

Handling Infinitely Branching WSTS*

Michael Blondin¹, Alain Finkel², and Pierre McKenzie¹

¹ Université de Montréal and ENS Cachan

{blondimi,mckenzie}@iro.umontreal.ca

² ENS Cachan

alain.finkel@lsv.ens-cachan.fr

Abstract. Most decidability results concerning well-structured transition systems apply to the *finitely branching* variant. Yet some models (inserting automata, ω -Petri nets, ...) are naturally infinitely branching. Here we develop tools to handle infinitely branching WSTS by exploiting the crucial property that in the (ideal) completion of a well-quasi-ordered set, downward-closed sets are *finite* unions of ideals. Then, using these tools, we derive decidability results and we delineate the undecidability frontier in the case of the termination, the control-state maintainability and the coverability problems. Coverability and boundedness under new effectivity conditions are shown decidable.

1 Introduction

Well-structured transition systems (WSTS) [12,11,2] as a general class of infinite-state systems have spawned decidability results for important problems such as termination, boundedness, control-state maintainability and coverability. WSTS consist of a (usually infinite) well ordered set of states, together with a monotone transition relation. WSTS have found multiple *uses*: in settling the decidability status of reachability and coverability for graph transformation systems [4,22], in the forward analysis of depth-bounded processes [26,27], in the verification of parameterized protocols [10] and the verification of multi-threaded asynchronous software [21]. WSTS remain under development and are actively being investigated [13,14,18,25,5,24].

Most existing decidability results for WSTS apply to the *finitely branching* variant. However, WSTS such as inserting FIFO automata [7], inserting automata [6] and ω -Petri nets [17], that can arbitrarily increase some values, are intrinsically *infinitely branching*, and any finitely branching WSTS parameterized with an infinite set of initial states (such as broadcast protocols [10]) also inherits an infinitely branching state. For instance, Geeraerts, Heußner, Praveen and Raskin argue in [17] that parametric concurrent systems with dynamic

* Supported by the French Agence Nationale de la Recherche, REACHARD (grant ANR-11-BS02-001), by the Fonds québécois de la recherche sur la nature et les technologies, by the Natural Sciences and Engineering Research Council of Canada and by the “Chaire DIGITEO, ENS Cachan - École Polytechnique”.

thread creation can naturally be modelled by some classes of infinitely branching systems, like ω -Petri nets, i.e. Petri net with arcs that can consume/create arbitrarily many tokens.

An outcome of our work is that the finite tree construction technique can be recovered, even in the infinitely branching case, for the purpose of deciding the boundedness problem for example.

The primary motivation for this paper is to explore the decidability status of the termination, boundedness, control-state maintainability and coverability problems for infinitely branching (general) WSTS. For the coverability problem, known to be decidable for WSTS fulfilling the so-called prebasis computability hypothesis [2], we wish to draw from the recent algebra-theoretic characterizations of downward-closed sets [13] and conceive of a post-oriented computability hypothesis suitable for the design of a forward algorithm. (Indeed, forward algorithms are arguably more intuitive than backward algorithms and post-oriented computability more easily verified than prebasis computability, where prebasis computability means computability of a finite basis of the upward closure of the set of immediate predecessors, the testing of which is provably undecidable in some WSTS.) Our contributions are the following:

1. As technical tools, we simplify and extend the analysis of the completion of a general WSTS and we relate the behavior of a WSTS to that of its completion. In particular, we provide a general presentation of the completion that is much less daunting than the presentations currently available in the literature. This sets the stage for exploiting the main property of the completion of a WSTS, namely, the expressibility of any downward-closed set as a (unique, as shown here) finite union of ideals, in the design of algorithms.
2. We uncover a new termination property (called *strong* termination) that is computationally equivalent to the usual termination property for finitely branching WSTS but that subtly differs from it in the presence of infinite branching. Indeed, we exhibit WSTS for which strong termination is decidable yet the usual termination is undecidable. A similar subtle issue arises as well in our generalization of the maintainability problem to infinitely branching.
3. We generalize most decidability results mentioned for finitely branching WSTS earlier to the infinitely branching case. This requires carefully tracking the effectiveness and the monotonicity conditions which support decidability. When possible, we delineate the frontier between decidability for a problem and the undecidability that results from dropping one of these conditions. The new decidability results for (strong) termination and (strong) maintainability exploit the completion. The new algorithm for coverability uses a forward strategy coupled with a post-oriented computability hypothesis.

Our work further highlights the naturalness of the class of ω^2 -WSTS. Indeed our decidability results apply in one blow to known classes of infinitely branching WSTS like inserting FIFO automata [7], inserting automata [6], ω -Petri nets [17] and broadcast protocols [10].

Section 2 below introduces notation and preliminaries. Section 3 surveys known decidabilities and exhibits some undecidabilities. Section 4 develops our tools to handle infinite branching. Section 5 contains the bulk of our decidability results for infinitely branching WSTS. Section 6 summarizes our contribution and suggests future work.

2 WSTS

Let X be a set and \leq a quasi-ordering on X (\leq reflexive and transitive), then \leq is a *well-quasi-ordering* (wqo) if for every infinite sequence x_0, x_1, \dots of elements $x_n \in X$, there exist $i < j$ such that $x_i \leq x_j$. It is well-known that \mathbb{N}^d is well-quasi-ordered under $(x_1, \dots, x_d) \leq_{\mathbb{N}^d} (x'_1, \dots, x'_d)$ where the latter means that $\forall i x_i \leq x'_i$ (Dickson's Lemma). We extend \mathbb{N} to \mathbb{N}_ω by adding an element ω verifying $\omega \geq_{\mathbb{N}_\omega} x$ for all $x \in \mathbb{N}_\omega$. The set \mathbb{N}_ω^d is also well-quasi-ordered. We simply write \leq for $\leq_{\mathbb{N}}$ and $\leq_{\mathbb{N}_\omega}$ when there is no ambiguity.

Recall that a *WSTS* is an ordered transition system $S = (X, \rightarrow_S, \leq)$ such that \leq is a well-quasi-ordering on X , and the relation $\rightarrow_S \subseteq X \times X$ is monotone (or compatible) with \leq meaning that for all x, y, x' such that $x \rightarrow_S y$ and $x \leq x'$, there exists a state y' such that $x' \xrightarrow{*}_S y'$ and $y \leq y'$. WSTS thus satisfy a general monotony by definition. There exist other variations of monotony:

strong: $x \rightarrow_S y \wedge x' \geq x \implies x' \rightarrow_S y' \geq y$,

stuttering: $x \rightarrow_S y \wedge x' \geq x \implies x' = x'_0 \rightarrow_S \dots \rightarrow_S x'_k \rightarrow_S y' \geq y, \forall i x'_i \geq x$,

transitive: $x \rightarrow_S y \wedge x' \geq x \implies x' \xrightarrow{+}_S y' \geq y$,

strict: $x \rightarrow_S y \wedge x' > x \implies x' \xrightarrow{*}_S y' > y$.

Strong monotony implies stuttering monotony which implies transitive monotony.

We denote, as usual, $\text{Pre}_S(x) = \{y : y \rightarrow_S x\}$, $\text{Post}_S(x) = \{y : x \rightarrow_S y\}$, $\text{Pre}_S(T) = \bigcup_{x \in T} \text{Pre}_S(x)$ and $\text{Post}_S(T) = \bigcup_{x \in T} \text{Post}_S(x)$.

Throughout this paper, WSTS will be assumed *effective* in the following sense: (1) the set of states X is r.e. (which suffices to compute $\text{Post}_S(x)$ when $|\text{Post}_S(x)|$ is known and finite); (2) the transition relation is decidable, i.e., the WSTS comes equipped with an algorithm that can decide, given $x, y \in X$, whether $x \rightarrow_S y$ or equivalently whether $y \in \text{Post}_S(x)$; (3) the quasi-ordering \leq is decidable, i.e., the WSTS also comes equipped with an algorithm that can decide, given $x, y \in X$, whether $x \leq y$. Forward analysis techniques for (finitely branching) WSTS typically compute the finite set $\text{Post}_S(x)$, which is made possible by assuming Post_S computable. Because our new setting allows $\text{Post}_S(x)$ to be infinite, we need to adapt this assumption. Our “post-effectivity” notion mildly weakens the usual hypothesis of “being able to compute Post_S ”:

Definition 2.1. *A transition system $S = (X, \rightarrow_S)$ is post-effective if S is effective and $f : X \rightarrow \mathbb{N} \cup \{\text{“infinite”}\}$ given by $f(x) = |\text{Post}_S(x)|$ is computable.*

Transition systems defined by a finite set of recursive functions are typical examples of finitely branching systems and they will be called *functional*. Let \mathcal{F}_d

denote the set of WSTS whose transitions relation is prescribed by finitely many increasing functions f from \mathbb{N}^d to \mathbb{N}^d (i.e. $x \leq y \implies f(x) \leq f(y)$) which are also recursive (i.e., given by halting Turing machines); these WSTS are finitely branching and post-effective. Inserting FIFO automata [7], inserting automata [6] and ω -Petri nets [17] are post-effective infinitely branching WSTS.

Recall that an effective ordered transition system is said *essentially finite branching* [2] if the subset $\text{maxpost}(x)$ of maximal elements of $\text{Post}_S(x)$ is non empty, finite and computable. Some WSTS, e.g. ω -Petri nets, are post-effective but are not essentially finite branching and conversely, we can exhibit essentially finite branching WSTS that are not post-effective.

Post-effectivity (Definition 2.1) is a weaker notion than “having a finite and computable Post_S ”. The weaker notion does imply “computable Post_S ” for effective WSTS that are finitely branching. Hence it is natural to ask whether the finitely branching property is decidable for post-effective WSTS. It is not:

Proposition 2.2. *Testing, given a post-effective WSTS S and $x_0 \in X$, whether there exists an execution $x_0 \xrightarrow{*}_S x$ such that $\text{Post}_S(x)$ is infinite is undecidable.*

Let $\uparrow T$ and $\downarrow T$ stand respectively for the set of states that are \geq and \leq some state in T . A set T is *upward closed* if $T = \uparrow T$ and *downward closed* if $T = \downarrow T$. An *upward basis* of a set T is a set B such that $T = \uparrow B$. An *ideal* I is a downward closed set that is also *directed*, i.e., $\forall a, b \in I, \exists c \in I$ such that $a \leq c$ and $b \leq c$. We note $\text{Ideals}(X)$ the set of ideals of an ordered set X . A *directed complete partial ordering* (dcpo) is an ordered set (X, \leq) such that every directed set $D \subseteq X$ has a least upper bound (lub) in X : for instance, (\mathbb{N}, \leq) , with the usual notations, is not a dcpo since the directed set \mathbb{N} has no lub in \mathbb{N} ; if we add the lub ω to \mathbb{N} , then $(\mathbb{N}_\omega, \leq)$ is a dcpo. There is a way to add all lubs to any ordered set (X, \leq) , that is called the *ideal completion*, since each element $x \in X$ can be identified with $\downarrow x \in \text{Ideals}(X)$ and since it is well-known that $(\text{Ideals}(X), \subseteq)$ is a dcpo [3,13]. We will consider the following problems for WSTS, where the input to each problem is an effective WSTS $S = (X, \rightarrow_S, \leq)$ and a state $x_0 \in X$, together with an $x \in X$ in the case of coverability, and a set $t_1, \dots, t_n \in X$ in the case of the maintainability problem:

- Coverability: \exists execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_k \geq x$?
- Boundedness: $\text{Post}_S^*(x_0)$ is infinite?
- Termination: \nexists infinite execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots$?
- Strong termination: $\exists k \in \mathbb{N}$ s.t. $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_m \implies m \leq k$?
- Control-state maintainability: \exists computation (i.e. an infinite execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots$ or a finite execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_k$ that cannot be further extended) such that $\forall i x_i \in \uparrow \{t_1, \dots, t_n\}$?
- Strong control-state maintainability: $\forall k \in \mathbb{N}, \exists$ execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_m$ such that $m \geq k$ and $\forall i x_i \in \uparrow \{t_1, \dots, t_n\}$?

3 Decidability for WSTS

Recall that a WSTS $S = (X, \rightarrow_S, \leq)$ has a *computable prebasis* [11,2] if the WSTS comes equipped with a computable function that maps each $x \in X$ to

some finite basis of the upward closed set $\uparrow \text{Pre}_S(\uparrow x)$. We summarize the four main decidability results known about (essentially) finite branching WSTS:

Theorem 3.1 ([12,11,2]).

- *Termination is decidable for post-effective finitely branching WSTS with transitive monotony [12], and for essentially finite branching effective WSTS with strong monotony [2].*
- *Boundedness is decidable for post-effective finitely branching WSTS with strict transitive monotony and well partial ordering [11].*
- *Control-state maintainability is decidable for post-effective finitely branching WSTS with stuttering monotony [11], and for essentially finite branching effective WSTS with strong monotony [2].*
- *Coverability is decidable for effective WSTS with prebasis computability [11,2].*

Theorem 3.1 states results exactly as they appear in the literature, but it would not be difficult to unify some of the hypotheses made here. For instance, termination can be shown decidable for essentially finite branching effective WSTS with transitive monotony. We defer a systematic treatment of this unification to a future version of the present paper.

Our goals in this paper are to extend the decidability of termination, boundedness and maintainability given by Theorem 3.1 to the more general case of *infinitely branching* WSTS. Our goal for the coverability problem is to investigate alternative effectivity hypotheses. We first note:

Theorem 3.2. *Termination is undecidable for post-effective WSTS with transitive (and even strong and strict) monotony.*

In Sect. 5, we prove boundedness decidable for *post-effective* infinitely branching WSTS with strict monotony and well partial ordering. By contrast, as exemplified by Petri nets with Reset [8], boundedness is well known to be undecidable for post-effective finitely branching WSTS with *non-strict* yet transitive (even strong) monotony and with well partial ordering. Concerning maintainability,

Theorem 3.3. *Control-state maintainability is undecidable for post-effective WSTS with stuttering (and even strong and strict) monotony.*

We now turn to coverability. Existing proofs that coverability is decidable need the prebasis hypothesis: Abdulla et al. use a backward algorithm [11,1] that computes a finite basis of $\uparrow \text{Pre}^*(\uparrow x)$ and Geeraerts et al. use a forward algorithm [18] that requires further hypotheses (i.e. restriction to an adequate domain of limits, a *mathematical* hypothesis subsequently shown superfluous [16,13]). Note that coverability for post-effective (even finitely branching) WSTS becomes undecidable without the prebasis hypothesis, as is the case for instance for WSTS in \mathcal{F}_2 (recall definition from Sect 2, i.e., WSTS composed of recursive increasing functions from \mathbb{N}^2 to \mathbb{N}^2) [15].

Prebasis computability is *sufficient* to ensure decidability of coverability. However, as we show in Prop. 3.4 below, prebasis computability is not *necessary*: there is a class of WSTS, namely \mathcal{F}_1 , for which coverability is decidable yet *no* prebasis function is computable.

Proposition 3.4. *Coverability for \mathcal{F}_1 is decidable, but no algorithm that takes as input $S \in \mathcal{F}_1$ and $x \in \mathbb{N}$ can systematically output a finite basis of $\uparrow \text{Pre}_S(\uparrow x)$.*

4 Handling Infinite Branching Finitely

In this section we prepare the ground for developing decision procedures capable of handling, under natural hypotheses, infinitely branching systems. First we would like the ability to compute finite representations of each term in the sequence $\downarrow x, \downarrow \text{Post}_S(\downarrow x), \downarrow \text{Post}_S(\downarrow \text{Post}_S(\downarrow x)), \dots$. This requires finitely representing downward closed sets, which is possible for wqo. This section describes how this is done and presents effective tools for doing it.

4.1 Downward Closed Sets and Ideals

It has long been known that in a wqo, any upward closed set has a finite basis; this is Dickson's lemma in (\mathbb{N}^k, \leq) and it is Higman's lemma in (Σ^*, \leq) when \leq is the subword relation. It has recently been discovered that a similar situation occurs for downward closed sets in wqo.

Theorem 4.1. [13] *Any downward closed subset in a wqo X is a finite union of ideals.*

The original proof of Theorem 4.1 needs a technical bridge between topological completions and ordering completions of a set. A short and self-contained proof of Theorem 4.1 was given by Goubault-Larrecq [19].

Theorem 4.3 below slightly refines Theorem 4.1. It shows that any downward closed set uniquely decomposes as a certain finite union of ideals. This requires:

Proposition 4.2. *Any ideal contained in a finite union of ideals is contained in one of these ideals. In particular, testing the inclusion of an ideal I in a union $J_1 \cup J_2 \cup \dots \cup J_k$ of ideals is equivalent to testing whether $I \subseteq J_j$ for some j such that $1 \leq j \leq k$.*

A finite union $D = \bigcup_{i=1}^m I_i$ of ideals will be said to *canonically* decompose D if the I_i 's are pairwise incomparable under inclusion. This terminology is justified:

Theorem 4.3. *Any downward closed subset in a wqo X admits a unique decomposition as a finite union of pairwise incomparable ideals. Therefore, a downward closed subset decomposes canonically as the union of its maximal ideals.*

Ideals in a wqo cannot necessarily be manipulated effectively. For instance, there exist some ordered countable sets X such that $\text{Ideals}(X)$ is not countable. Consider $X = \Sigma^*$, with the prefix ordering. Then $\text{Ideals}(X)$ is isomorphic to $\Sigma^* \cup \Sigma^\omega$ and is not countable when Σ contains at least two letters. However:

Proposition 4.4. *A wqo X is countable iff $\text{Ideals}(X)$ is countable.*

Fortunately, inclusion between ideals is decidable for well-quasi-ordered sets obtained by closing finite sets and closing natural numbers under finite products, disjoint sums, multiset operator and Kleene star (respectively with their natural associated orderings) [13]. Therefore inclusion of ideals of \mathbb{N}^d and inclusion of ideals of Σ^* are decidable.

4.2 Completion of WSTS

Recall that for a functional WSTS $S = (X, \xrightarrow{F}, \leq)$ where F is a finite set of increasing recursive functions $f : X \rightarrow X$, the *functional completion* [14] is defined by $\bar{S} = (\bar{X}, \xrightarrow{\bar{F}}, \subseteq)$ where $\bar{X} = \text{Ideals}(X)$ and \bar{F} is the set of functions $\bar{f} : \text{Ideals}(X) \rightarrow \text{Ideals}(X)$ defined by $\bar{f}(I) \stackrel{\text{def}}{=} \downarrow f(I)$ for every $f \in F$. We note that $\bar{f}(I)$ is an ideal if I is an ideal. Here we extend the completion process to any (infinitely) branching WSTS:

Definition 4.5. *The completion \hat{S} of a WSTS $S = (X, \rightarrow_S, \leq)$ is the ordered transition system $\hat{S} = (\hat{X}, \rightarrow_{\hat{S}}, \subseteq)$ where $\hat{X} = \text{Ideals}(X)$, and $I \rightarrow_{\hat{S}} J$ if J appears in the canonical decomposition of $\downarrow \text{Post}(I)$.*

Let $S = (X, \xrightarrow{F}, \leq)$ be a functional WSTS, then the following relation holds between \bar{S} , \hat{S} and S for every ideal $I \in \text{Ideals}(X)$:

$$\text{Post}_{\bar{S}}(I) = \bigcup_{\bar{f} \in \bar{F}} \bar{f}(I) = \bigcup_{f \in F} \downarrow f(I) = \bigcup_{J \in \text{Post}_S(I)} J = \downarrow \text{Post}_S(I).$$

Another good news is that:

Proposition 4.6. *The completion \hat{S} of any WSTS S is finitely branching.*

Moreover the completion computes exactly the downward closure of the reachability set of its original system.

Proposition 4.7. *Let $S = (X, \rightarrow_S, \leq)$ be a WSTS and $\text{Post}_{\hat{S}}^*(\downarrow x) = \{J_1, \dots, J_n\}$. We have $\downarrow \text{Post}_S^*(x) = J_1 \cup \dots \cup J_n$.*

A natural question that arises is whether the completion of a WSTS is also a WSTS. It does indeed have monotony:

Proposition 4.8. *Let $S = (X, \rightarrow_S, \leq)$ then \hat{S} has strong monotony.*

However, $(\text{Ideals}(X), \subseteq)$ is not always a wqo and therefore the completion is not always a WSTS. In fact, it is known to be a wqo iff (X, \leq) is a so-called ω^2 -wqo, a notion we will not define here. In general, a wqo is not necessarily a ω^2 -wqo and the typical counter-example is the Rado ordering [20]. Now, a result from Jancar [20] simplifies the characterization of ω^2 -wqos as follows: a wqo \leq is a ω^2 -wqo iff $\leq^\#$ is a wqo, where $\leq^\#$ is the Hoare ordering defined by $A \leq^\# B$ iff $\uparrow B \subseteq \uparrow A$.

Extending the terminology to WSTS, we obtain the following result generalizing the known result for functional WSTS [14]:

Theorem 4.9. *Let S be a WSTS, then \hat{S} is a WSTS iff S is a ω^2 -WSTS.*

We end this section with the observations that a WSTS inherits the strict monotony of its completion but not conversely, and that post-effectivity of a WSTS is independent from the post-effectivity of its completion.

Proposition 4.10. *Let (X, \rightarrow_S, \leq) be a WSTS. If \widehat{S} has strict monotony, then so does S . However, if S has strict monotony then \widehat{S} doesn't necessarily have it.*

Proposition 4.11. *There exists a post-effective WSTS whose completion is not post-effective. Conversely, there exists a non post-effective WSTS whose completion is post-effective.*

4.3 Post-effectiveness of Completions in Concrete Examples

An *affine net* S is a WSTS in \mathcal{F}_d in which the recursive functions are affine and a Petri net can be seen as an affine net where all matrices are the identity. An ω -Petri net [17] is an (extended) Petri net in which arcs can be labelled by positive integers or by ω . The completions of affine nets, ω -Petri nets and Lossy Channel Systems can be shown post-effective.

5 Decidability in Infinitely Branching Post-effective WSTS

5.1 (Strong) Termination

We are able to strengthen the hypotheses of Theorem 3.2 and to obtain: *termination is undecidable, even for post-effective ω^2 -WSTS with strong and strict monotony, and with post-effective completion* by reducing from structural termination for Transfer Petri nets [9].

When a WSTS is infinitely branching, its termination problem differs in a subtle way from its strong termination problem. We show the latter decidable under suitable hypotheses:

Theorem 5.1. *Strong termination is decidable for ω^2 -WSTS with transitive monotony and post-effective completion.*

Proving Theorem 5.1 requires comparing executions in a system with executions in its completion:

Proposition 5.2. *Let $S = (X, \rightarrow_S, \leq)$ be a WSTS, and $I, J \in \widehat{X}$. If $I \xrightarrow{k}_{\widehat{S}} J$, then for every $x_J \in J$ there exists $x_I \in I$, $y \in \uparrow x_J$ and $k' \in \mathbb{N}$ such that $x_I \xrightarrow{k'}_S y$. Moreover, if S has transitive monotony then $k' \geq k$; if S has strong monotony then $k' = k$.*

Proposition 5.3. *Let $S = (X, \rightarrow_S, \leq)$ be a WSTS and $x, y \in X$. If $x \xrightarrow{k}_S y$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \xrightarrow{k}_{\widehat{S}} J$.*

Proof sketch of Theorem 5.1. Consider a ω^2 -WSTS $S = (X, \rightarrow_S, \leq)$ such that \widehat{S} is post-effective. Finkel and Schnoebelen [11, Theorem 4.6] show that termination, and thus strong termination, is decidable for post-effective WSTS having

transitive monotony. By hypothesis, \widehat{S} is a WSTS and \widehat{S} has strong (and transitive) monotony by Prop. 4.8. Therefore, strong termination for \widehat{S} is decidable. From Prop. 5.2 and Prop. 5.3, no bound on the length of executions from x_0 exists in S iff no bound on the length of executions from $\downarrow x_0$ exists in \widehat{S} . Hence decidability of strong termination from x_0 in S follows from being able to decide strong termination from $\downarrow x_0$ in \widehat{S} . Note that we have implicitly assumed that a representation of $\downarrow x_0$ can be effectively computed. \square

5.2 Boundedness

Drawing from [8], we know that *boundedness is undecidable, even for finitely branching post-effective ω^2 -WSTS with strong (but not strict) monotony and post-effective completion*. Petri net with reset arcs are such a class.

It is known that for finitely branching post-effective WSTS with strict transitive monotony and a well partial ordering (wpo), the boundedness problem is decidable [11]. We generalize this result to (possibly) infinitely branching WSTS and we note that the hypothesis of transitive monotony was not necessary in the proof of [11]. The proof follows [11] by building a finite reachability tree, with the extra step of testing whether $\text{Post}_S(x)$ is infinite for each new node.

Theorem 5.4. *Boundedness is decidable for post-effective WSTS with strict monotony and with well partial ordering.*

5.3 (Strong) Control-State Maintainability

By a reduction from the termination problem, the hypotheses of Theorem 3.3 can be strengthened: *control-state maintainability is undecidable, even for post-effective ω^2 -WSTS with strong and strict monotony, and with post-effective completion*. By contrast, the *strong* variant of the problem introduced in this paper is decidable, under suitable hypotheses, for infinitely branching WSTS:

Theorem 5.5. *Strong control-state maintainability is decidable for ω^2 -WSTS with strong monotony and a post-effective completion.*

Before proving Theorem 5.5, we need Prop. 5.6 and Prop. 5.7 to relate covering executions in a WSTS to covering executions in its completion.

Proposition 5.6. *Let $S = (X, \rightarrow_S, \leq)$ be a WSTS with strong monotony and $\{t_1, \dots, t_n\} \subseteq X$. Let $I_0 \rightarrow_{\widehat{S}} I_1 \rightarrow_{\widehat{S}} \dots \rightarrow_{\widehat{S}} I_k$ be an execution such that for all $0 \leq j \leq k$ we have $I_j \in \uparrow_{\widehat{X}} \{\downarrow t_1, \dots, \downarrow t_n\}$. Then for every $y \in I_k$ there exists an execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_k$ such that $x_0 \in I_0$, $x_k \in \uparrow y$ and for all $0 \leq j \leq k$ we have $x_j \in \uparrow \{t_1, \dots, t_n\}$.*

Proof. Let I_0 be an execution of length 0 in \widehat{S} as described in the proposition, and let $y \in I_0$. By hypothesis, there exists t_i such that $\downarrow t_i \subseteq I_0$ and thus $t_i \in I_0$. Since I_0 is an ideal, there exists $x_0 \in I_0$ such that $x_0 \geq y$ and $x_0 \geq t_i$. Therefore the execution x_0 of length 0 in S meets all requirements.

Let $I_0 \rightarrow_{\widehat{S}} I_1 \rightarrow_{\widehat{S}} \dots \rightarrow_{\widehat{S}} I_k$ be an execution of length $k > 0$ in \widehat{S} as described in the proposition. By induction, for every $y \in I_k$ there exists an execution $x_1 \rightarrow_S x_2 \rightarrow_S \dots \rightarrow_S x_k$ such that $x_1 \in I_1$, $x_k \in \uparrow y$ and for all $1 \leq j \leq k$ we have $x_j \in \uparrow \{t_1, \dots, t_n\}$.

Since $x_1 \in I_1 \subseteq \downarrow \text{Post}_S(I_0)$, there exists $x_0 \in I_0$ and $y' \in \uparrow x_1$ such that $x_0 \rightarrow_S y'$. By hypothesis, there exists t_i such that $\downarrow t_i \subseteq I_0$ and thus $t_i \in I_0$. Since I_0 is an ideal, there exists $x'_0 \in I_0$ such that $x'_0 \geq x_0$ and $x_0 \geq t_i$. By strong monotony, there exists $x'_1 \geq y'$ such that $x'_0 \rightarrow_S x'_1$.

Moreover, applying strong monotony to $x_1 \rightarrow_S x_2 \rightarrow_S \dots \rightarrow_S x_k$ with $x'_1 \geq x_1$, we obtain an execution $x'_1 \rightarrow_S x'_2 \rightarrow_S \dots \rightarrow_S x'_k$ such that for all $1 \leq j \leq k$ we have $x'_j \geq x_j$. Therefore, $x'_0 \rightarrow_S x'_1 \rightarrow_S \dots \rightarrow_S x'_k$, $x'_0 \in I_0$, $x'_k \in \uparrow y$ and for all $0 \leq j \leq k$ we have $x'_j \in \uparrow \{t_1, \dots, t_n\}$. \square

Proposition 5.7. *Let $S = (X, \rightarrow_S, \leq)$ be a WSTS and $\{t_1, \dots, t_n\} \subseteq X$. Let $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_k$ be an execution such that for all $0 \leq j \leq k$ we have $x_i \in \uparrow \{t_1, \dots, t_n\}$. Then for every ideal $I_0 \supseteq \downarrow x_0$ there exists an execution $I_0 \rightarrow_{\widehat{S}} I_1 \rightarrow_{\widehat{S}} \dots \rightarrow_{\widehat{S}} I_k$ such that $I_k \supseteq \downarrow x_k$ and for all $0 \leq j \leq k$ we have $I_j \in \uparrow_{\widehat{X}} \{\downarrow t_1, \dots, \downarrow t_n\}$.*

Proof of Theorem 5.5. By Prop. 5.6 and Prop. 5.7 there exists an execution $x_0 \rightarrow_S x_1 \rightarrow_S \dots \rightarrow_S x_k$ such that for all $0 \leq j \leq k$ we have $x_j \in \uparrow \{t_1, \dots, t_n\}$ iff there exists an execution $I_0 \rightarrow_{\widehat{S}} I_1 \rightarrow_{\widehat{S}} \dots \rightarrow_{\widehat{S}} I_k$ such that for all $0 \leq i \leq k$ we have $I_j \in \uparrow_{\widehat{X}} \{\downarrow t_1, \dots, \downarrow t_n\}$. Therefore, it suffices to solve the problem in \widehat{S} with $\downarrow x_0$ and $\{\downarrow t_1, \dots, \downarrow t_n\}$.

The algorithm from [11] solving the control-state maintainability problem for finitely branching post-effective WSTS with stuttering monotony can easily be adapted to solve strong control-state maintainability for finitely branching WSTS. Since \widehat{S} is a post-effective WSTS by hypothesis and has strong (and stuttering) monotony by Prop. 4.8, we obtain an algorithm.

More specifically, it suffices to build the finite reachability tree of \widehat{S} and verify that it contains a maximal path labelled I_0, I_1, \dots, I_k with $I_j \in \uparrow_{\widehat{X}} \{\downarrow t_1, \dots, \downarrow t_n\}$ for every $0 \leq j \leq k$ and $I_j \subseteq I_k$ for some $0 \leq j < k$. \square

5.4 Coverability

Some classes of WSTS admit both post-effective completions and prebasis computability, e.g., WSTS from \mathcal{F}_d where the recursive increasing functions have computable limits (called ω -well-structured nets in [15]). Therefore, coverability was already known to be decidable for these classes. However, the following Theorem 5.8 yields an algorithm that relies on evaluating Post_S on ideals rather than Pre_S on upward closed sets. Often this is more efficient, e.g., it is easier to evaluate affine functions in \mathbb{N}_ω^d than inverting them.

Theorem 5.8. *Coverability is decidable for WSTS having a post-effective completion.*

Proof. Let $S = (X, \rightarrow_S, \leq)$ be a post-effective WSTS and $x_0 \in X$.

Coverability is semi-decidable by iteratively building larger portions of the reachability tree looking for a path with some state $x' \geq x$.

We note that x is coverable from x_0 in S iff there exists an ideal $I \supseteq \downarrow x$ reachable from $\downarrow x_0$ in \widehat{S} . To prove that non-coverability is semi-decidable, one enumerates all the downward closed sets D_i (as finite unions of ideals) that are inductive invariants, i.e., such that $x_0 \in D_i$ and $\downarrow \text{Post}_S(D_i) \subseteq D_i$. If x is not coverable, a downward closed set D_i such that $x \notin D_i$ will inevitably be found.

The inclusion $\downarrow \text{Post}_S(D_i) \subseteq D_i$ is decidable for WSTS whose completion is post-effective since there is an algorithm, which runs $\text{Post}_{\widehat{S}}$ on D_i (expressed as the union $J_1 \cup \dots \cup J_m$ of ideals) to obtain ideals I_1, \dots, I_n such that $\downarrow \text{Post}_S(D_i) = \cup_{1 \leq i \leq m} \cup_{I \in \text{Post}_{\widehat{S}}(J_i)} \downarrow I = I_1 \cup \dots \cup I_n$. Now Prop. 4.2 says that this inclusion $I_1 \cup I_2 \cup \dots \cup I_n \subseteq J_1 \cup J_2 \cup \dots \cup J_m$ is decidable. \square

The technique of enumerating inductive invariants, used in our coverability algorithm, was already used by Pachl in 1982 to provide a witness of non-reachability for finite automata communicating through fifo channels, having recognizable reachability sets (Corollary 9.6 in [23]). More recently, Raskin et al. [18,16] also used enumeration of inductive invariants to provide forward algorithms for deciding coverability of WSTS. Note that their forward algorithms use the prebasis hypothesis while we appeal to post-effective completion.

6 Conclusion and Further Work

Here we have continued the development of tools to manipulate completions of wqos and we have applied these tools together with new ideas to deduce the following decidabilities: strong termination for ω^2 -WSTS with transitive monotony and post-effective completion, boundedness for post-effective WSTS with strict transitive monotony and with well partial ordering, strong control-state maintainability for ω^2 -WSTS with strong monotony and a post-effective completion and finally, coverability for WSTS having a post-effective completion.

Future work should apply these decidabilities to parameterized WSTS and should investigate algorithmic aspects of these decidabilities, including a comparison of the relative efficiencies of backward and forward strategies.

Acknowledgements. We thank the referees for helpful comments and pointers.

References

1. Abdulla, P.A., Cerans, K., Jonsson, B., Tsay, Y.K.: General decidability theorems for infinite-state systems. In: LICS, pp. 313–321 (1996)
2. Abdulla, P.A., Cerans, K., Jonsson, B., Tsay, Y.K.: Algorithmic analysis of programs with well quasi-ordered domains. *Inf. Comput.* 160(1-2), 109–127 (2000)
3. Abramsky, S., Jung, A.: Domain theory. In: *Handbook of Logic in Comp. Sci.*, vol. 3, pp. 1–168. Oxford University Press (1994)

4. Bertrand, N., Delzanno, G., König, B., Sangnier, A., Stückrath, J.: On the decidability status of reachability and coverability in graph transformation systems. In: RTA, pp. 101–116 (2012)
5. Bertrand, N., Schnoebelen, P.: Computable fixpoints in well-structured symbolic model checking. *Formal Methods in System Design* 43(2), 233–267 (2013)
6. Bouyer, P., Markey, N., Ouaknine, J., Schnoebelen, P., Worrell, J.: On termination and invariance for faulty channel systems. *FAC* 24(4-6), 595–607 (2012)
7. Cécé, G., Finkel, A., Iyer, S.P.: Unreliable channels are easier to verify than perfect channels. *Inf. Comput.* 124(1), 20–31 (1996)
8. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset nets between decidability and undecidability. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) *ICALP 1998*. LNCS, vol. 1443, pp. 103–115. Springer, Heidelberg (1998)
9. Dufourd, C., Jančar, P., Schnoebelen, P.: Boundedness of reset P/T nets. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) *ICALP 1999*. LNCS, vol. 1644, pp. 301–310. Springer, Heidelberg (1999)
10. Esparza, J., Finkel, A., Mayr, R.: On the verification of broadcast protocols. In: *LICS*, pp. 352–359 (1999)
11. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! *Theoret. Comput. Sci.* 256(1–2), 63–92 (2001)
12. Finkel, A.: Reduction and covering of infinite reachability trees. *Information and Computation* 89(2), 144–179 (1990)
13. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, part I: Completions. In: *STACS*, pp. 433–444 (2009)
14. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, Part II: Complete WSTS. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikolettseas, S., Thomas, W. (eds.) *ICALP 2009, Part II*. LNCS, vol. 5556, pp. 188–199. Springer, Heidelberg (2009)
15. Finkel, A., McKenzie, P., Pícarony, C.: A well-structured framework for analysing Petri net extensions. *Information and Computation* 195(1-2), 1–29 (2004)
16. Ganty, P., Raskin, J.-F., Van Begin, L.: A complete abstract interpretation framework for coverability properties of WSTS. In: Emerson, E.A., Namjoshi, K.S. (eds.) *VMCAI 2006*. LNCS, vol. 3855, pp. 49–64. Springer, Heidelberg (2006)
17. Geeraerts, G., Heußner, A., Praveen, M., Raskin, J.F.: ω -Petri nets. In: *Petri Nets*, pp. 49–69 (2013)
18. Geeraerts, G., Raskin, J.F., Begin, L.V.: Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *JCSS* 72(1), 180–203 (2006)
19. Goubault-Larrecq, J., Schnoebelen, P.: Personal communication (October 2013)
20. Jancar, P.: A note on well quasi-orderings for powersets. *Inf. Process. Lett.* 72(5-6), 155–160 (1999)
21. Kaiser, A., Kroening, D., Wahl, T.: Efficient coverability analysis by proof minimization. In: Koutny, M., Ulidowski, I. (eds.) *CONCUR 2012*. LNCS, vol. 7454, pp. 500–515. Springer, Heidelberg (2012)
22. König, B., Stückrath, J.: Well-structured graph transformation systems with negative application conditions. In: Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G. (eds.) *ICGT 2012*. LNCS, vol. 7562, pp. 81–95. Springer, Heidelberg (2012)
23. Pahl, J.K.: Reachability problems for communicating finite state machines. Technical Report CS-82-12, University of Waterloo (1982)
24. Schmitz, S., Schnoebelen, P.: Multiply-recursive upper bounds with Higman’s lemma. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) *ICALP 2011, Part II*. LNCS, vol. 6756, pp. 441–452. Springer, Heidelberg (2011)

25. Schmitz, S., Schnoebelen, P.: The power of well-structured systems. In: D'Argenio, P.R., Melgratti, H. (eds.) CONCUR 2013 – Concurrency Theory. LNCS, vol. 8052, pp. 5–24. Springer, Heidelberg (2013)
26. Wies, T., Zufferey, D., Henzinger, T.A.: Forward analysis of depth-bounded processes. In: Ong, L. (ed.) FOSSACS 2010. LNCS, vol. 6014, pp. 94–108. Springer, Heidelberg (2010)
27. Zufferey, D., Wies, T., Henzinger, T.A.: Ideal abstractions for well-structured transition systems. In: Kuncak, V., Rybalchenko, A. (eds.) VMCAI 2012. LNCS, vol. 7148, pp. 445–460. Springer, Heidelberg (2012)