.2933593.

[25] M. Benedikt, T. Duff, A. Sharad, J. Worrell, Polynomial automata: zeroness and applications, in: LICS 2017, IEEE Computer Society, 2017, https://doi.org/10.1109/LICS.2017.8005101.

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

[27] K.N. Verma, J. Goubault-Larrecq, Karp-Miller trees for a branching extension of VASS, Discrete Math. Theor. Comput. Sci. 7 (1) (2005) 217–230, http://eudml.org/doc/129486.

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

[29] J. Leroux, P. Schnoebelen, On functions weakly computable by Petri nets and vector addition systems, in: RP 2014, in: Lecture Notes in Computer Science, vol. 8762, Springer, 2014, pp. 190–202, https://doi.org/10.1007/978-3-319-11439-2_15.

[30] A. Arnold, M. Latteux, Récursivité et cônes rationnels fermés par intersection, Calcolo 15 (4) (1978) 381–394, https://doi.org/10.1007/BF02576519.

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

[32] S. Schmitz, Complexity hierarchies beyond elementary, ACM Trans. Comput. Theory 8 (1) (2016) 1–36, https://doi.org/10.1145/2858784.

[33] M. Blondin, A. Finkel, P. McKenzie, Handling infinitely branching WSTS, in: ICALP 2014, in: Lecture Notes in Computer Science, vol. 8573, 2014, pp. 13–25, https://doi.org/10.1007/978-3-662-43951-7_2.

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

[35] A. Marcone, Fine analysis of the quasi-orderings on the power set, Order 18 (4) (2001) 339–347, https://doi.org/10.1023/A:1013952225669.

[36] Ph. de Groote, B. Guillaume, S. Salvati, Vector addition tree automata, in: LICS 2004, IEEE Computer Society, 2004, pp. 64–73, https://doi.org/10.1109/LICS.2004.51.

[37] S. Schmitz, On the computational complexity of dominance links in grammatical formalisms, in: ACL 2010, ACL Press, 2010, pp. 514–524.

[38] P. Lincoln, J. Mitchell, A. Scedrov, N. Shankar, Decision problems for propositional linear logic, Ann. Pure Appl. Log. 56 (1–3) (1992) 239–311, https://doi.org/10.1016/0168-0072(92)90075-B.

[39] A. Finkel, J. Goubault-Larrecq, Forward analysis for WSTS. Part II: complete WSTS, Log. Methods Comput. Sci. 8 (3) (2012) 28, https://doi.org/10.2168/LMCS-8(3:28)2012.

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