Forward Analysis for WSTS, Part III: Karp-Miller Trees[*]

Michael Blondin[†1], Alain Finkel[2], and Jean Goubault-Larrecq[2]

1 **Technische Universität München, Germany**
`blondin@in.tum.de`
2 **LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France**
`finkel@lsv.fr`
3 **LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France**
`goubault@lsv.fr`

Abstract

This paper is a sequel of "Forward Analysis for WSTS, Part I: Completions" [STACS 2009, LZI Intl. Proc. in Informatics 3, 433–444] and "Forward Analysis for WSTS, Part II: Complete WSTS" [Logical Methods in Computer Science 8(3), 2012]. In these two papers, we provided a framework to conduct forward reachability analyses of WSTS, using finite representations of downwards-closed sets. We further develop this framework to obtain a generic Karp-Miller algorithm for the new class of very-WSTS. This allows us to show that coverability sets of very-WSTS can be computed as their finite ideal decompositions. Under natural assumptions on positive sequences, we also show that LTL model checking for very-WSTS is decidable. The termination of our procedure rests on a new notion of acceleration levels, which we study. We characterize those domains that allow for only finitely many accelerations, based on ordinal ranks.

1998 ACM Subject Classification F.1.1. Models of Computation, F.3.1 Specifying and Verifying and Reasoning about Programs

Keywords and phrases WSTS, model checking, coverability, Karp-Miller algorithm, ideals

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2017.16

1 Introduction

Context. A well-structured transition system (WSTS) is an infinite well-quasi-ordered set of states equipped with transition relations satisfying one of various possible monotonicity properties. WSTS were introduced in [16] for the purpose of capturing properties common to a wide range of formal models used in verification. Since their inception, much of the work on WSTS has been dedicated to identifying generic classes of WSTS for which verification problems are decidable. Such problems include termination, boundedness [16, 17, 21] and coverability [1, 2, 6, 7]. In general, verifying safety and liveness properties corresponds respectively to deciding the coverability and the repeated control-state reachability problems. Coverability can be decided for WSTS by two different algorithms: the backward algorithm [1, 2] and by combining two forward semi-procedures, one of which enumerates all downwards-closed invariants [25, 6, 7]. Repeated control-state reachability is undecidable for general WSTS, but decidable for Petri nets by use of the Karp-Miller coverability tree [29] and the detection of positive sequences. That technique fails on well-structured extensions of Petri

[*] A full version of the paper is available at `https://arxiv.org/abs/1710.07258`.
[†] M. Blondin was supported by the Fonds de recherche du Québec – Nature et technologies (FRQNT).

37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017).
Editors: Satya Lokam and R. Ramanujam; Article No. 16; pp. 16:1–16:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

nets: generating the Karp-Miller tree does not always terminate on ν-Petri nets [36], on reset Petri nets [11], on transfer Petri nets, on broadcast protocols, and on the depth-bounded π-calculus [28, 35, 41] which can simulate reset Petri nets. This is perhaps why little research has been conducted on coverability tree algorithms and model checking of liveness properties for general WSTS. Nonetheless, some recent Petri nets extensions, e.g. ω-Petri nets [24] and unordered data Petri nets [27], benefit from algorithms in the style of Karp and Miller. Hence, there is hope of finding a general framework of WSTS with Karp-Miller-like algorithms.

The Karp-Miller coverability procedure. In 1967, Karp and Miller [29] proposed what is now known as the Karp-Miller coverability tree algorithm, which computes a finite representation (the *clover*) of the downward closure (the *cover*) of the reachability set of a Petri net. In 1978, Valk extended the Karp-Miller algorithm to post-self-modifiying nets [38], a strict extension of Petri nets. In 1987, the second author proposed a generalization of the Karp-Miller algorithm that applies to a class of finitely branching WSTS with strong-strict monotonicity, and having a WSTS completion in which least upper bounds replace the original Petri nets ω-accelerations [16, 17]. In 2004, Finkel, McKenzie and Picaronny [20] applied the framework of [17] to the construction of Karp-Miller trees for strongly increasing ω-recursive nets, a class generalizing post-self-modifiying nets. In 2005, Verma and the third author [40] showed that the construction of Karp-Miller trees can be extended to branching vector addition systems with states. In 2009, the second and the third authors [19] proposed a non-terminating procedure that computes the clover of *any* complete WSTS; this procedure terminates exactly on so-called cover-flattable systems. Recently, this framework has been used for defining computable accelerations in non-terminating Karp-Miller algorithms for both the depth-bounded π-calculus [28] and for ν-Petri nets; terminating Karp-Miller trees are obtained for strict subclasses.

Model checking WSTS. In 1994, Esparza [14] showed that model checking the linear time μ-calculus is decidable for Petri nets by using both the Karp-Miller algorithm and a decidability result due to Valk and Jantzen [39] on infinite T-continual sequences in Petri nets. LTL is undecidable for Petri net extensions such as lossy channel systems [3] and lossy counter machines [37]. In 1998, Emerson and Namjoshi [12] studied the model checking of liveness properties for complete WSTS, but their procedure is not guaranteed to terminate. In 2004, Kouzmin, Shilov and Sokolov [30] gave a generic computability result for a fragment of the μ-calculus; in 2006 and 2013, Bertrand and Schnoebelen [4, 5] studied fixed points in well-structured regular model checking; both [30] and [5] are concerned with formulas with upwards-closed atomic propositions, and do not subsume LTL. In 2011, Chambart, Finkel and Schmitz [9, 10] showed that LTL is decidable for the recursive class of trace-bounded complete WSTS; a class which does not contain all Petri nets.

Our contributions.

- We define *very-well-structured transition systems (very-WSTS)*; a class defined in terms of WSTS completions, and which encompasses models such as Petri nets, ω-Petri nets, post-self-modifying nets and strongly increasing ω-recursive nets. We show that coverability sets of very-WSTS are computable as finite sets of ideals.
- The general clover algorithm of [19], based on the ideal completion studied in [18], does not necessarily terminate and uses an abstract acceleration enumeration. We give an algorithm, the Ideal Karp-Miller algorithm, which organizes accelerations within a tree. We show that this algorithm terminates under natural order-theoretic and effectiveness

conditions, which we make explicit. This allows us to unify various versions of Karp-Miller algorithms in particular particular classes of WSTS.

- We identify the crucial notion of *acceleration level* of an ideal, and relate it to ordinal ranks of sets of reachable states in the completion. We show, notably, that termination is equivalent to the rank being strictly smaller than ω^2. This classifies WSTS into those with high rank (the bad ones), among which those whose sets of states consist of words (e.g., lossy channel systems) or multisets; and those with low rank (the good ones), among which Petri nets and post-self-modifying nets.
- We show that the downward closure of the trace language of a very-WSTS is computable, again as a finite union of ideals. This shows that downward traces inclusion is decidable for very-WSTS.
- Finally, we prove the decidability of model checking liveness properties for very-WSTS under some effectiveness hypotheses.

Differences between very-WSTS and WSTS of [17]. The class of WSTS of [17, Def. 4.17] is reminiscent of very-WSTS. It requires WSTS to be finitely branching and strictly monotone, whereas our definition allows infinite branching and requires the *completion* to be strictly monotone. Moreover, [17, Thm. 4.18], which claims that its Karp-Miller procedure terminates, is incorrect since it does not terminate on transfer Petri nets and broadcast protocols [15], which are finitely branching and strictly monotone WSTS. Finally, some assumptions required to make the Karp-Miller procedure of [17] effective are missing.

Due to space constraints, some proofs are deferred to an extended version of this paper freely available online under the same title.

2 Preliminaries

We write $\subseteq$ for set inclusion and $\subset$ for strict set inclusion. A relation $\leq \subseteq X \times X$ over a set X is a *quasi-ordering* if it is reflexive and transitive, and a *partial ordering* if it is antisymmetric as well. It is *well-founded* if it has no infinite descending chain. A quasi-ordering $\leq$ is a *well-quasi-ordering* (resp. *well partial order*), *wqo* (resp. *wpo*) for short, if for every infinite sequence $x_0, x_1, \dots \in X$, there exist $i < j$ such that $x_i \leq x_j$. This is strictly stronger than being well-founded.

One example of well-quasi-ordering is the componentwise ordering of tuples over $\mathbb{N}$. More formally, $\mathbb{N}^d$ is well-quasi-ordered by $\leq$ where, for every $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{N}^d$, $\boldsymbol{x} \leq \boldsymbol{y}$ if and only if $\boldsymbol{x}(i) \leq \boldsymbol{y}(i)$ for every $i \in [d]$. We extend $\mathbb{N}$ to $\mathbb{N}_\omega \stackrel{\text{def}}{=} \mathbb{N} \cup \{\omega\}$ where $n \leq \omega$ for every $n \in \mathbb{N}_\omega$. $\mathbb{N}_\omega^d$ ordered componentwise is also well-quasi-ordered. Let Σ be a finite alphabet. We denote the set of finite words and infinite words over Σ respectively by Σ^* and Σ^ω. For every $u, v \in \Sigma^*$, we write $u \preceq v$ if u is a subword of v, i.e. u can be obtained from v by removing zero, one or multiple letters. Σ^* is well-quasi-ordered by $\preceq$.

Transition systems. A *(labeled and ordered) transition system* is a triple $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ such that X is a set, Σ is a finite alphabet, $\xrightarrow{a} \subseteq X \times X$ for every $a \in \Sigma$, and $\leq$ is a quasi-ordering on X. Elements of X are called the *states* of $\mathcal{S}$, each $\xrightarrow{a}$ is a *transition relation* of $\mathcal{S}$, and $\leq$ is the *ordering* of $\mathcal{S}$. A *class* $\mathcal{C}$ *of transition systems* is any set of transition systems. We extend transition relations to sequences over Σ, i.e. for every $x, y \in X$, $x \xrightarrow{\varepsilon} x$, and $x \xrightarrow{wa} y$ if there exists $x' \in X$ such that $x \xrightarrow{w} x' \xrightarrow{a} y$. We write $x \xrightarrow{*} y$ (resp. $x \xrightarrow{+} y$) if there exists $w \in \Sigma^*$ (resp. $w \in \Sigma^+$) such that $x \xrightarrow{w} y$. The finite and infinite *traces* of a transition system $\mathcal{S}$ from a state $x \in X$ are respectively defined as $\text{Traces}_{\mathcal{S}}(x) \stackrel{\text{def}}{=} \{w \in \Sigma^* : x \xrightarrow{w} y \text{ for some } y \in X\}$ and $\omega\text{-Traces}_{\mathcal{S}}(x) \stackrel{\text{def}}{=} \{w \in \Sigma^\omega : x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \cdots \text{ for some } x_1, x_2, \ldots \in X\}$.

We define the *immediate successors* and *immediate predecessors* of a state x under some sequence $w \in \Sigma^*$ as $\text{Post}_{\mathcal{S}}(x, w) \stackrel{\text{def}}{=} \{y \in X : x \xrightarrow{w} y\}$ and $\text{Pre}_{\mathcal{S}}(x, w) \stackrel{\text{def}}{=} \{y \in X : y \xrightarrow{w} x\}$. The *successors* and *predecessors* of $x \in X$ are $\text{Post}^*_{\mathcal{S}}(x) \stackrel{\text{def}}{=} \{y \in X : x \xrightarrow{*} y\}$ and $\text{Pre}^*_{\mathcal{S}}(x) \stackrel{\text{def}}{=} \{y \in X : y \xrightarrow{*} x\}$. These notations are naturally extended to sets, e.g. $\text{Post}_{\mathcal{S}}(A, w) \stackrel{\text{def}}{=} \{\text{Post}_{\mathcal{S}}(x, w) : x \in A\}$. We say that $\mathcal{S}$ is *deterministic* if $|\text{Post}_{\mathcal{S}}(x, a)| \leq 1$ for every $x \in X$ and $a \in \Sigma$. When $\mathcal{S}$ is deterministic, each $a \in \Sigma$ induces a partial function $t_a : X \to X$ such that $t_a(x) = y$ for each $x \in X$ such that $\text{Post}_{\mathcal{S}}(x, a) = \{y\}$. For readability, we simply write a for t_a, i.e. $a(x) = t_a(x)$. For every $w \in \Sigma^*$, we write $w(x)$ for $\text{Post}_{\mathcal{S}}(x, w)$ if $\text{Post}_{\mathcal{S}}(x, w) \neq \emptyset$.

Well-structured transition systems. A *(labeled and ordered) transition system* $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ is a *well-structured transition system (WSTS)* if $\leq$ is a well-quasi-ordering and $\mathcal{S}$ is *monotone*, i.e. for all $x, x', y \in X$ and $a \in \Sigma$ such that $x \xrightarrow{a} y$ and $x' \geq x$, there exists $y' \in X$ such that $x' \xrightarrow{*} y'$ and $y' \geq y$. Many other types of monotonicities were defined in the literature (see [21]), but, for our purposes, we only need to introduce strong monotonicities. We say that $\mathcal{S}$ has *strong monotonicity* if for all $x, x', y \in X$ and $a \in \Sigma$, $x \xrightarrow{a} y$ and $x' \geq x$ implies $x' \xrightarrow{a} y'$ for some $y' \geq y$. We say that $\mathcal{S}$ has *strong-strict monotonicity*[1] if it has strong monotonicity and for all $x, x', y \in X$ and $a \in \Sigma$, $x \xrightarrow{a} y$ and $x' > x$ implies $x' \xrightarrow{a} y'$ for some $y' > y$.

Verification problems. We say that a *target state* $y \in X$ is *coverable from* an *initial state* $x \in X$ if there exists $z \geq y$ such that $x \xrightarrow{*} z$ and $z \geq y$. The *coverability problem* asks whether a target state y is coverable from an initial state x. The *repeated coverability problem* asks whether a target state y is coverable infinitely often from an initial state x; i.e. whether there exist $z_0, z_1, \dots \in X$ such that $x \xrightarrow{*} z_0 \xrightarrow{+} z_1 \xrightarrow{+} \cdots$ and $z_i \geq y$ for every $i \in \mathbb{N}$.

3 An investigation of the Karp-Miller algorithm

In order to present our Karp-Miller algorithm for WSTS, we first highlight the key components of the Karp-Miller algorithm for vector addition systems. A *d-dimensional vector addition system (d-VAS)* is a WSTS $\mathcal{V} = (\mathbb{N}^d, \xrightarrow{T}, \leq)$ induced by a finite set $T \subseteq \mathbb{Z}^d$ and the rules:

$$\boldsymbol{x} \xrightarrow{\boldsymbol{t}} \boldsymbol{y} \stackrel{\text{def}}{\iff} \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{t}, \text{ for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{N}^d, \boldsymbol{t} \in T.$$

Vector addition systems are deterministic and have strong-strict monotonicity. Given a d-VAS and a vector $\boldsymbol{x}_{\text{init}} \in \mathbb{N}^d$, the Karp-Miller algorithm initializes a rooted tree whose root is labeled by $\boldsymbol{x}_{\text{init}}$. For every $\boldsymbol{t} \in T$ such that $\boldsymbol{x} + \boldsymbol{t} \geq \boldsymbol{0}$, a child labeled by $\boldsymbol{x} + \boldsymbol{t}$ is added to the root. This process is repeated successively to the new nodes. If a newly added node $c : \boldsymbol{x}$ has an ancestor $c' : \boldsymbol{x}'$ such $\boldsymbol{x} = \boldsymbol{x}'$, then it is not explored furthermore. If a newly added node $c : \boldsymbol{x}$ has an ancestor $c' : \boldsymbol{x}'$ such $\boldsymbol{x} > \boldsymbol{x}'$, then c is relabeled by the vector $\boldsymbol{y} \in \mathbb{N}^d_\omega$ such that $\boldsymbol{y}(i) \stackrel{\text{def}}{=} \boldsymbol{x}(i)$ if $\boldsymbol{x}(i) = \boldsymbol{x}'(i)$ and $\boldsymbol{y}(i) \stackrel{\text{def}}{=} \omega$ if $\boldsymbol{x}(i) > \boldsymbol{x}'(i)$. The latter operation is called an *acceleration* of c.

[1] Strong-strict monotonicity should not be confused with strong *and* strict monotonicities. Here strongness and strictness have to hold at the *same* time.

A vector $\boldsymbol{x}_{\text{tgt}}$ is coverable from $\boldsymbol{x}_{\text{init}}$ if and only if the resulting tree $\mathcal{T}$ contains a node $c : \boldsymbol{x}$ such that $\boldsymbol{x} \geq \boldsymbol{x}_{\text{tgt}}$. Similarly, $\boldsymbol{x}_{\text{tgt}}$ is repeatedly coverable from $\boldsymbol{x}_{\text{init}}$ if and only if $\mathcal{T}$ contains a node $c : \boldsymbol{x}$ that has an ancestor that was accelerated, and such that $\boldsymbol{x} \geq \boldsymbol{x}_{\text{tgt}}$.

3.1 Ideals and completions

One feature of the Karp-Miller algorithm is that it works over $\mathbb{N}_\omega^d$ instead of $\mathbb{N}^d$. Intuitively, vectors containing some ω correspond to "limit" elements. For a generic WSTS $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$, a similar extension of X is not obvious. Let us present one, called the *completion* of $\mathcal{S}$ in [19]. Instead of operating over X, the completion of $\mathcal{S}$ operates over the so-called *ideals* of X. In particular, the ideals of $\mathbb{N}^d$ are isomorphic to $\mathbb{N}_\omega^d$.

Let X be a set quasi-ordered by $\leq$. The *downward closure* of $D \subseteq X$ is defined as $\downarrow D \overset{\text{def}}{=} \{x \in X : x \leq y \text{ for some } y \in D\}$. A subset $D \subseteq X$ is *downwards-closed* if $D = \downarrow D$. An *ideal* is a downwards-closed subset $I \subseteq X$ that is additionally *directed*: I is non-empty and for all $x, y \in I$, there exists $z \in I$ such that $x \leq z$ and $y \leq z$ (equivalently, every finite subset of I has an upper bound in I). We denote the set of ideals of X by $\text{Idl}(X)$, i.e. $\text{Idl}(X) \overset{\text{def}}{=} \{D \subseteq X : D = \downarrow D \text{ and } D \text{ is directed}\}$.

It is known that $\text{Idl}(\mathbb{N}^d) = \{A_1 \times \cdots \times A_d : A_1, \ldots, A_d \in \{\downarrow n : n \in \mathbb{N}\} \cup \{\mathbb{N}\}\}$. Therefore, every ideal of $\mathbb{N}^d$ is naturally represented by some vector of $\mathbb{N}_\omega^d$, and vice versa. We write ω-rep(I) for this representation, for every $I \in \text{Idl}(\mathbb{N}^d)$. For example, the ideal $I = \mathbb{N} \times \downarrow 8 \times \downarrow 3 \times \mathbb{N}$ is represented by $\omega\text{-rep}(I) = (\omega, 8, 3, \omega)$.

Downwards-closed subsets can often be represented by finitely many ideals:

▶ **Theorem 1** ([13, 8, 33, 34, 23, 31]). *Let X be a well-quasi-ordered set. For every downwards-closed subset $D \subseteq X$, there exist $I_1, I_2, \ldots, I_n \in \text{Idl}(X)$ s.t. $D = I_1 \cup I_2 \cup \cdots \cup I_n$.*

This theorem gives rise to a canonical decomposition of downwards-closed sets. The *ideal decomposition* of a downwards-closed subset $D \subseteq X$ is the set of maximal ideals contained in D with respect to inclusion. We denote the ideal decomposition of D by $\text{IdealDecomp}(D) \overset{\text{def}}{=} \max_{\subseteq}\{I \in \text{Idl}(X) : I \subseteq D\}$. By Theorem 1, IdealDecomp(D) is finite, and $D = \bigcup_{I \in \text{IdealDecomp}(D)} I$. In [19, 6], the notion of ideal decomposition is used to define the completion of unlabeled WSTS. We slightly extend this notion to labeled WSTS:

▶ **Definition 2.** Let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a labeled WSTS. The *completion* of $\mathcal{S}$ is the labeled transition system $\widehat{\mathcal{S}} = (\text{Idl}(X), \overset{\Sigma}{\rightsquigarrow}, \subseteq)$ such that $I \overset{a}{\rightsquigarrow} J$ if, and only if,

$$J \in \text{IdealDecomp}(\downarrow \text{Post}_{\mathcal{S}}(I, a)).$$

The completion of a WSTS enjoys numerous properties. In particular, it has strong monotonicity, and it is finitely branching [6], i.e. $\text{Post}_{\widehat{\mathcal{S}}}(I, a)$ is finite for every $I \in \text{Idl}(X)$ and $a \in \Sigma$. Note that if $\mathcal{S}$ has strong-strict monotonicity, then this property is not necessarily preserved by $\widehat{\mathcal{S}}$ [6]. Moreover, the completion of a WSTS may not be a WSTS since Idl(X) is not always well-quasi-ordered by $\subseteq$. However, for the vast majority of models used in verification, Idl(X) is well-quasi-ordered, and hence completions remain well-structured. Indeed, Idl(X) is well-quasi-ordered if and only if X is a so-called ω^2-wqo, and all known wqos, except possibly graphs under minor embedding, are ω^2-wqo, as discussed in [19]. The traces of a WSTS are closely related to those of its completion:

▶ **Proposition 3** ([6]). *The following holds for every WSTS $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$:*

1. *For all $x, y \in X$ and $w \in \Sigma^*$, if $x \xrightarrow{w} y$, then for every ideal $I \supseteq \downarrow x$, there exists an ideal $J \supseteq \downarrow y$ such that $I \overset{w}{\rightsquigarrow} J$.*

2. *For all $I, J \in \mathrm{Idl}(X)$ and $w \in \Sigma^*$, if $I \overset{w}{\rightsquigarrow} J$, then for every $y \in J$, there exist $x \in I, y' \in X$ and $w' \in \Sigma^*$ such that $x \xrightarrow{w'} y'$ and $y' \geq y$. If $\mathcal{S}$ has strong monotonicity, then $w' = w$.*
3. *if $\mathcal{S}$ has strong monotonicity, then $\bigcup_{J \in Post_{\widehat{\mathcal{S}}}(I,w)} J = {\downarrow} Post_{\mathcal{S}}(I, w)$ for all $I \in \mathrm{Idl}(X)$ and $w \in \Sigma^*$.*
4. *if $\mathcal{S}$ has strong monotonicity, then $\mathrm{Traces}_{\mathcal{S}}(x) = \mathrm{Traces}_{\widehat{\mathcal{S}}}({\downarrow} x)$ and ω-$\mathrm{Traces}_{\mathcal{S}}(x) \subseteq \omega$-$\mathrm{Traces}_{\widehat{\mathcal{S}}}({\downarrow} x)$ for every $x \in X$.*

It is worth noting that if $\mathcal{S}$ is infinitely branching, then an infinite trace of $\widehat{\mathcal{S}}$ from ${\downarrow} x$ is not necessarily an infinite trace of $\mathcal{S}$ from x (e.g. see [6]). Whenever the completion of a WSTS $\mathcal{S}$ is deterministic, we will often write $w(I)$ for $\mathrm{Post}_{\widehat{\mathcal{S}}}(I, w)$ if the latter is nonempty and if there is no ambiguity with $\mathrm{Post}_{\mathcal{S}}(I, w)$.

3.2 Levels of ideals

The Karp-Miller algorithm terminates for the following reasons: $\mathbb{N}_\omega^d$ is well-quasi-ordered and ω's can only be added to vectors along a branch at most d times. Loosely speaking, the latter property means that $\mathrm{Idl}(\mathbb{N}^d)$ has $d+1$ "levels". Here, we generalize this notion. We say that an infinite sequence of ideals $I_0, I_1, \ldots \in \mathrm{Idl}(X)$ is an *acceleration candidate* if $I_0 \subset I_1 \subset \cdots$.

▶ **Definition 4.** For every $n \in \mathbb{N}$, the n^{th} *level* of $\mathrm{Idl}(X)$ is defined as

$$\mathrm{Acc}_n(X) = \begin{cases} \mathrm{Idl}(X) & \text{if } n = 0, \\ \left\{\bigcup_{i \in \mathbb{N}} I_i : I_0, I_1, \ldots \in \mathrm{Acc}_{n-1}(X) \text{ is an acceleration candidate}\right\} & \text{if } n > 0. \end{cases}$$

We observe that $\mathrm{Acc}_{n+1}(X) \subseteq \mathrm{Acc}_n(X)$ for every $n \in \mathbb{N}$. Moreover, as expected:

$$\mathrm{Acc}_n(\mathbb{N}^d) = \{I \in \mathrm{Idl}(\mathbb{N}^d) : \omega\textit{-rep}(I) \textit{ has at least } n \textit{ occurrences of } \omega\}.$$

We say that $\mathrm{Idl}(X)$ has *finitely many levels* if there exists $n \in \mathbb{N}$ such that $\mathrm{Acc}_n(X) = \emptyset$. For example, $\mathrm{Acc}_{d+1}(\mathbb{N}^d) = \emptyset$.

3.3 Accelerations

The last key aspect of the Karp-Miller algorithm is the possibility to accelerate nodes. In order to generalize this notion, let us briefly develop some intuition. Recall that a newly added node $c : \boldsymbol{x}$ is accelerated if it has an ancestor $c' : \boldsymbol{x}'$ such that $\boldsymbol{x} > \boldsymbol{x}'$. Consider the non-empty sequence w labeling the path from c' to c. Since d-VAS have strong-strict monotonicity, both over $\mathbb{N}^d$ and $\mathbb{N}_\omega^d$, $w^n(\boldsymbol{x})$ is defined for every $n \in \mathbb{N}$. For example, if $(5, 0, 1) \xrightarrow{w} (5, 1, 3)$ is encountered, $(5, 1, 3)$ is replaced by $(5, \omega, \omega)$. This represents the fact that for every $n \in \mathbb{N}$, there exists some reachable marking $\boldsymbol{y} \geq (5, n, n)$. Note that an acceleration increases the number of occurrences of ω. In our example, the ideal $I = {\downarrow} 5 \times {\downarrow} 1 \times {\downarrow} 3$, which is of level 0, is replaced by $I' = {\downarrow} 5 \times \mathbb{N} \times \mathbb{N}$, which is of level 2. Based on these observations, we extend the notion of acceleration to completions:

▶ **Definition 5.** Let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a WSTS such that $\widehat{\mathcal{S}}$ is deterministic and has strong-strict monotonicity, let $w \in \Sigma^+$ and let $I \in \mathrm{Idl}(X)$. The *acceleration* of I under w is defined as:

$$w^\infty(I) \overset{\text{def}}{=} \begin{cases} \bigcup_{k \in \mathbb{N}} w^k(I) & \text{if } I \subset w(I), \\ I & \text{otherwise.} \end{cases}$$

Note that for every ideal I, $w^\infty(I)$ is also an ideal. As for $\mathrm{Idl}(\mathbb{N}^d)$, any successor J of an ideal I belongs to the same level of I, and accelerating an ideal increases its level.

▶ **Proposition 6.** *Let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a WSTS such that $\mathcal{S}$ has strong monotonicity, and $\widehat{\mathcal{S}}$ is deterministic and has strong-strict monotonicity. For every $I \in \mathrm{Idl}(X)$ and $w \in \Sigma^+$,*

1. *if $Post_{\widehat{\mathcal{S}}}(I, w) \neq \emptyset$ and $I \in \mathrm{Acc}_n(X)$ for some $n \in \mathbb{N}$, then $w(I) \in \mathrm{Acc}_n(X)$;*
2. *if $I \subset w(I)$ and $I \in \mathrm{Acc}_n(X)$ for some $n \in \mathbb{N}$, then $w^\infty(I) \in \mathrm{Acc}_{n+1}(X)$.*

4 The Ideal Karp-Miller algorithm

We may now present our generalization of the Karp-Miller algorithm. To do so, we first define the class of WSTS that enjoys all of the properties introduced in the previous section:

▶ **Definition 7.** A *very-WSTS* is a labeled WSTS $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ such that:

- $\mathcal{S}$ has strong monotonicity,
- $\widehat{\mathcal{S}}$ is a deterministic WSTS with strong-strict monotonicity,
- $\mathrm{Idl}(X)$ has finitely many levels.

The class of very-WSTS includes vector addition systems, Petri nets, ω-Petri nets [24], post-self-modifying nets [38] and strongly increasing ω-recursive nets [20]. However, very-WSTS do not include transfer Petri nets, since $\widehat{S}$ does not have strict monotonicity, and unordered data Petri nets, since $\mathrm{Idl}(X)$ has infinitely many levels. Note that $\widehat{\mathcal{S}}$ may be deterministic (and finitely branching) even when $\mathcal{S}$ is not, and even when $\mathcal{S}$ is not finitely branching, as the example of ω-Petri nets shows.

We present the *Ideal Karp-Miller algorithm (IKM)* for this class in Algorithm 4.1. The algorithm starts from an ideal I_0, successively computes its successors in $\widehat{\mathcal{S}}$ and performs accelerations as in the classical Karp-Miller algorithm for VAS. Note that we do *not* allow for nested accelerations. For every node $c : \langle I, n \rangle$ of the tree built by the algorithm, we write ideal(c) for I, and num-accel(c) for n, which will be the number of accelerations made along the branch from the root to c (inclusively). Let us first show that the algorithm terminates.

Algorithm 4.1: Ideal Karp-Miller algorithm.

```
initialize a tree $\mathcal{T}$ with root $r\colon \langle I_0, 0 \rangle$
while $\mathcal{T}$ contains an unmarked node $c\colon \langle I, n \rangle$ do
    if $c$ has an ancestor $c'\colon \langle I', n' \rangle$ s.t. $I' = I$ then  mark $c$
    else
        if $c$ has an ancestor $c'\colon \langle I', n' \rangle$ s.t. $I' \subset I$
             and $n' = n$ /* no acceleration occurred between $c'$ and $c$ */
          then
            $w \leftarrow$ sequence of labels from $c'$ to $c$
            replace $c\colon \langle I, n \rangle$ by $c\colon \langle w^\infty(I), n + 1 \rangle$
        for $a \in \Sigma$ do
            if $a(I)$ is defined then
                add arc labeled by $a$ from $c$ to a new child $d\colon \langle a(I), n \rangle$
        mark $c$
return $\mathcal{T}$
```

▶ **Theorem 8.** *Algorithm 4.1 terminates for very-WSTS.*

Proof. We note the following invariants: (1) for every node $c : \langle I, n\rangle$ of $\mathcal{T}$, I is in $\mathrm{Acc}_n(X)$; (2) at line 2, i.e., each time control returns to the beginning of the loop, all unmarked nodes of $\mathcal{T}$ are leaves; (3) num-accel(c) is non-decreasing on each branch of $\mathcal{T}$, that is: for every branch $c_0 : \langle I_0, n_0\rangle$, $c_1 : \langle I_1, n_1\rangle$, …, $c_k : \langle I_k, n_k\rangle$ of $\mathcal{T}$, we have $n_1 \leq n_2 \leq \cdots \leq n_k$. (1) is by Proposition 6, (2) is an easy induction on the number of times through the loop, and (3) is also by induction, noticing that by (2) only n_k can increase when line 8 is executed.

The rest of the argument is as for the classical Karp-Miller algorithm. Suppose the algorithm does not terminate. Let $\mathcal{T}_n$ be the finite tree obtained after n iterations. The infinite sequence $\mathcal{T}_0, \mathcal{T}_1, \ldots$ defines a unique infinite tree $\mathcal{T}_\infty = \bigcup_{n\in\mathbb{N}} \mathcal{T}_n$. Since $\widehat{\mathcal{S}}$ is finitely branching, $\mathcal{T}_\infty$ is also finitely branching. Therefore, $\mathcal{T}_\infty$ contains an infinite path $c_0 : \langle I_0, n_0\rangle$, $c_1 : \langle I_1, n_1\rangle$, …, $c_k : \langle I_k, n_k\rangle$, …, by König's lemma. By (1), and since $\mathrm{Idl}(X)$ has finitely many levels, the numbers n_k assume only finitely many values. Let N be the largest of those values. Using (3), there is a $k_0 \in \mathbb{N}$ such that $n_k = N$ for every $k \geq k_0$. Since $\widehat{\mathcal{S}}$ is a WSTS, hence $\mathrm{Idl}(X)$ is wqo, we can find two indices i, j with $k_0 \leq i < j$ and such that $I_i \subseteq I_j$. If $I_i = I_j$, then line 3 of the algorithm would have stopped the exploration of the path. Hence $I_i \subset I_j$, but then line 8 would have replaced num-accel(c_j) $= N$ by $N + 1$, contradiction. ◀

4.1 Properties of the algorithm

Let $\mathcal{T}_I$ denote the tree induced by the set of nodes returned by Algorithm 4.1 on input $(\mathcal{S}, I)$. Let $D_I \overset{\text{def}}{=} \bigcup_{c\in\mathcal{T}_I} \mathrm{ideal}(c)$. We claim that $D_I = {\downarrow}\,\mathrm{Post}^*_{\mathcal{S}}(I)$. Instead of proving this claim directly, we take traces into consideration and prove a stronger statement. We define two word automata that will be useful for this purpose.

▶ **Definition 9.** The *stuttering automaton*^{2} is the finite word automaton $\mathcal{A}_I$ obtained by making all of the states of $\mathcal{T}_I$ accepting, by taking the root r as the initial state, and by taking the arcs of $\mathcal{T}_I$ as transitions, together with the following additional transitions:

- If a leaf c of $\mathcal{T}_I$ has an ancestor c' such that $\mathrm{ideal}(c) = \mathrm{ideal}(c')$, then a transition from c to c' labeled by ε is added to $\mathcal{A}_I$.

The *Karp-Miller automaton* is the automaton $\mathcal{K}_I$ obtained by extending $\mathcal{A}_I$ as follows:

- If a node c of $\mathcal{T}_I$ has been accelerated because of an ancestor c', then a transition from c to c' labeled by ε is added to $\mathcal{K}_I$.

Both $\mathcal{A}_I$ and $\mathcal{K}_I$ can be computed from $\mathcal{T}_I$. Moreover, they give precious information about the traces of $\mathcal{S}$. Let $\mathrm{L}(\mathcal{A}_I)$ and $\mathrm{L}(\mathcal{K}_I)$ denote the language over Σ accepted by $\mathcal{A}_I$ and $\mathcal{K}_I$. We will show the following theorem:

▶ **Theorem 10.** *For every very-WSTS $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ and $I \in \mathrm{Idl}(X)$,*

$$D_I = {\downarrow}\,\mathit{Post}^*_{\mathcal{S}}(I),\ \mathrm{Traces}_{\mathcal{S}}(I) \subseteq \mathrm{L}(\mathcal{A}_I) \textit{ and } \mathrm{L}(\mathcal{K}_I) \subseteq {\downarrow_{\preceq}}\,\mathrm{Traces}_{\mathcal{S}}(I).$$

In particular, for every $x \in X$, $D_{\downarrow x} = {\downarrow}\,\mathit{Post}^*_{\mathcal{S}}(x)$, ${\downarrow_{\preceq}}\,\mathrm{L}(\mathcal{K}_{\downarrow x}) = {\downarrow_{\preceq}}\,\mathrm{Traces}_{\mathcal{S}}(x)$, and ${\downarrow_{\preceq}}\,\mathrm{Traces}_{\mathcal{S}}(x)$ is a computable regular language.

The proof of Theorem 10 follows from the forthcoming Prop. 11 describing the relations between traces of $\mathcal{A}_I$ and $\mathcal{K}_I$ with traces of $\mathcal{S}$ and $\widehat{\mathcal{S}}$. We write $c \overset{w}{\dashrightarrow}_{\mathcal{T}} c'$, $c \overset{w}{\dashrightarrow}_{\mathcal{A}} c'$ and $c \overset{w}{\dashrightarrow}_{\mathcal{K}} c'$ whenever node c' can be reached by reading w from c in $\mathcal{T}_I$, $\mathcal{A}_I$ and $\mathcal{K}_I$ respectively.

[2] We use the term *stuttering* as paths of the automaton correspond to stuttering paths of [24].

▶ **Proposition 11.** *Let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a very-WSTS and let $I_0 \in \mathrm{Idl}(X)$.*

1. *For every $y, z \in X$, $w \in \Sigma^*$ and $c \in \mathcal{A}_{I_0}$, if $y \xrightarrow{w} z$ and $y \in \mathrm{ideal}(c)$, then there exists $d \in \mathcal{A}_{I_0}$ such that $c \overset{w}{\dashrightarrow}_{\mathcal{A}} d$ and $z \in \mathrm{ideal}(d)$.*
2. *For every $z \in X$, $w \in \Sigma^*$ and $c, d \in \mathcal{K}_{I_0}$, if $c \overset{w}{\dashrightarrow}_{\mathcal{K}} d$ and $z \in \mathrm{ideal}(d)$, then there exist $y \in \mathrm{ideal}(c)$, $w' \succeq w$ and $z' \geq z$ such that $y \xrightarrow{w'} z'$.*

Proof. We only prove (2). The proof is by induction on $|w|$. If $|w| = 0$, then $w = \varepsilon$. We stress the fact that even though w is empty, d might differ from c since $\mathcal{K}_{I_0}$ contains ε-transitions. However, by definition of $\mathcal{K}_{I_0}$, we know that $\mathrm{ideal}(d) \subseteq \mathrm{ideal}(c)$. Therefore, $z \in \mathrm{ideal}(c)$, and we are done since $z \xrightarrow{\varepsilon} z$.

Suppose that $|w| > 0$. Assume the claim holds for every word of length less than $|w|$. There exist $u, v \in \Sigma^*$, $a \in \Sigma$ and $d' \in \mathcal{K}_{I_0}$ such that $w = uav$, $c \overset{u}{\dashrightarrow}_{\mathcal{K}} d' \overset{a}{\dashrightarrow}_{\mathcal{K}} d \overset{v}{\dashrightarrow}_{\mathcal{K}} d$ and d' is the parent of d in T_{I_0}. Let $I \overset{\mathrm{def}}{=} \mathrm{ideal}(c)$, $J \overset{\mathrm{def}}{=} \mathrm{ideal}(d')$, $K \overset{\mathrm{def}}{=} \mathrm{ideal}(d)$, and $K' \overset{\mathrm{def}}{=} a(J)$. By induction hypothesis, there exist $y_K \in K$, $v' \succeq v$ and $z' \geq z$ such that $y_K \xrightarrow{v'} z'$.

- If $K = K'$, then $J \overset{a}{\rightsquigarrow} K$. By definition of $\overset{a}{\rightsquigarrow}$, there exist $y_J \in J$ and $y'_K \geq y_K$ such that $y_J \xrightarrow{a} y'_K$. By induction hypothesis, there exist $y_I \in I$, $u' \succeq u$ and $y'_J \geq y_J$ such that $y_I \xrightarrow{u'} y'_J$. By strong monotonicity of $\mathcal{S}$, there exists $z'' \geq z'$ such that $y_I \xrightarrow{u'av'} z''$. We are done since $u'av' \succeq uav$.
- If $K \neq K'$, then K was obtained through an acceleration. Therefore, $K = \sigma^\infty(K')$ for some $\sigma \in \Sigma^+$. This implies that $y_K \in \sigma^k(K')$ for some $k \in \mathbb{N}$. Let $L \overset{\mathrm{def}}{=} \sigma^k(K')$. Note that $J \overset{a}{\rightsquigarrow} K' \overset{\sigma^k}{\rightsquigarrow} L$. By Prop. 3(2), there exist $y_J \in J$ and $y'_K \geq y_K$ such that $y_J \xrightarrow{a\sigma^k} y'_K$. By induction hypothesis, there exist $y_I \in I$, $u' \succeq u$ and $y'_J \geq y_J$ such that $y_I \xrightarrow{u'} y'_J$. By strong monotonicity of $\mathcal{S}$, there exists $z'' \geq z'$ such that $y_I \xrightarrow{u'av^kv'} z''$. ◀

4.2 Effectiveness of the algorithm

The Ideal Karp-Miller algorithm can be implemented provided that (1) ideals can be effectively manipulated, (2) inclusion of ideals can be tested, (3) $\mathrm{Post}_{\widehat{\mathcal{S}}}(I)$ can be computed for every ideal I, and (4) $w^\infty(I)$ can be computed for every ideal I and sequence w. A class of WSTS satisfying (1–3) is called *completion-post-effective*, and a class satisfying (4) is called ∞*-completion-effective*. By Theorem 10, we obtain the following result:

▶ **Theorem 12.** *Let $\mathcal{C}$ be a completion-post-effective and ∞-completion-effective class of very-WSTS. The ideal decomposition of $\downarrow \mathit{Post}^*_{\mathcal{S}}(x)$ can be computed for every $\mathcal{S} = (X, \rightarrow, \leq) \in C$ and $x \in X$. In particular, coverability for $\mathcal{C}$ is decidable.*

5 A characterization of acceleration levels

We pause for a moment, and give a precise characterization of ideals that have finitely many levels. We shall then discuss some extensions briefly, beyond the finitely many level case.

Let Z be a well-founded partially ordered set, abstracting away from the case $Z = \mathrm{Idl}(X)$. The *rank* of $z \in Z$, denoted $\mathrm{rk}\, z$, is the ordinal defined inductively by $\mathrm{rk}\, z \overset{\mathrm{def}}{=} \sup\{\mathrm{rk}\, y + 1 : y < z\}$, where $\sup(\emptyset) \overset{\mathrm{def}}{=} 0$. The *rank* of Z is defined as $\mathrm{rk}\, Z \overset{\mathrm{def}}{=} \sup\{\mathrm{rk}\, z + 1 : z \in Z\}$. We say that a sequence $z_0, z_1, \ldots \in Z$ is an *acceleration candidate* if $z_1 < z_2 < \cdots < z_i < \cdots$. Such an acceleration candidate *goes through* a set A if $z_i \in A$ for some $i \in \mathbb{N}$, and is *below* $z \in Z$ if $z_i \leq z$ for every $i \in \mathbb{N}$. We define a family of sets $A_\alpha(Z)$ closely related to levels of ideals:

▶ **Definition 13.** Let Z be a partially ordered set. Let $A_0(Z) \stackrel{\text{def}}{=} \emptyset$. For every ordinal $\alpha > 0$, $A_\alpha(Z)$ is the set of elements $z \in Z$ such that every acceleration candidate below z goes through $A_\beta(Z)$ for some $\beta < \alpha$.

Observe that $A_\alpha(Z) \subseteq A_\beta(Z)$ for every $\alpha \leq \beta$, and that $A_n(\mathbb{N}^d_\omega)$ is the set of d-tuples with less than n components equal to ω. It is easily shown that $A_n(\mathrm{Idl}(X))$ is the upward closure of the complement of $\mathrm{Acc}_n(X)$. Consequently, $A_n(\mathrm{Idl}(X)) = \mathrm{Idl}(X)$ if and only if $\mathrm{Acc}_n(X) = \emptyset$, and we can bound levels of $\mathrm{Idl}(X)$ by means of $A_n(\mathrm{Idl}(X))$.

Let us first show that $A_n(Z)$ is exactly the set of elements of rank less than $\omega \cdot n$. This rests on the following, which is perhaps less obvious than it seems.

▶ **Lemma 14.** *Let Z be a countable wpo. For every $z \in Z$ such that $\mathrm{rk}\, z$ is a limit ordinal, z is the supremum of some acceleration candidate $z_0 < z_1 < \cdots$. Moreover, for any given ordinal $\beta < \mathrm{rk}\, z$, the acceleration candidate can be chosen such that $\beta \leq z_i$ for every $i \in \mathbb{N}$.*

This fails if Z is not countable: take $Z = \omega_1 + 1$, where ω_1 is the first uncountable ordinal, then $\omega_1 \in Z$ is not the supremum of countably many ordinals $< \omega_1$. This also fails if Z is not wqo, even when Z is well-founded: consider the set with one root r above chains of length n, one for each $n \in \mathbb{N}$: $\mathrm{rk}\, r = \omega$, but there is no acceleration candidate below r.

Proof. Let $\alpha \stackrel{\text{def}}{=} \mathrm{rk}\, z$. A *fundamental sequence* for α is a monotone sequence of ordinals strictly below α whose supremum equals α. Fundamental sequences exist for all countable limit ordinals, in particular for α, since Z is countable (e.g. see [22]). Pick one such fundamental subsequence $(\gamma_i)_{i\in\mathbb{N}}$. Replacing γ_i by $\sup(\beta, \gamma_i)$ if necessary, we may assume that $\beta \leq \gamma_m$ for every $i \in \mathbb{N}$. By the definition of rank, for every $i \in \mathbb{N}$, there is an element $z_i < z$ of rank at least γ_i. Since Z is well-quasi-ordered, we may extract a non-decreasing subsequence from $(z_i)_{i\in\mathbb{N}}$. Without loss of generality, assume that $z_0 \leq z_1 \leq \cdots$. If all but finitely many of these inequalities were equalities, then z would be equal to z_i for m large enough, but that is impossible since $z_i < z$. We can therefore extract a strictly increasing subsequence from $(z_i)_{i\in\mathbb{N}}$. This is an acceleration sequence, its supremum is z, and $\beta \leq \gamma_i \leq z_i$ for every i. ◀

▶ **Lemma 15.** *Let Z be a countable wpo, and let $n \in \mathbb{N}$. For every $z \in Z$, $\mathrm{rk}\, z < \omega \cdot n$ if and only if $z \in A_n(Z)$.*

Proof. $\Rightarrow$) By induction on n. The case $n = 0$ is immediate. Let $n \geq 1$. Given any acceleration candidate $z_1 < z_2 < \cdots$ below z, we must have $\mathrm{rk}\, z_1 < \mathrm{rk}\, z_2 < \cdots < \mathrm{rk}\, z$. Since $\mathrm{rk}\, z < \omega \cdot n$, there exist $\ell, m \in \mathbb{N}$ with $\ell < n$ such that $\mathrm{rk}\, z = \omega.\ell + m$. Therefore, $\mathrm{rk}\, z_i \geq \omega \cdot \ell$ for only finitely many i. In particular, there exists some i such that $\mathrm{rk}\, z_i < \omega \cdot \ell$. Since $\ell < n$, we have $\mathrm{rk}\, z_i < \omega \cdot (n-1)$. By induction hypothesis, $z_i \in A_{n-1}(Z)$, and hence $z \in A_n(Z)$.

$\Leftarrow$) We show by induction on n that $\mathrm{rk}\, z \geq \omega \cdot n$ implies $z \notin A_n(Z)$. The case $n = 0$ is immediate. Let $n \geq 1$. In general, $\mathrm{rk}\, z$ is not a limit ordinal, but can be written as $\alpha + \ell$ for some limit ordinal α and some $\ell \in \mathbb{N}$. By definition of rank, z is larger than some element of rank $\alpha + (\ell - 1)$, which is itself larger than some element of rank $\alpha + (\ell - 2)$, and so on. Iterating this way, we find an element $y \leq z$ of rank exactly α. Since $\mathrm{rk}\, y$ is a limit ordinal, Lemma 14 entails that y is the supremum of some acceleration candidate $z_0 < z_1 < \cdots$. Moreover, since $\omega \cdot (n-1) < \mathrm{rk}\, y$, we may assume that $\mathrm{rk}\, z_i \geq \omega \cdot (n-1)$ for every $i \in \mathbb{N}$. By induction hypothesis, $z_i \notin A_{n-1}(Z)$ for every $i \in \mathbb{N}$, and hence $z \notin A_n(Z)$. ◀

▶ **Theorem 16.** *Let X be a countable wqo such that* $\mathrm{Idl}(X)$ *is well-quasi-ordered by inclusion*[3]*. The following holds:* $\mathrm{Idl}(X)$ *has finitely many levels if and only if* $\mathrm{rk}\,\mathrm{Idl}(X) < \omega^2$.

Proof. We apply Lemma 15 to $Z = \mathrm{Idl}(X)$, a wpo by assumption. For that, we need to show that Z is countable. There are countably many upwards-closed subsets, since they are all determined by their finitely many minimal elements. Downwards-closed subsets are in one-to-one correspondence with upwards-closed subsets, through complementation, hence are countably many as well, and ideals are particular downwards-closed subsets.

We conclude by noting that the following are equivalent: (1) $\mathrm{rk}\,\mathrm{Idl}(X) < \omega^2$; (2) $\mathrm{rk}\,\mathrm{Idl}(X) \leq \omega \cdot n$ for some $n \in \mathbb{N}$; (3) $A_n(\mathrm{Idl}(X)) = \mathrm{Idl}(X)$ for some $n \in \mathbb{N}$ (by Lemma 15); (4) $\mathrm{Acc}_n(X) = \emptyset$ for some $n \in \mathbb{N}$. ◀

While $\mathrm{rk}\,\mathrm{Idl}(\mathbb{N}^d) = \omega{\cdot}d{+}1 < \omega^2$, not all wqos X used in verification satisfy $\mathrm{rk}\,\mathrm{Idl}(X) < \omega^2$. For example, $\mathrm{rk}\,\mathrm{Idl}(\Sigma^*) = \omega^{|\Sigma|}{+}1$, for any finite alphabet Σ; a similar result holds for multisets over Σ.

Note that the IKM algorithm still terminates if, for each branch $B = (c_0 : \langle I_0, n_0\rangle, c_1 : \langle I_1, n_1\rangle, \ldots, c_k : \langle I_k, n_k\rangle, \ldots)$ of the Ideal Karp-Miller tree, $[B] \overset{\text{def}}{=} \{I \in \mathrm{Idl}(X) : \exists j, k \in \mathbb{N}, j \leq k \text{ and } I_j \subseteq I \subseteq I_k\}$ has rank less than ω^2. Indeed, the IKM algorithm terminates if and only if each branch B is finite, and the states involved in computing the branch, as well as all needed accelerations, are all included in $[B]$. Therefore, relaxing "$\mathrm{rk}\,\mathrm{Idl}(X) < \omega^2$" to the more technical condition "$\mathrm{rk}\,[B] < \omega^2$" may allow one to extend the notion of very-WSTS.

6 Model checking liveness properties for very-WSTS

In this section, we show how the Ideal Karp-Miller algorithm can be used to test whether a very-WSTS violates a liveness property specified by an LTL formula. Testing that $\mathcal{S}$ violates a property φ amounts to constructing a Büchi automaton $\mathcal{B}_{\neg\varphi}$ for $\neg\varphi$ and to test whether $\mathcal{B}_{\neg\varphi}$ accepts an infinite trace of $\mathcal{S}$. We first introduce positive very-WSTS, and show that repeated coverability is decidable for them under some effectiveness hypothesis. Then, we show how LTL model checking for positive very-WSTS reduces to repeated coverability.

6.1 Deciding repeated coverability

Let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a WSTS and let $x \in X$. We say that $w \in \Sigma^*$ is *positive for x* if there exists some $y \in X$ such that $x \xrightarrow{w} y$ and $x \leq y$. We say that $w \in \Sigma^*$ is *positive* if w is positive for every $x \in X$ such that $\mathrm{Post}\,(x, w) \neq \emptyset$. We say that a WSTS $\mathcal{S} = (X, \xrightarrow{\Sigma} \leq)$ is *positive* if for every $w \in \Sigma^*$, w is positive for some $x \in X$ if and only if w is positive.

We establish a necessary and sufficient condition for repeated coverability in terms of the stuttering automaton and positive sequences:

▶ **Proposition 17.** *Let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a positive very-WSTS, and let $x, y \in X$. State y is repeatedly coverable from x if and only if there are states c, d of the stuttering automaton $\mathcal{A}_{\downarrow x}$ and $w \in \Sigma^+$ such that $c \overset{w}{\dashrightarrow}_{\mathcal{A}} d$,* $\text{num-accel}(c) = \text{num-accel}(d)$*, w is positive and $y \in$* $\mathrm{ideal}(c)$.

Proposition 17 allows us to show the decidability of repeated coverability under the following effectiveness hypothesis. A class $\mathcal{C}$ of WSTS is *positive-effective* if there is an

[3] Recall that such a wqo is known as an ω^2-wqo [19]. That we find the ordinal ω^2 in the statement of Theorem 16 and in the notion of ω^2-wqo seems to be coincidental.

algorithm that decides, on input $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq) \in \mathcal{C}$ and a finite automaton A, whether the language of A contains a positive sequence. VAS, Petri nets and ω-Petri nets are positive-effective, since, for these models, testing whether a finite automaton A accepts some positive sequence amounts to computing the Parikh image of $L(A)$, which is effectively semilinear [32].

▶ **Theorem 18.** *Repeated coverability is decidable for completion-post-effective, ∞-completion-effective and positive-effective classes of positive very-WSTS.*

Proof. By Prop. 17, y is repeatedly coverable from x if and only if there are states c, d in $\mathcal{A}_{\downarrow x}$ and $w \in \Sigma^+$ such that:

$$c \overset{w}{\dashrightarrow}_{\mathcal{A}} d, \text{num-accel}(c) = \text{num-accel}(d), w \text{ is positive and } y \in \text{ideal}(c). \qquad (1)$$

We show how (1) can be tested. For every $c \in \mathcal{A}_{\downarrow x}$, let A_c be the finite automaton over alphabet Σ whose set of states is $Q_c \overset{\text{def}}{=} \{d \in \mathcal{A}_{\downarrow x} : c \overset{*}{\dashrightarrow}_{\mathcal{A}} d \text{ and num-accel}(c) = \text{num-accel}(d)\}$, the initial state is c, all states are accepting, and transitions are as in $\mathcal{A}_{\downarrow x}$. For every $d \in Q_c$ and $w \in \Sigma^*$, $c \overset{w}{\dashrightarrow}_{\mathcal{A}} d$ if and only if w is in the language L_c of A_c. Build a new finite automaton A_c^+ that recognizes $L_c \setminus \{\varepsilon\}$. Let $C_y \overset{\text{def}}{=} \{c \in \mathcal{A}_{\downarrow x} : y \in \text{ideal}(c)\}$. By (1), y is repeatedly coverable from x if and only if there exists $c \in C_y$ such that the language $L_c \setminus \{\varepsilon\}$ of A_c^+ contains a positive sequence. The latter is decidable since $\mathcal{C}$ is positive-effective, since A_c^+ can be constructed effectively for every c (because $\mathcal{A}_{\downarrow x}$ can, using the fact that $\mathcal{C}$ is completion-post-effective and ∞-completion-effective), and since we can build C_y by enumerating the states c of $\mathcal{A}_{\downarrow x}$, checking whether $y \in \text{ideal}(c)$ for each (item (2) in the definition of completion-post-effectiveness). ◀

6.2 From model checking to repeated coverability

We conclude this section by reducing LTL model checking to repeated coverability. Recall that a *Büchi automaton* $\mathcal{B}$ is a non-deterministic finite automaton $\mathcal{B} = (Q, \Sigma, \delta, q_0, F)$ interpreted over Σ^ω. An infinite word is acccepted by $\mathcal{B}$ if it contains an infinite path from q_0 labeled by w and visiting F infinitely often. We denote by $\mathrm{L}(\mathcal{B})$ the set of infinite words accepted by $\mathcal{B}$.

Let $\mathcal{B} = (Q, \Sigma, \delta, q_0, F)$ be a Büchi automaton and let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a WSTS. The product of $\mathcal{B}$ and $\mathcal{S}$ is defined as $\mathcal{B} \times \mathcal{S} \overset{\text{def}}{=} (Q \times X, \xrightarrow{\Sigma \times Q}, = \times \leq)$ where $(p, x) \xrightarrow{(a,r)} (q, y)$ if $(p, a, r) \in \delta, q = r$ and $x \xrightarrow{a} y$. The point in including r in the label is so that $\widehat{\mathcal{B} \times \mathcal{S}}$ is deterministic, a requirement for very-WSTS. For every WSTS $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$, we extend $\mathcal{S}$ with a new "minimal" element $\perp$ smaller than every other states, i.e. $\mathcal{S}_\perp \overset{\text{def}}{=} (X \cup \{x_\perp\}, \xrightarrow{\Sigma}, \leq_\perp)$ where transition relations are unchanged, and $\leq_\perp \overset{\text{def}}{=} \leq \cup \{(\perp, y) : y \in X \cup \{\perp\}\}$. It can be shown that if $\mathcal{S}$ is a positive very-WSTS, then $\mathcal{B} \times \mathcal{S}_\perp$ is also. Taking the product of $\mathcal{B}$ and $\mathcal{S}_\perp$ allows us to test whether a word of $\mathrm{L}(\mathcal{B})$ is also an infinite trace of $\mathcal{S}$:

▶ **Proposition 19.** *Let $\mathcal{B} = (Q, \Sigma, \delta, q_0, F)$ be a Büchi automaton, let $\mathcal{S} = (X, \xrightarrow{\Sigma}, \leq)$ be a very-WSTS, and let $x_0 \in X$. There exists $w \in \mathrm{L}(\mathcal{B}) \cap \omega\text{-Traces}_{\mathcal{S}}(x_0)$ if and only if there exists $q_f \in F$ such that $(q_f, \perp)$ is repeatedly coverable from (q_0, x_0) in $\mathcal{B} \times \mathcal{S}_\perp$.*

Theorem 18 and Proposition 19 imply the decidability of LTL model checking:

▶ **Theorem 20.** *LTL model checking is decidable for completion-post-effective, ∞-completion-effective and positive-effective classes of positive very-WSTS.*

Theorem 20 implies that LTL model checking for ω-Petri nets is decidable. This includes, and generalizes strictly, the decidability of termination in ω-Petri nets [24].

7 Discussion and further work

We have presented the framework of very-WSTS, for which we have given a Karp-Miller algorithm. This allowed us to show that ideal decompositions of coverability sets of very-WSTS are computable, and that LTL model checking is decidable under some additional assumptions. We have also characterized acceleration levels in terms of ordinal ranks. Finally, we have shown that downward traces inclusion is decidable for very-WSTS.

As future work, we propose to study well-structured models beyond very-WSTS for which there exist Karp-Miller algorithms, e.g. unordered data Petri nets (UDPN) [28, 27], or for which reachability is decidable, e.g. recursive Petri nets[4] [26] with strict monotonicity. It is conceivable that LTL model checking is decidable for such models. Our approach will have to be extended to tackle this problem. For example, UDPN do not have finitely many acceleration levels. To circumvent this issue, Hofman et al. [27] make use of two types of accelerations that can be nested. One type is prioritized to ensure that acceleration levels along a branch grow "fast enough" for the algorithm to terminate.

References

1 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Proc. 11th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 313–321, 1996.

2 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Inf. Comput.*, 160(1-2):109–127, 2000.

3 Parosh Aziz Abdulla and Bengt Jonsson. Undecidable verification problems for programs with unreliable channels. In *Proc.* 21st *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 316–327, 1994.

4 Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. On computing fixpoints in well-structured regular model checking, with applications to lossy channel systems. In *Proc.* 13th *International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, (LPAR)*, pages 347–361, 2006.

5 Nathalie Bertrand and Philippe Schnoebelen. Computable fixpoints in well-structured symbolic model checking. *Formal Methods in System Design*, 43(2):233–267, 2013.

6 Michael Blondin, Alain Finkel, and Pierre McKenzie. Handling infinitely branching WSTS. In *Proc. 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 13–25, 2014.

7 Michael Blondin, Alain Finkel, and Pierre McKenzie. Well behaved transition systems. *Logical Methods in Computer Science*, 2017 (accepted).

8 Robert 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* 60^{t}*extth birthday*, pages 189–198. North-Holland, 1975.

9 Pierre Chambart, Alain Finkel, and Sylvain Schmitz. Forward analysis and model checking for trace bounded WSTS. In *Proc.* 32nd *International Conference on Applications and Theory of Petri Nets*, 2011.

10 Pierre Chambart, Alain Finkel, and Sylvain Schmitz. Forward analysis and model checking for trace bounded WSTS. *Theor. Comput. Sci.*, 637:1–29, 2016.

[4] Recursive Petri nets are WSTS for the tree embedding.

11 Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *Proc. 25th International Colloquium Automata, Languages and Programming (ICALP)*, pages 103–115, 1998.

12 E. Allen Emerson and Kedar S. Namjoshi. On model checking for non-deterministic infinite-state systems. In *Proc.* 13th *IEEE Symposium on Logic in Computer Science (LICS)*, pages 70–80, 1998.

13 Paul Erdős and Alfred Tarski. On families of mutually exclusive sets. *Annals of Mathematics*, 2(44):315–329, 1943.

14 Javier Esparza. On the decidability of model checking for several μ-calculi and Petri nets. In *Proc.* 19th *International Colloquium on Trees in Algebra and Programming (CAAP)*, pages 115–129, 1994.

15 Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *Proc.* 14th *Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 352–359, 1999.

16 Alain Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In *Proc.* 14th *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 499–508, 1987.

17 Alain Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144–179, 1990.

18 Alain Finkel and Jean Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In *STACS'09*, pages 433–444, Freiburg, Germany, 2009. Leibniz-Zentrum für Informatik, Intl. Proc. in Informatics 3.

19 Alain Finkel and Jean Goubault-Larrecq. Forward analysis for WSTS, part II: complete WSTS. *Logical Methods in Computer Science*, 8(3), 2012.

20 Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, 2004.

21 Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

22 Thomas Forster. A tutorial on countable ordinals. Available from the Web at `https://www.dpmms.cam.ac.uk/~tf/fundamentalsequence.pdf`, 2010. Read on Feb. 03, 2017.

23 Roland Fraïssé. Theory of relations. *Studies in Logic and the Foundations of Mathematics*, 118:1–456, 1986.

24 Gilles Geeraerts, Alexander Heußner, M. Praveen, and Jean-François Raskin. ω-Petri nets: Algorithms and complexity. *Fundamenta Informaticae*, 137(1):29–60, 2015.

25 Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *J. Comput. Syst. Sci.*, 72(1):180–203, 2006. `doi:10.1016/j.jcss.2005.09.001`.

26 Serge Haddad and Denis Poitrenaud. Recursive Petri nets. *Acta Inf.*, 44(7-8):463–508, 2007.

27 Piotr Hofman, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, Sylvain Schmitz, and Patrick Totzke. Coverability trees for Petri nets with unordered data. In *FoSSaCS*, pages 445–461, 2016.

28 Reiner Hüchting, Rupak Majumdar, and Roland Meyer. Bounds on mobility. In *Proc.* 25th *International Conference on Concurrency Theory (CONCUR)*, pages 357–371, 2014.

29 Richard M. Karp and Raymond E. Miller. Parallel program schemata: a mathematical model for parallel computation. In *Proc. 8th Annual Symposium on Switching and Automata Theory*, pages 55–61. IEEE Computer Society, 1967.

30 E. V. Kouzmin, Nikolay V. Shilov, and Valery A. Sokolov. Model checking mu-calculus in well-structured transition systems. In *Proc.* 11th *International Symposium on Temporal Representation and Reasoning (TIME)*, pages 152–155, 2004.

31 J.D. Lawson, M. Mislove, and H. Priestley. Ordered sets with no infinite antichains. *Discrete Mathematics*, 63(2):225–230, 1987.

32 Rohit J. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

33 Maurice Pouzet. Relations non reconstructibles par leurs restrictions. *Journal of Combinatorial Theory, Series B*, 26(1):22–34, 1979.

34 Maurice Pouzet and Nejib Zaguia. Dimension de Krull des ensembles ordonnés. *Discrete Mathematics*, 53:173–192, 1985.

35 Fernando Rosa-Velardo and María Martos-Salgado. Multiset rewriting for the verification of depth-bounded processes with name binding. *Inf. Comput.*, 215:68–87, 2012.

36 Fernando Rosa-Velardo, María Martos-Salgado, and David de Frutos-Escrig. Accelerations for the coverability set of Petri nets with names. *Fundamenta Informaticae*, 113(3-4):313–341, 2011.

37 Philippe Schnoebelen. Lossy counter machines decidability cheat sheet. In *Proc. 4th International Workshop on Reachability Problems (RP)*, pages 51–75, 2010.

38 Rüdiger Valk. Self-modifying nets, a natural extension of Petri nets. In *Proc. 5th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 464–476, 1978.

39 Rüdiger Valk and Matthias Jantzen. The residue of vector sets with applications to decidability problems in Petri nets. *Acta Inf.*, 21:643–674, 1985.

40 Kumar N. Verma and Jean Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, 2005.

41 Damien Zufferey, Thomas Wies, and Thomas A. Henzinger. Ideal abstractions for well-structured transition systems. In Viktor Kuncak and Andrey Rybalchenko, editors, *Verification, Model Checking, and Abstract Interpretation - 13th International Conference, VMCAI 2012, Philadelphia, PA, USA, January 22-24, 2012. Proceedings*, volume 7148 of *Lecture Notes in Computer Science*, pages 445–460. Springer, 2012. doi:10.1007/978-3-642-27940-9_29.