Handling Infinitely Branching Well-structured Transition Systems☆

Michael Blondin[a,b,1,2], Alain Finkel[b,3], Pierre McKenzie[a,b,2,4]

[a] *LITQ, DIRO, Université de Montréal, C.P. 6128 succursale Centre-ville, Montréal (Québec), H3C 3J7, Canada*
[b] *LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, 61 avenue du Président Wilson, 94235 Cachan Cedex, France*

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 maintainability and the coverability problems. Coverability and boundedness under new effectiveness conditions are shown decidable.

Keywords: Well-structured transition systems, infinite branching, completion, decidability, coverability, termination

1. Introduction

Well-structured transition systems (WSTS) [1, 2, 3] as a general class of infinite-state systems have spawned decidability results for important problems such as termination, boundedness, 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

☆Extended and expanded version of "Handling Infinitely Branching WSTS" by M. Blondin, A. Finkel and P. McKenzie, in Proc. 41st International Colloquium on Automata, Languages, and Programming (ICALP), 2014.

Email addresses: `blondimi@iro.umontreal.ca` (Michael Blondin), `finkel@lsv.fr` (Alain Finkel), `mckenzie@iro.umontreal.ca` (Pierre McKenzie)

[1]Supported by the Fonds de recherche du Québec – Nature et technologies (FRQNT) and by the French Centre national de la recherche scientifique (CNRS).

[2]Supported by the "Chaire Digiteo, ENS Cachan - École Polytechnique (France)"

[3]Supported by the French Agence nationale de la recherche, ReacHard (grant ANR-11-BS02-001).

[4]Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

status of reachability and coverability for graph transformation systems [4, 5], in the forward analysis of depth-bounded processes [6, 7], in the verification of parameterized protocols [8] and the verification of multi-threaded asynchronous software [9]. WSTS remain under development and are actively being investigated [10, 11, 12, 13, 14, 15].

Most existing decidability results for WSTS apply to the *finitely branching* variant. However, there are many intrinsically *infinitely branching* WSTS. Let us cite inserting FIFO automata [16] which are able to insert any word at any time in a FIFO buffer, inserting automata [17], recursive-parallel systems [18] and ω-Petri nets [19]. Moreover, any finitely branching WSTS parameterized with an infinite set of initial states (such as broadcast protocols [8]) also inherits an infinitely branching state. For instance, Geeraerts, Heußner, Praveen and Raskin argue in [19] that parametric concurrent systems with dynamic thread creation can naturally be modeled by some classes of infinitely branching systems, like ω-Petri nets, *i.e.* Petri net with arcs that can consume/create arbitrarily many tokens.

The primary motivation for this paper is to explore the decidability status of the termination, boundedness, maintainability and coverability problems for infinitely branching (general) WSTS. For the coverability problem, known to be decidable for WSTS fulfilling upward pre-effectiveness [3] (which roughly 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), we wish to draw from the recent algebra-theoretic characterizations of downward-closed sets [10] 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 pre-oriented computability. 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 infinitely branching WSTS. 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 undecidabilty that results from dropping one of these conditions. The new decidability results for (strong) termination and (weak) maintainability exploit the completion. 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 new algorithm for coverability uses a forward strategy coupled with a post-oriented computability hypothesis.

Section 2 below fixes the notation pertaining to orderings and transition systems. Section 3 recalls the notion of WSTS, gives examples, discusses branching and effectiveness, defines the computation problems at issue and adds two undecidability results concerning finitely branching WSTS. Section 4 develops tools to handle infinitely branching WSTS and forms the theoretical backbone of our paper. Section 5 contains our decidability results for infinitely branching WSTS. Section 6 summarizes and suggests future work.

2. Preliminaries

2.1. Orderings

Let X be a set and $\leq \subseteq X \times X$. We say that $\leq$ is a *quasi-ordering* (*qo*) for X if it is reflexive and transitive. If $\leq$ is also antisymmetric, then it is a *partial ordering* (*po*). A quasi-ordering (resp. partial ordering) $\leq$ is said to be a *well-quasi-ordering* (resp. *well partial ordering*), abbreviated *wqo* (resp. *wpo*), if for every infinite sequence $x_0, x_1, \ldots$ 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 partially ordered under $\leq_{\mathbb{N}^d}$ defined by

$$(x_1, x_2, \ldots, x_d) \leq_{\mathbb{N}^d} (x'_1, x'_2, \ldots, x'_d) \iff \forall i \in \{1, 2, \ldots, d\} \;\; x_i \leq x'_i \;.$$

In this work, we extend $\mathbb{N}$ to $\mathbb{N}_\omega \stackrel{\text{def}}{=} \mathbb{N} \cup \{\omega\}$ and we extend $\leq_{\mathbb{N}}$ to $\leq_{\mathbb{N}_\omega}$ with $x \leq_{\mathbb{N}_\omega} \omega$ for all $x \in \mathbb{N}_\omega$. The quasi-ordering $\leq_{\mathbb{N}_\omega}$ is also a wpo and is naturally extended to the wpo $\leq_{\mathbb{N}^d_\omega}$ over $\mathbb{N}^d_\omega$. We will simply write $\leq$ for $\leq_{\mathbb{N}}$, $\leq_{\mathbb{N}_\omega}$, $\leq_{\mathbb{N}^d}$ and $\leq_{\mathbb{N}^d_\omega}$ when there is no ambiguity. We also write $x < y$ whenever $x \leq y$ and $\neg(y \leq x)$. In some examples, we will also consider the subword ordering denoted $\preceq$. For every finite alphabet Σ and $u, v \in \Sigma^*$, $u \preceq v$ if, and only if, $u = v$ or u can be obtained from v by removing some letters. It is well-known that $\preceq$ is a wqo.

Let $T \subseteq X$. We define the *upward closure* of T as $\uparrow T \stackrel{\text{def}}{=} \{x \in X : y \leq x \text{ for some } y \in T\}$ and the *downward closure* of T as $\downarrow T \stackrel{\text{def}}{=} \{x \in X : x \leq y \text{ for some } y \in T\}$. We say that T is *upward closed* if $T = \uparrow T$ and *downward closed* if $T = \downarrow T$. Let $x \in X$, we simply write $\uparrow x$ for $\uparrow \{x\}$, and $\downarrow x$ for $\downarrow \{x\}$. An *(upward) basis* of an upward closed set T is a set B such that $T = \uparrow B$. It is known that every upward closed subset of a well-quasi-ordered set has a minimal finite basis. An *ideal* I is a downward closed subset of X that is also *directed*, *i.e.*, nonempty and such that $\forall a, b \in I, \exists c \in I$ such that $a \leq c$ and

$b \leq c$. We define $\mathsf{Ideals}(X)$ as the set of ideals of X, *i.e.*, $\mathsf{Ideals}(X) \stackrel{\text{def}}{=} \{\emptyset \subset I \subseteq X : I = \downarrow I \text{ and } I \text{ is directed}\}$.

2.2. Transition systems

A *transition system* is a pair $S = (X, \to)$ where X is a set, called the set of *states*, and $\to$ is a relation $\to \subseteq X \times X$, called the *transition relation*. If a transition system S is also equipped with a quasi-ordering $\leq$ on X, we say that $S = (X, \to, \leq)$ is an *ordered transition system*. We let

$$\xrightarrow{k} \stackrel{\text{def}}{=} \underbrace{\to \circ \to \circ \cdots \circ \to}_{k \text{ times}}, \quad \xrightarrow{+} \stackrel{\text{def}}{=} \bigcup_{k \geq 1} \xrightarrow{k} \quad \text{and} \quad \xrightarrow{*} \stackrel{\text{def}}{=} \xrightarrow{0} \cup \xrightarrow{+} \ .$$

In other words, $\xrightarrow{+}$ is the transitive closure of $\to$, and $\xrightarrow{*}$ is the reflexive and transitive closure of $\to$. We define, respectively, the set of *immediate predecessors* and *immediate successors* of $x \in X$ by $\mathrm{Pre}_S(x) \stackrel{\text{def}}{=} \{y : y \to x\}$ and $\mathrm{Post}_S(x) \stackrel{\text{def}}{=} \{y : x \to y\}$. These sets are extended naturally for $T \subseteq X$ by $\mathrm{Pre}_S(T) \stackrel{\text{def}}{=} \bigcup_{x \in T} \mathrm{Pre}_S(x)$ and $\mathrm{Post}_S(T) \stackrel{\text{def}}{=} \bigcup_{x \in T} \mathrm{Post}_S(x)$. We also define, respectively, the set of *predecessors* and *successors* of $x \in X$ by $\mathrm{Pre}^*_S(x) \stackrel{\text{def}}{=} \{y : y \xrightarrow{*} x\}$ and $\mathrm{Post}^*_S(x) \stackrel{\text{def}}{=} \{y : x \xrightarrow{*} y\}$. These sets are also extended naturally for $T \subseteq X$ by $\mathrm{Pre}^*_S(T) \stackrel{\text{def}}{=} \bigcup_{x \in T} \mathrm{Pre}^*_S(x)$ and $\mathrm{Post}^*_S(T) \stackrel{\text{def}}{=} \bigcup_{x \in T} \mathrm{Post}^*_S(x)$. Let $x, y \in X$ and $T \subseteq X$, we say that $x \to_T y$ if $x, y \in T$. We define $\xrightarrow{k}_T, \xrightarrow{+}_T$ and $\xrightarrow{*}_T$ similarly in the natural way.

2.3. Undecidability

We denote by Turing_i the i^{th} Turing machine in a classical enumeration. We assume that the problem of testing, on input $i \in \mathbb{N}$, whether Turing_i halts on its encoding is undecidable. We will use this fact throughout this work in order to show that some problems are undecidable.

3. Well-structured transition systems (WSTS)

In this section we introduce well-structured transition systems and the different problems we will study in the following sections.

Definition 1. A *well-structured transition system* (*WSTS*) is an ordered transition system $S = (X, \to, \leq)$ such that $\leq$ is a wqo, and the relation $\to$ is *monotone* (or *compatible*) with $\leq$, *i.e.*,

$$\forall x, x', y \in X \quad x \leq x' \text{ and } x \to y \implies \exists y' \in X \text{ s.t. } y \leq y' \text{ and } x' \xrightarrow{*} y' \ . \quad (1)$$

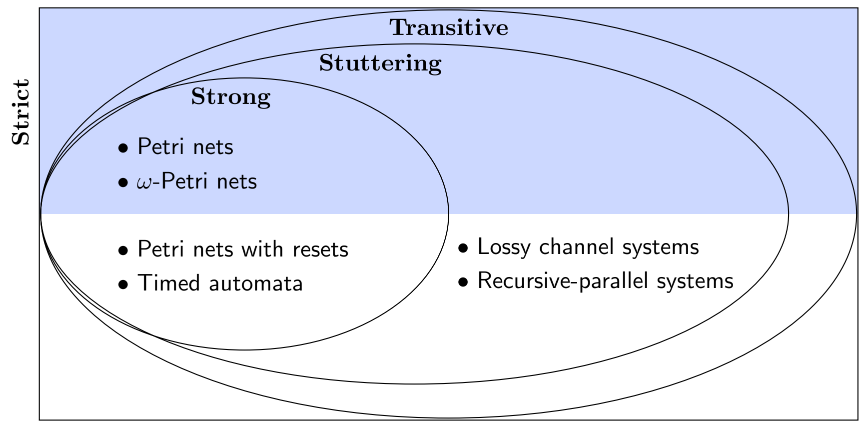

Figure 1: WSTS monotonicity inclusions with examples.

3.1. Monotonicity

There exist many other variations of monotonicity obtained by modifying (1). In this work we will consider the following ones:

$$\begin{array}{llll}
\textit{strong:} & x \leq x' \text{ and } x \to y \implies \exists y' & \text{s.t. } y \leq y' \text{ and } x' \to y' \\
\textit{stuttering:} & x \leq x' \text{ and } x \to y \implies \exists x'', y' & \text{s.t. } y \leq y' \text{ and } x' \to^*_{\uparrow x} x'' \to y' \\
\textit{transitive:} & x \leq x' \text{ and } x \to y \implies \exists y' & \text{s.t. } y \leq y' \text{ and } x' \to^+ y' \, .
\end{array}$$

When (1) or any of the above conditions also holds with $<$, we say that the monotonicity is additionally *strict*. For example, transitive and strict monotonicity entails both:

$$x \leq x' \text{ and } x \to y \implies \exists y' \text{ s.t. } y \leq y' \text{ and } x' \to^+ y'$$
$$x < x' \text{ and } x \to y \implies \exists y' \text{ s.t. } y < y' \text{ and } x' \to^+ y' \, .$$

Note that strong monotonicity implies stuttering monotonicity which implies transitive monotonicity. A classification of the different types of monotonicity, with some examples drawn from the literature, is illustrated in Fig. 1. See [2, Fig. 9] for a more detailed classification. Note that among the 22 WSTS models in [2, Fig. 9], there is no WSTS having a transitive but non-stuttering monotonicity.

3.2. Functional WSTS and concrete examples of WSTS

We define a generic class of WSTS that will be used in some parts of the paper. Recall that a partial function $f : X \to X$ is *non decreasing* if, for every $x \in X$, if $f(x)$ is defined then $f(y)$ is defined for every $y \geq x$, and $x \leq y \implies f(x) \leq f(y)$.

Definition 2 ([11]). A *functional WSTS* is an ordered transition system $S = (X, \to, \leq)$ such that $\leq$ is a wqo for X, and $\to$ is defined by a finite set F of non decreasing partial functions, *i.e.*, $x \to y$ if $f(x) = y$ for some $f \in F$.

To see that functional WSTS are indeed WSTS, let us show that they have strong monotonicity. Let $x, y, x' \in X$ be such that $x \leq x'$ and $x \to y$, then $f(x) = y$ for some $f \in F$. Hence, since f is non decreasing, $f(x')$ is defined and thus $y = f(x) \leq f(x') = y'$ for some $y' \in X$. Therefore, $x' \to y'$ and $y \leq y'$. Note that in general the monotonicity is not necessarily strict, *e.g.* when the functions are constant.

For completeness, we also give some concrete examples of WSTS drawn from the literature.

Example 3 (Affine nets). A d-dimensional *affine net* [20] is a functional WSTS $S = (\mathbb{N}^d, \to, \leq)$ where $\to$ is defined by a finite set F of affine functions:

- each $f \in F$ is an affine function given by some $A \in \mathbb{N}^{d \times d}$ and $b \in \mathbb{Z}^d$, *i.e.*, $f(x) = Ax + b$, and
- $x \to y$ if $f(x) = y \in \mathbb{N}^d$ for some $f \in F$.

Affine nets encompass Petri nets and most of their extensions (with resets, transfers, etc.).

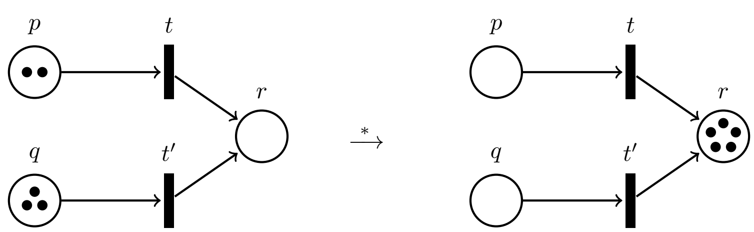

Figure 2: Left: example of a Petri net marked (counterclockwise) with $(2, 3, 0)$. Right: marking obtained, *i.e.* $(0, 0, 5)$, after firing t twice and t' three times.

Example 4 (Petri nets). A *Petri net* is an affine net such that each of its affine functions is of the form $f(x) = Ix + b$ where I is the identity matrix. Petri nets are often defined equivalently with the notion of places, transitions and markings. In the formalism of d-dimensional affine nets, the number of *places* is d, the *transitions* correspond to the affine functions, and *markings* are vectors $x \in \mathbb{N}^d$. Petri nets are WSTS with strong and strict monotonicity.

For example, consider the Petri net $S = (\mathbb{N}^3, \to, \leq)$ defined by

$$f_t(x_p, x_q, x_r) = I \begin{pmatrix} x_p \\ x_q \\ x_r \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

$$f_{t'}(x_p, x_q, x_r) = I \begin{pmatrix} x_p \\ x_q \\ x_r \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}.$$

S is illustrated on the left side of Fig. 2 with the initial marking $(2, 3, 0)$. Its places $\{p, q, r\}$ correspond to the three components of markings, and its transitions $\{t, t'\}$ correspond to the functions f_t and $f_{t'}$. This Petri net "weakly computes" the sum of p and q into r, *i.e.* $(0, 0, x_p + x_q)$ can be reached from $(x_p, x_q, 0)$, and $x'_r \leq x_p + x_q$ for every reachable marking (x'_p, x'_q, x'_r). For example, $(0, 0, 5)$ may be reached from $(2, 3, 0)$ as illustrated in Fig. 2.

Example 5 (ω-Petri nets). An *ω-Petri net* [19] is an extended Petri net in which arcs can be labelled with positive integers or with ω. The semantics of an ω-output arc from transition t to place p is that transition t can create nondeterministically an unbounded albeit finite number of tokens in p; hence from a marking x in which t may be fired, there are infinitely many reachable markings y such that $y_p \in x_p + \mathbb{N}$. The semantics of an ω-input arc from a place p to a transition t is that transition t can consume any positive amount of tokens from p. ω-Petri nets are WSTS with strong and strict monotonicity.

For example, the ω-Petri net illustrated on the left side of Fig. 3 is initially marked with $(3, 5, 0)$ and a single firing of its unique transition may lead to $(1, 1, 3)$.

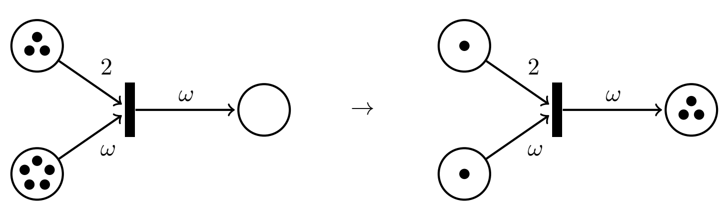

Figure 3: Left: example of an ω-Petri net marked (counterclockwise) with $(3, 5, 0)$. Right: example of a possible marking, *i.e.* $(1, 1, 3)$, obtained after firing the unique transition.

Example 6 (Lossy channel systems). A *channel system* [21, 22] is a finite state automaton (Q, Σ, δ) equipped with finitely many FIFO channels. The transitions of a channel system are labelled either with $c_i!\sigma$ or $c_i?\sigma$. A transition labelled $c_i!\sigma$ adds the letter σ to channel i. A transition labelled $c_i?\sigma$ may be taken if the first letter of channel i is σ, in which case it is consumed. A channel system is said to be a *lossy channel system* [23] when the system is additionally allowed to lose any letter of any channel in any configuration. For example, consider the lossy channel system illustrated in Fig. 4 with $Q = \{p, q\}, \Sigma = \{a, b\}$ and channels c_1, c_2. From the initial configuration $(p, c_1 = \varepsilon, c_2 = \varepsilon)$, the configuration $(p, c_1 = b, c_2 = a)$ can be reached by executing $t_1\, t_3\, t_3$, losing the first letter of c_1, and executing $t_2\, t_4$.

A lossy channel system with d channels is a WSTS $(Q \times (\Sigma^*)^d, \to, \leq)$ where $(p, w_1, w_2, \ldots, w_d) \leq (p', w_1', w_2', \ldots, w_d')$ if, and only if,

$$(p = q) \wedge (w_1 \preceq w_1') \wedge (w_2 \preceq w_2') \wedge \cdots \wedge (w_d \preceq w_d').$$

Lossy channel systems have stuttering monotonicity [2].

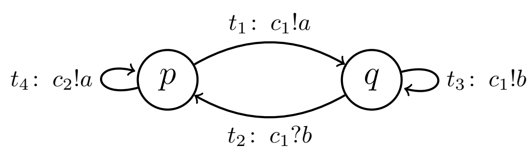

Figure 4: Example of a lossy channel system with two channels.

3.3. Branching

In the literature, $\mathrm{Post}_S(x)$ is usually assumed to be finite, this will not be the case in this paper and therefore we need to introduce the notion of branching.

Definition 7. A WSTS $S = (X, \to, \leq)$ is *finitely branching* if $\mathrm{Post}_S(x)$ is finite for every $x \in X$. Otherwise, S is said to be *infinitely branching*.

Example 8. Most well-known transition systems such as affine nets, Petri nets and lossy channel systems (see Sect. 3.2) are finitely branching. However, some WSTS, such as inserting FIFO automata [16], inserting automata [17], ω-Petri nets (see Example 5) and broadcast protocols [8], are not finitely branching. More generally, parameterized transition systems (*i.e.* with an infinite set of initial states) are naturally infinitely branching.

Even though we will not consider it in this paper, it is worth mentioning that a subclass of infinitely branching WSTS has been studied in [3]. They define a WSTS to be *essentially finitely branching* if $\mathrm{Post}_S(x)$ has a finite number of maximal elements for every $x \in X$.

3.4. Encoding, classes and effectiveness

The point of view taken in this paper is upstream from computational complexity issues. We introduce classes of WSTS in order to parametrize computational problems.

Definition 9. A *class $\mathcal{C}$ of WSTS* is any countable set of WSTS. We denote the i^{th} WSTS of $\mathcal{C}$ by $\mathcal{C}(i)$.

Classes of WSTS in this generality will only serve to parametrize computational problems. For the purpose of manipulating WSTS, we extend Finkel & Goubault-Larrecq [11] and require for any class $\mathcal{C}$ of WSTS that a set $E_{\mathcal{C}} \subseteq \mathbb{N}$ and a surjective *representation map* $r : E_{\mathcal{C}} \to \bigcup_i X_i$ be understood where X_i is the set of states of $\mathcal{C}(i)$. Let $E_{X_i} = \{e \in E_{\mathcal{C}} : r(e) \in X_i\}$, we further require the set $\{(i, e) : i \in \mathbb{N}, e \in E_{X_i}\}$ to be decidable. Any $e \in E_{X_i}$ such that $r(e) = x$ is called an *encoding* of $x \in X_i$. A Turing machine M over $\mathbb{N} \times \mathbb{N}$ is said to *compute* a relation $\rho \subseteq X_i \times X_i$ if M halts at least on $E_{X_i} \times E_{X_i}$ and for each $e, e' \in E_{X_i}$, M accepts (e, e') if, and only if, $(r(e), r(e')) \in \rho$.

Throughout this paper, WSTS will be assumed effective in the following sense:

Definition 10. A class $\mathcal{C}$ of WSTS is *effective* if a pair of Turing machines $(M_{\to}, M_{\leq})$ operating on $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ exists such that, for each $i \in \mathbb{N}$, $M_{\to}$ with first argument set to i computes the transition relation "$\to$" of $\mathcal{C}(i)$ and $M_{\leq}$ with first argument set to i computes the ordering relation "$\leq$" of $\mathcal{C}(i)$. By extension, we say that a WSTS S is *effective* if the degenerate class $\{S\}$ is effective.

Most well-known classes of WSTS are effective, *e.g.*, Petri nets and their usual extensions with reset/transfer transitions or with ω-transitions, FIFO automata with perfect or lossy channels. Note the subtle, yet crucial, "uniformity" distinction between effective classes of WSTS and classes of WSTS such that each of their WSTS is effective:

Proposition 11. *There exists a class $\mathcal{C}$ of WSTS such that each $S \in \mathcal{C}$ is effective, yet $\mathcal{C}$ is not effective.*

Proof. Let $i \in \mathbb{N}$ and let $S_i = (\{0\}, \to, \{(0, 0)\})$ be such that

$$\to \stackrel{\text{def}}{=} \begin{cases} \{(0,0)\} & \text{if } \mathsf{Turing}_i \text{ halts on its encoding,} \\ \emptyset & \text{otherwise.} \end{cases}$$

Let $\mathcal{C} = \{S_i : i \in \mathbb{N}\}$. S_i is an effective WSTS since there exists a Turing machine that computes $\to$, *i.e.* either the machine accepting every input or the machine refusing every input.

Suppose that $\mathcal{C}$ is an effective class of WSTS, then we can decide whether Turing_i halts on its encoding by verifying whether $0 \to 0$ in S_i. This is a contradiction, hence $\mathcal{C}$ is not effective. □

In the literature, $\text{Post}_S(x)$ is usually assumed to be finite and most analysis techniques compute the finite set $\text{Post}_S(x)$, which is made possible by assuming Post_S to be computable. Because our setting allows $\text{Post}_S(x)$ to be infinite, we need to adapt this assumption.

Definition 12. A class $\mathcal{C}$ of WSTS is said to be *post-effective* if it is effective, and if there exists a Turing machine $M_{|\text{Post}|}$ that computes $|\text{Post}_{\mathcal{C}(i)}(x)| \in \mathbb{N} \cup$

$\{\infty\}$ on input (i, x), with $i \in \mathbb{N}$ and x a state[5] of $\mathcal{C}(i)$. By extension, we say that a WSTS S is *post-effective* if the degenerate class $\{S\}$ is post-effective.

We note that even when a class of WSTS is post-effective, it is undecidable to test whether its WSTS are finitely branching or not.

Proposition 13. *There exists a post-effective class $\mathcal{C}$ of WSTS with strong and strict monotonicity for which testing, on input $i \in \mathbb{N}$, whether $\mathcal{C}(i)$ is finitely branching is undecidable.*

PROOF. Let $i \in \mathbb{N}$ and let $S_i = (\mathbb{N}, \rightarrow, \leq)$ be such that

$$\begin{array}{ll} x \rightarrow x & \text{if } \mathsf{Turing}_i \text{ does not halt on its encoding in } x \text{ steps or less,} \\ x \rightarrow x, x+1, \ldots & \text{otherwise.} \end{array}$$

Let $\mathcal{C} = \{S_i : i \in \mathbb{N}\}$. Note that S_i has strong and strict monotonicity, and both $x \rightarrow y$ and $|\mathrm{Post}_{S_i}(x)|$ can be computed by executing Turing_i for a finite amount of steps. Moreover, Turing_i halts on its encoding if, and only if, there exists $x \in X$ such that $\mathrm{Post}_{S_i}(x)$ is infinite. $\square$

Even though our work does not rely on upward pre-effectiveness, which was already mentioned in the introduction, we will compare it with our effectiveness hypotheses. Therefore, we define it formally.

Definition 14 ([3]). A class $\mathcal{C}$ of WSTS is said to be *upward pre-effective* if there exists a Turing machine M_{prebasis} that computes the minimal basis of $\uparrow \mathrm{Pre}_{\mathcal{C}(i)}(\uparrow x)$ on input (i, x), with $i \in \mathbb{N}$ and x a state of $\mathcal{C}(i)$. By extension, we say that a WSTS S is *upward pre-effective* if the degenerate class $\{S\}$ is upward pre-effective.

3.5. Decision problems

In this section, we formally define the decision problems considered throughout this paper. Let $\mathcal{C}$ be a class of WSTS, we parametrize these problems by $\mathcal{C}$, *i.e.*, the input WSTS are given by their index $i \in \mathbb{N}$ in $\mathcal{C}$.

TERMINATION$_{\mathcal{C}}$

INPUT: A WSTS $S = (X, \rightarrow, \leq) \in \mathcal{C}$ and $x_0 \in X$.
QUESTION: Is it the case that no infinite sequence $x_0, x_1, \ldots$ such that $x_0 \rightarrow x_1 \rightarrow \ldots$ exists?

STRONG TERMINATION$_{\mathcal{C}}$

INPUT: A WSTS $S = (X, \rightarrow, \leq) \in \mathcal{C}$ and $x_0 \in X$.
QUESTION: $\exists m \in \mathbb{N}$ such that $x_0 \rightarrow x_1 \rightarrow \ldots \rightarrow x_k \implies k \leq m$?

[5] As in Definition 10, manipulating the elements of the set of states X_i of $\mathcal{C}(i)$ is done via E_{X_i}, with $M_{|\mathrm{Post}|}$ required to halt at least on $\mathbb{N} \times E_{X_i}$.

<u>Boundedness$_\mathcal{C}$</u>

INPUT: A WSTS $S = (X, \to, \leq) \in \mathcal{C}$ and $x_0 \in X$.

QUESTION: Is $\mathrm{Post}_S^*(x_0)$ finite?

<u>Maintainability$_\mathcal{C}$</u>

INPUT: A WSTS $S = (X, \to, \leq) \in \mathcal{C}$, $x_0 \in X$ and $t_1, t_2, \ldots, t_n \in X$.

QUESTION: Let $T = \uparrow\{t_1, t_2, \ldots, t_n\}$. Is there a maximal execution from x_0 staying in T? Namely, is there an infinite sequence $x_0, x_1, \ldots$ such that $x_0 \to_T x_1 \to_T \ldots$, or a finite sequence $x_0, x_1, \ldots, x_k$ such that $x_0 \to_T x_1 \to_T \ldots \to_T x_k$ and $\mathrm{Post}_S(x_k) = \emptyset$?

<u>Weak maintainability$_\mathcal{C}$</u>

INPUT: A WSTS $S = (X, \to, \leq) \in \mathcal{C}$, $x_0 \in X$ and $t_1, t_2, \ldots, t_n \in X$.

QUESTION: Let $T = \uparrow\{t_1, t_2, \ldots, t_n\}$. For every $m \in \mathbb{N}$, is there a finite sequence $x_0, x_1, \ldots, x_k$ such that $x_0 \to_T x_1 \to_T \ldots \to_T x_k$ and $k \geq m$?

<u>Coverability$_\mathcal{C}$</u>

INPUT: A WSTS $S = (X, \to, \leq) \in \mathcal{C}$ and $x_0, x \in X$.

QUESTION: $\exists x' \in X$ such that $x \leq x'$ and $x_0 \to^* x'$?

3.6. Decidability status of the decision problems for finitely branching WSTS

We summarize the four main decidability results known about finitely branching WSTS. Theorem 15 recalls results exactly as they appear in the literature except for two new results, proven in Prop. 16 and Prop. 17, about the undecidability of termination and maintainability.

Theorem 15 ([1, 2, 3, 24, 20]). *The following holds, where classes $\mathcal{C}$ and $\mathcal{D}$ are assumed to be post-effective classes of finitely branching WSTS:*

- *Termination is decidable for any class $\mathcal{C}$ with transitive monotonicity [1, 2], and there exists a class $\mathcal{D}$ of WSTS with well partial ordering for which the problem is undecidable (see Prop. 16).*
- *Boundedness is decidable for any class $\mathcal{C}$ with strict monotonicity and well partial ordering [1, 2], and there exists a class $\mathcal{D}$ of WSTS with strong monotonicity and well partial ordering for which the problem is undecidable [24].*
- *Maintainability is decidable for any class $\mathcal{C}$ with stuttering monotonicity [2], and there exists a class $\mathcal{D}$ of WSTS with well partial ordering for which the problem is undecidable (see Prop. 17).*

The following holds, where classes $\mathcal{C}$ and $\mathcal{D}$ are assumed to be effective classes of WSTS:

- *Coverability is decidable for any upward pre-effective class $\mathcal{C}$ [2, 3], and there exists a post-effective class $\mathcal{D}$ of finitely branching WSTS with strong and strict monotonicity for which the problem is undecidable [20].*

Proposition 16. *There exists a post-effective class of finitely branching WSTS for which termination is undecidable.*

Proof. Let $S_i = (\mathbb{N}, \to, \leq)$ be the transition system such that $x \to x+1$ if Turing_i does not halt on its encoding in x steps or less. Let $\mathcal{C} = \{S_i : i \in N\}$.

Let us show that $\mathcal{C}$ is a post-effective class of finitely branching WSTS. It is clear that S_i is finitely branching. Moreover, $\mathcal{C}$ is post-effective because $\to$ and $|\mathrm{Post}_{S_i}(x)|$ can be computed by executing Turing_i for a finite amount of steps. Since $\leq$ is a wqo, it only remains to show that S is monotone. Let $x, x', y \in \mathbb{N}$ be such that $x \leq x'$ and $x \to y$. By definition, we have $y = x+1$. If $x = x'$, then $x' \to x+1$ and we are done since $y \leq x+1$. If $x < x'$, then trivially $x' \to^* x'$ and we are also done since $y = x+1 \leq x'$.

Now, we note that for each i there exists an infinite sequence $x_0, x_1, \ldots$ such that $x_0 = 0$ and $x_0 \to x_1 \to \ldots$ if, and only if, Turing_i does not halt on its encoding. Hence, termination for $\mathcal{C}$ is undecidable. □

Proposition 17. *There exists a post-effective class of finitely branching WSTS for which maintainability is undecidable.*

Proof. Let $S_i = (\mathbb{N}, \to, \leq)$ be the transition system such that

- $x \to x+1$ if $x \geq 1$ and Turing_i does not halt on its encoding in x steps or less,
- $x \to 0$ if $x \geq 1$ and Turing_i halts on its encoding in x steps or less.

Let $\mathcal{C} = \{S_i : i \in N\}$. With a similar argument as in the proof of Prop. 16, we can show that $\mathcal{C}$ forms a post-effective class of finitely branching WSTS.

Let $x_0 = 1$ and $T = \uparrow 1$. Suppose that, for any i, there exists a finite sequence $x_0, x_1, \ldots, x_k$ such that $x_0 \to_T x_1 \to_T \ldots \to_T x_k$ in S_i and $\mathrm{Post}_{S_i}(x_k) = \emptyset$, then by definition $x_k = 0$, hence there is a contradiction since $0 \notin T$. Moreover, there exists an infinite sequence $x_0, x_1, \ldots$ such that $x_0 \to_T x_1 \to_T \ldots$ in S_i if, and only if, Turing_i does not halt. Hence, maintainability for $\mathcal{C}$ is undecidable. □

4. Handling infinitely branching WSTS finitely

In this section we develop the theory that will support the design in Section 5 of procedures capable of handling, under natural hypotheses, infinitely branching systems.

We begin by discussing downward-closed sets and their ideals. Then we revisit the so-called WSTS completion in the general setting of infinitely branching WSTS. We will see that completing a WSTS yields a system that is finitely branching, but at the expense of blurring some, though fortunately not all, of the original WSTS behaviors. Furthermore, we will see that computing the completion of a WSTS requires computing finite representations of each term in the sequence

$$\downarrow x, \downarrow \mathrm{Post}(\downarrow x), \downarrow \mathrm{Post}(\downarrow \mathrm{Post}(\downarrow x)), \ldots$$

which requires the ability to represent downward closed sets finitely. Concrete examples are scattered throughout this section, with a final subsection devoted to examples of completions.

4.1. Downward closed sets and ideals

It has long been known that in a well-quasi-ordered set, any upward closed subset has a finite basis; this is Dickson's lemma in $(\mathbb{N}^k, \leq)$ and it is Higman's lemma in $(\Sigma^*, \preceq)$. It has recently been (re)discovered that a similar situation occurs for well-quasi-ordered downward closed sets. We will see this in Theorem 21, but first we give two examples of downward closed sets and observe that they can be represented by finitely many ideals.

Example 18. Let us consider the ideals of $\mathbb{N}^d$. It can be shown that

$$\mathsf{Ideals}(\mathbb{N}^d) = \underbrace{\mathsf{Ideals}(\mathbb{N}) \times \mathsf{Ideals}(\mathbb{N}) \times \cdots \times \mathsf{Ideals}(\mathbb{N})}_{d \text{ times}}$$

and that $I \in \mathsf{Ideals}(\mathbb{N})$ is either $\mathbb{N}$ or of the form $\downarrow x$ for some $x \in \mathbb{N}$. Therefore, any ideal $I \in \mathsf{Ideals}(\mathbb{N}^d)$ may be represented by some $x \in \mathbb{N}_\omega^d$ where $x_i = \omega$ represents $\mathbb{N}$ and $x_i = y$ represents $\downarrow y$. Consider the following downward closed set

$$X = \{(x_1, x_2) \in \mathbb{N}^2 : (x_1 \leq 4) \vee (x_1 \leq 8 \wedge x_2 \leq 10) \vee (x_2 \leq 5)\}.$$

As illustrated in Fig. 5, it is possible to write X as the following finite union of ideals:

$$\downarrow 4 \times \mathbb{N} \ \cup \ \downarrow 8 \times \downarrow 10 \ \cup \ \mathbb{N} \times \downarrow 5$$

which can be represented by $\{(4, \omega), (8, 10), (\omega, 5)\}$.

Example 19. It has been recently shown that downward closed languages (under the subword ordering) coincide with the class of *strictly piecewise-testable languages* [25]. Previously, downward closed languages were studied and used in [26] for representing infinite reachability subsets of lossy channel systems; it is proved that every downward closed language on Σ^*, where Σ is a finite alphabet, is a finite union of *products* $P_1 P_2 \cdots P_m$ where each P_i is either $\{\varepsilon, \sigma\}$ for some $\sigma \in \Sigma$, or A^* for some $A \subseteq \Sigma$. It has been remarked in [10] that every ideal $I \in \mathsf{Ideals}(\Sigma^*)$, is exactly a product $I = P_1 P_2 \cdots P_m$ like in [26]. Following [10], the previous result on downward closed languages is then a particular instance of a more general result: every downward closed set (here a downward closed language on Σ^*), in a wqo, is a finite union of ideals.

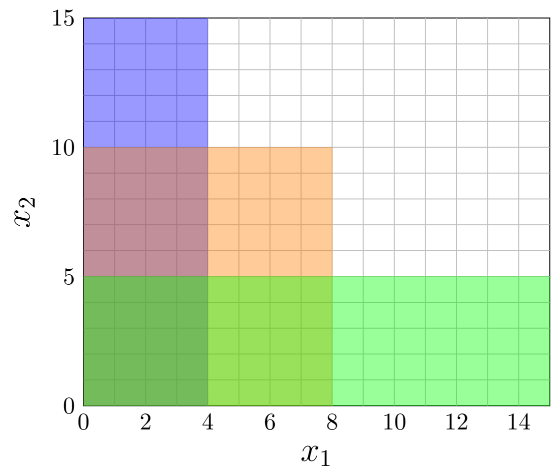

Figure 5: Decomposition of $X = \{(x_1, x_2) \in \mathbb{N}^2 : (x_1 \leq 4) \vee (x_1 \leq 8 \wedge x_2 \leq 10) \vee (x_2 \leq 5)\}$ into finitely many ideals. The three ideals $\downarrow 4 \times \mathbb{N}, \downarrow 8 \times \downarrow 10$ and $\mathbb{N} \times \downarrow 5$ appear respectively in blue, orange and green.

For example, consider the language of words over $\Sigma = \{a, b, c\}$ where the first letter does not reappear, *i.e.*, let

$$\begin{aligned} L &= \{w \in \Sigma^+ : w_i \neq w_1 \text{ for } 1 < i \leq |w|\} \\ &= a\{b, c\}^* \cup b\{a, c\}^* \cup c\{a, b\}^* \ . \end{aligned}$$

It can be shown that

$$\begin{aligned} \downarrow L &= L \cup \{w \in \Sigma^* : |w|_\sigma = 0 \text{ for some } \sigma \in \Sigma\} \\ &= L \cup \{a, b\}^* \cup \{a, c\}^* \cup \{b, c\}^* \\ &= \{a, \varepsilon\}\{b, c\}^* \cup \{b, \varepsilon\}\{a, c\}^* \cup \{c, \varepsilon\}\{a, b\}^* \ . \end{aligned}$$

Hence, $\downarrow L$ decomposes into finitely many ideals.

Before proving formally that ideal decompositions such as in Example 18 and Example 19 exist, we prove the following simple yet useful proposition that states that any ideal contained in a finite union of ideals is contained in one of these ideals.

Proposition 20. *Let $I, J_1, J_2, \ldots, J_m \in \mathsf{Ideals}(X)$ where X is a quasi-ordered set, then $I \subseteq J_1 \cup J_2 \cup \cdots \cup J_m$ if, and only if, $I \subseteq J_j$ for some $1 \leq j \leq m$.*

Proof. We claim that if a directed set I is included in $J \cup K$ where J and K are downward closed, then either $I \subseteq J$ or $I \subseteq K$. The claim implies the proposition by a straightforward induction since an ideal is directed and any union of ideals is downward closed.

To see the claim, let $I \subseteq J \cup K$ under the conditions stated and suppose to the contrary that there exist $s \in I \setminus J$ and $t \in I \setminus K$. Since I is directed, there exists $u \in I$ such that $s \leq u$ and $t \leq u$. Since $u \in I$, either $u \in J$ or $u \in K$. By downward closures of J and K, either $s \in J$ or $t \in K$, a contradiction that proves the claim. □

Theorem 21. *Let D be a downward closed subset of a well-quasi-ordered set X, then $D = I_1 \cup I_2 \cup \cdots \cup I_m$ for some $I_1, I_2, \ldots, I_m \in \mathsf{Ideals}(X)$. Moreover, there is exactly one collection of such ideals that are pairwise incomparable, namely the maximal ideals contained in D (w.r.t. inclusion).*

Proof. We say that a subset $D \subseteq X$ is *bad* if it is downward closed and does not admit a finite decomposition in ideals. Assume, to obtain a contradiction, that a bad D exists. We can assume that D is minimal for inclusion among bad subsets since strictly decreasing subsequences of downward closed subsets are finite in a well-quasi-ordered set.

Let us now show that D is directed. First D is not empty since $\emptyset$ is equal to an empty union. Let $x_1, x_2 \in D$. Since $D \setminus \uparrow x_1$ and $D \setminus \uparrow x_2$ are downward closed and strictly included in D, they are not bad, by minimality of D. Thus, $D \setminus \uparrow x_1 = \bigcup_{j=1}^{n} I_j$ and $D \setminus \uparrow x_2 = \bigcup_{j=n+1}^{m} I_j$ for some ideals $I_1, I_2, \ldots, I_m \subseteq X$. Hence

$$D \setminus (\uparrow x_1 \cap \uparrow x_2) = (D \setminus \uparrow x_1) \cup (D \setminus \uparrow x_2) = \bigcup_{j=1}^{m} I_j$$

is not bad, hence differs from D. Therefore, $D \cap (\uparrow x_1 \cap \uparrow x_2)$ is not empty and, thus, there exists $y \in D$ such that $x_1 \leq y$ and $x_2 \leq y$. We conclude that D is directed and therefore an ideal, contradicting our assumption. Thus, D is equal to a finite union of ideals.

Assume that there exist two distinct sets $\{I_1, I_2, \ldots, I_m\}$ and $\{J_1, J_2, \ldots, J_n\}$ of pairwise incomparable ideals each of whose union is D. Suppose with no loss of generality that $I_1 \notin \{J_1, J_2, \ldots, J_n\}$. By applying Prop. 20 twice, $I_1 \subseteq J_j \subseteq I_i$ for some $1 \leq j \leq n$ and some $1 \leq i \leq m$. Since $I_1 \neq J_j$, $I_1 \subsetneq J_j \subseteq I_i$ so that $i \neq 1$. But then I_1 and I_i are comparable, a contradiction that completes the proof. □

Remark 22. A proof of the first half of Theorem 21 appeals to a technical bridge between topological completions and ordering completions of sets [10]. The above self-contained proof was later provided by Goubault-Larrecq [27] and will be part of a future paper by a group of authors including Goubault-Larrecq [28]. It has recently been noticed that this result has a long history. It has seemingly been first proved by Erdős and Tarski in 1943 in a more general setting [29]. The proof presented here is more reminiscent of proofs given later by Bonnet [30] and Fraïssé [31].

Definition 23. We will denote the *ideal decomposition* of a downward closed set D, stated in Theorem 21, as $\mathsf{IdealDecomp}(D)$, *i.e.*, $\mathsf{IdealDecomp}(D)$ is the finite set of maximal ideals contained in D w.r.t. inclusion.

Since downward closed sets can be represented by finitely many ideals, it will often be sufficient for algorithms to manipulate ideals, in particular to test for inclusion. Fortunately, inclusion between ideals is decidable for well-quasi-ordered sets obtained by closing finite sets and natural numbers under finite products, disjoint sums, the multiset operator and the Kleene star (respectively with their natural associated orderings) [10]. Therefore inclusion of ideals of $\mathbb{N}^d$ and inclusion of ideals of Σ^* are both decidable. Moreover:

Proposition 24. *Let X be a well-quasi-ordered set, then $\mathsf{Ideals}(X)$ is countable if, and only if, X is countable.*

Proof. Suppose that X is countable. Any upward closed subset is the upward closure of a finite basis, hence the number of upward closed sets is equal to the number of finite subsets of X, which is countable. Since the complement of an upward closed set is downward closed and vice versa, upward closed subsets are in bijection with downward closed subsets. Since ideals are downward closed sets, we conclude that $\mathsf{Ideals}(X)$ is countable.

Suppose that $\mathsf{Ideals}(X)$ is countable, then X is countable since $x \mapsto \downarrow x$ is an injective mapping from X to $\mathsf{Ideals}(X)$. $\square$

4.2. Completion of WSTS

Let $S = (X, \to, \leq)$ be a functional WSTS (recall Definition 2) defined by a set F of non decreasing partial functions, the *functional completion* of S, introduced by Finkel & Goubault-Larrecq [11], is defined by

$$\overline{S} = (\overline{X}, \overline{\to}, \subseteq)$$

where $\overline{X} = \mathsf{Ideals}(X)$ and $\overline{\to}$ is defined by $\overline{F}$ the set of functions $\overline{f} : \mathsf{Ideals}(X) \to \mathsf{Ideals}(X)$ such that $\overline{f}(I) \stackrel{\text{def}}{=} \downarrow f(I)$ for every $f \in F$. We note that $\overline{f}$ is well-defined since $\overline{f}(I)$ is an ideal. An elementary proof of this fact is given in the next proposition. A more general result expressed in a topological framework can be found in [11].

Proposition 25. *Let X be a well-quasi-ordered set, let $f : X \to X$ be a non decreasing function, and let $I \in \mathsf{Ideals}(X)$, then $\overline{f}(I)$ is an ideal.*

Proof. First, $\overline{f}(I)$ is downward-closed by definition. Let us verify that $\overline{f}(I)$ is also directed. Let $a', b' \in \overline{f}(I)$ then, by definition, there exist $a, b \in I$ such that $a' \leq f(a)$ and $b' \leq f(b)$. Now, since I is an ideal, it is directed, hence there exists $c \in I$ such that $a \leq c$ and $b \leq c$. Hence, as f is non decreasing, we have $f(a) \leq f(c)$ and $f(b) \leq f(c)$. By transitivity, we obtain $a' \leq f(c)$ and $b' \leq f(c)$. Since $f(c) \in f(I) \subseteq \overline{f}(I)$, we conclude that $\overline{f}(I)$ is directed. $\square$

Here we extend the notion of completion to any WSTS, including infinitely branching WSTS:

Definition 26. The *completion* $\widehat{S}$ of a WSTS $S = (X, \rightarrow, \leq)$ is the ordered transition system $\widehat{S} = (\widehat{X}, \rightsquigarrow, \subseteq)$ where $\widehat{X} = \mathsf{Ideals}(X)$ and $I \rightsquigarrow J$ if $J \in \mathsf{IdealDecomp}(\downarrow \mathrm{Post}_S(I))$.

Let $S = (X, \rightarrow, \leq)$ be a functional WSTS, then, by definition, the following relations hold between S, $\overline{S}$ and $\widehat{S}$ for every $I \in \mathsf{Ideals}(X)$:

$$\bigcup_{J \in \mathrm{Post}_{\overline{S}}(I)} J = \bigcup_{\overline{f} \in \overline{F}} \overline{f}(I) = \bigcup_{f \in F} \downarrow f(I) = \bigcup_{J \in \mathrm{Post}_{\widehat{S}}(I)} J = \downarrow \mathrm{Post}_S(I).$$

Another good news is that:

Proposition 27. *$\widehat{S}$ is finitely branching for every WSTS $S = (X, \rightarrow, \leq)$.*

Proof. Let $I \in \mathsf{Ideals}(X)$. By definition of $\widehat{S}$,

$$\mathrm{Post}_{\widehat{S}}(I) = \mathsf{IdealDecomp}(\downarrow \mathrm{Post}_S(I)).$$

By Theorem 21, this set is finite, hence $\widehat{S}$ is finitely branching. $\square$

Moreover, the reachability sets of a WSTS and its completion are related in the following way:

Proposition 28. *For every WSTS $S = (X, \rightarrow, \leq)$ and every $x \in X$,*

$$\downarrow Post^*_S(x) = \bigcup_{I \in Post^*_{\widehat{S}}(\downarrow x)} I\ .$$

Proposition 28 is proved by comparing executions in a system with executions in its completion. This relation between executions, which will also be useful later to prove decidability results, is proved in Prop. 29 and Prop. 30 below.

Proposition 29. *Let $S = (X, \rightarrow, \leq)$ be a WSTS, and $I, J \in \widehat{X}$. If $I \rightsquigarrow^k J$ for some $k \in \mathbb{N}$, then for every $x_J \in J$ there exist $x_I \in I$, $y \in X$ and $k' \in \mathbb{N}$ such that $y \geq x_J$ and $x_I \rightarrow^{k'} y$. Moreover, if S has transitive monotonicity then $k' \geq k$; if S has strong monotonicity then $k' = k$.*

Proof. If $I \rightsquigarrow^0 J$ then $I = J$ and for every $x_J \in J$, one can pick $x_I = y = x_J$ and $k' = 0$.

Let $I \rightsquigarrow I' \rightsquigarrow^k J$. By induction, for every $x_J \in J$ there exist $x_{I'} \in I'$, $y' \geq x_J$ and $k' \in \mathbb{N}$ (resp. $k' \geq k$; $k' = k$) such that $x_{I'} \rightarrow^{k'} y'$. Since $x_{I'} \in \downarrow \mathrm{Post}_S(I)$, there exist $x_I \in I$ and $y'' \geq x_{I'}$ such that

$$x_I \rightarrow y''. \tag{2}$$

Moreover, since $y'' \geq x_{I'}$, we can apply (resp. transitive; strong) monotonicity to $x_{I'} \rightarrow^{k'} y'$ and deduce

$$y'' \rightarrow^{k''} y \tag{3}$$

for some $y \geq y'$ and $k'' \in \mathbb{N}$ (resp. $k'' \geq k'$; $k'' = k'$). Hence, by (2) and (3), we have $x_I \rightarrow y'' \rightarrow^{k''} y$ of length $k''+1$ (resp. $k''+1 \geq k'+1 \geq k+1$; $k''+1 = k'+1 = k+1$), with $y \geq y' \geq x_J$, completing the induction and proving the proposition. □

Proposition 30. *Let $S = (X, \rightarrow, \leq)$ be a WSTS and $x, y \in X$. If $x \rightarrow^k y$ for some $k \in \mathbb{N}$, then for every ideal $I \supseteq \downarrow x$ there exists an ideal $J \supseteq \downarrow y$ such that $I \rightsquigarrow^k J$.*

Proof. If $x \rightarrow^0 y$ then $x = y$ and $I \rightsquigarrow^0 I$ for every ideal $I \supseteq \downarrow x = \downarrow y$.

Let $x \rightarrow x' \rightarrow^k y$. Pick any ideal $I \supseteq \downarrow x$, then $x' \in \mathrm{Post}_S(I) \subseteq \downarrow \mathrm{Post}_S(I)$. Therefore, $I \rightsquigarrow I'$ for some ideal $I' \supseteq \{x'\}$. Since I' is downward closed, $I' \supseteq \downarrow x'$. By induction, there exists an ideal $J \supseteq \downarrow y$ such that $I' \rightsquigarrow^k J$, hence $I \rightsquigarrow I' \rightsquigarrow^k J$ has length $k+1$. □

We may now prove Prop. 28 that gives a relation between the reachability sets of a WSTS and its completion.

Proof of Proposition 28. Let $S = (X, \rightarrow, \leq)$ be a WSTS and $x \in X$.

Let $y \in \mathrm{Post}^*_S(x)$. By applying Prop. 30 with $I = \downarrow x$, we know that there exists an ideal $J \supseteq \downarrow y$ such that $J \in \mathrm{Post}^*_S(\downarrow x)$. Hence, $\downarrow y \subseteq \bigcup_{I \in \mathrm{Post}^*_S(\downarrow x)} I$. Thus, we have $\downarrow \mathrm{Post}^*_S(x) \subseteq \bigcup_{I \in \mathrm{Post}^*_{\widehat{S}}(\downarrow x)} I$.

Let $I = \downarrow x$, $J \in \mathrm{Post}^*_{\widehat{S}}(I)$ and $x_J \in J$. By Prop. 29, there exist $x_I \in I$, $y \geq x_J$ and $k \in \mathbb{N}$ such that $x_I \rightarrow^k y$. By definition of I and by monotonicity of S, $x \rightarrow^* y'$ for some $y' \geq y$. Hence, since $x_J \leq y \leq y'$, we have $x_J \in \downarrow \mathrm{Post}^*_S(x)$ and consequently $J \subseteq \downarrow \mathrm{Post}^*_S(x)$. □

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

Proposition 31. *$\widehat{S}$ has strong monotonicity for every WSTS $S = (X, \rightarrow, \leq)$.*

Proof. Let $I, I', J \in \widehat{X}$ and suppose that $I \rightsquigarrow J$ and $I \subseteq I'$. We have to show that $I' \rightsquigarrow J'$ for some $J' \in \widehat{X}$ such that $J \subseteq J'$. Let $\mathrm{Post}_{\widehat{S}}(I') = \{J'_1, J'_2, \ldots, J'_m\}$. Since $J \in \mathrm{Post}_{\widehat{S}}(I)$, we have

$$J \subseteq \downarrow \mathrm{Post}_S(I) \subseteq \downarrow \mathrm{Post}_S(I') = J'_1 \cup J'_2 \cup \ldots \cup J'_m.$$

By Prop. 20, we have $J \subseteq J'_j$ for some $1 \leq j \leq k$. Thus, $I' \rightsquigarrow J'_j$ and $J \subseteq J'_j$. □

Unfortunately, $\subseteq$ is not always a wqo for $\widehat{X}$, and therefore the completion is not always a WSTS. But we may totally characterize those WSTS S such that $\widehat{S}$ is still a WSTS. Let us first recall the characterization of ω^2-wqos from Jančar [32]: a wqo $\leq$ is a *ω^2-wqo* if, and only if, $\leq^\#$ is a wqo, where $\leq^\#$

is the Smyth ordering defined by $A \leq^{\#} B \Leftrightarrow \uparrow B \subseteq \uparrow A$ (or equivalently, $A \leq^{\#} B \Leftrightarrow \forall b \in B, \exists a \in A, a \leq b$). Now, it is known that $\subseteq$ is a wqo for $\widehat{X}$ if, and only if, $\leq$ is an ω^2-wqo for X (*e.g.* see [11, Prop. 3]).

In general, a wqo is not necessarily an ω^2-wqo and the typical counter-example is the Rado set [33],

$$X_{\text{Rado}} \stackrel{\text{def}}{=} \{(m, n) \in \mathbb{N}^2 : m < n\},$$

ordered by $\leq_{\text{Rado}}$ where

$$(m, n) \leq_{\text{Rado}} (m', n') \Leftrightarrow (m = m' \wedge n \leq n') \vee (n < m').$$

It is well-known that $\leq_{\text{Rado}}$ is a well-quasi-ordering, and that $2^{X_{\text{Rado}}}$ is not well-quasi-ordered by $\leq^{\#}_{\text{Rado}}$ [32].

We extend the terminology to WSTS and say that a WSTS $S = (X, \rightarrow, \leq)$ is a ω^2-*WSTS* if $\leq$ is an ω^2-wqo for X. We obtain the following result generalizing the known result for functional WSTS [11]:

Theorem 32. *Let S be a WSTS, then $\widehat{S}$ is a WSTS if, and only if, S is an ω^2-WSTS.*

We also observe that a WSTS inherits the strict monotonicity of its completion but not conversely.

Proposition 33. *Let $S = (X, \rightarrow, \leq)$ be a WSTS. If $\widehat{S}$ has strict monotonicity, then so does S. However, if S has strict monotonicity then $\widehat{S}$ does not necessarily have it.*

Proof. Suppose $\widehat{S}$ has strict monotonicity. Let $x, x', y \in X$ be such that $x \rightarrow y$ and $x < x'$. We have to show that $x' \rightarrow^* y'$ for some $y' \in X$ such that $y < y'$.

Let $\text{Post}_{\widehat{S}}(\downarrow x) = \{J_1, J_2, \ldots, J_m\} \subseteq \mathsf{Ideals}(X)$. Since $y \in \text{Post}_S(x)$, we have

$$\downarrow y \subseteq \downarrow \text{Post}_S(\downarrow x) = J_1 \cup J_2 \cup \ldots \cup J_m.$$

By Prop. 20, we have $\downarrow y \subseteq J_i$ for some $1 \leq i \leq n$. Let $J = J_i$. We have $\downarrow x \rightsquigarrow J$ and $\downarrow y \subseteq J$.

Since $\downarrow x \subset \downarrow x'$, there exists $J' \in \widehat{X}$ such that $\downarrow x' \rightsquigarrow^* J'$ and $J \subset J'$ by hypothesis on strict monotonicity. Since $J \subseteq J'$, we have $y \in J'$. Let $b \in J' \setminus J$, then there exists $c \in J'$ such that $y \leq c$ and $b \leq c$ since J' is directed. Note that $c \notin J$, because the opposite would imply $b \in J$ since J is downward closed. Moreover $c > y$ since $c = y$ implies $c \in J$.

By Prop. 29, there exists $x'' \in \downarrow x'$ and $c' \geq c$ such that $x'' \rightarrow^* c'$. By (standard) monotonicity, we have $x' \rightarrow^* y'$ for some $y' \geq c'$. Since $y' \geq c' \geq c > y$, it shows strict monotonicity for S.

To show that the converse fails, let $S = (\mathbb{N}^2, \rightarrow, \leq)$ be the WSTS such that $(a, b) \rightarrow (0, a + b)$, then S has strict monotonicity. Let $I = \mathbb{N} \times \downarrow 1$ and $I' = \mathbb{N} \times \downarrow 2$. We have $I \subset I'$, but $\text{Post}_{\widehat{S}}(I) = \text{Post}_{\widehat{S}}(I') = \{\downarrow 0 \times \mathbb{N}\}$. Therefore $\widehat{S}$ does not have strict monotonicity. □

4.2.1. Effectiveness

The completion of a WSTS will often be useful, assuming, minimally, that we can manipulate it. Let $\mathcal{C}$ be a class of WSTS, we let $\widehat{\mathcal{C}} \stackrel{\text{def}}{=} \{\widehat{S} : S \in \mathcal{C}\}$ and $\widehat{\mathcal{C}}(i) \stackrel{\text{def}}{=} \mathcal{C}(i)$. When manipulating the completions, we naturally require for any class $\mathcal{C}$ of WSTS that a set $\widehat{E}_{\mathcal{C}} \subseteq \mathbb{N}$ and a surjective map $\widehat{r} : \widehat{E}_{\mathcal{C}} \to \bigcup_i \widehat{X_i}$ be understood where X_i is the set of states of $\mathcal{C}(i)$. Let $\widehat{E}_{X_i} = \{e \in \widehat{E}_{\mathcal{C}} : r(e) \in \widehat{X_i}\}$, we further require the set $\{(i, e) : i \in \mathbb{N}, e \in \widehat{E}_{X_i}\}$ to be decidable.

Definition 34. Let a class $\mathcal{C}$ of WSTS have the property that

(a) there exists a Turing machine $M_{\downarrow}$ that computes, on input (i, x) where $i \in \mathbb{N}$ and x is a state of $\mathcal{C}(i)$, some $\widehat{e} \in \widehat{E}_{\mathcal{C}}$ such that $\widehat{r}(\widehat{e}) = \downarrow x$;

(b) there exists a Turing machine $M_{\uparrow^c}$ that computes, on input $(i, \{x_1, x_2, \ldots, x_m\})$ where $i \in \mathbb{N}$ and $\{x_1, x_2, \ldots, x_m\}$ is a set of states of $\mathcal{C}(i)$, some $\widehat{e_1}, \widehat{e_2}, \ldots, \widehat{e_n} \in \widehat{E}_{\mathcal{C}}$ such that

$$\mathsf{IdealDecomp}\left(\overline{\uparrow\{r(x_1), r(x_2), \ldots, r(x_m)\}}\right) = \{\widehat{r}(\widehat{e}_1), \widehat{r}(\widehat{e}_2), \ldots, \widehat{r}(\widehat{e}_n)\}\,.$$

Such a class $\mathcal{C}$ is called *completion-effective* (resp. *completion-post-effective*[6]) if the class $\widehat{\mathcal{C}}$ is effective (resp. post-effective). By extension, we say that a WSTS S is *completion-effective* (resp. *completion-post-effective*) if the degenerate class $\{S\}$ is completion-effective (resp. completion-post-effective).

Definitions of effective "S-representations" for various ideals (and for various topological spaces) S, similar to Definition 34, can be found in [10]; the idea is to ensure that irreductible downward closed sets (like $\downarrow x$) can be manipulated and that inclusion and intersection between ideals can effectively be tested. The specific conditions appearing in Definition 34 also appeared in a similar form in [34, Sect. 4.3.4] but the latter is not adapted for dealing with infinite sets of WSTS.

We note that the post-effectiveness of a class $\mathcal{C}$ of WSTS is independent from the post-effectiveness of its completion $\widehat{C}$. Actually, this is even true for degenerate classes of WSTS as shown in the two following propositions.

Proposition 35. *There exists a post-effective WSTS that is not completion-effective.*

Proof. The argument mimics the proof of [20, Prop. 2.4]. Let $S = (\mathbb{N}^2, \to, \leq)$ be the WSTS such that

$$(m, n) \to (m + |\{i \leq m : \mathsf{Turing}_i \text{ halts on its encoding in at most } n \text{ steps}\}|\,, n)\,.$$

Note that S is post-effective because $\to$ and $\leq$ are decidable under a natural encoding of $\mathbb{N}^2$ and $|\mathrm{Post}_S(x)| = 1$ for every $x \in \mathbb{N}^2$.

[6] It happens here that, unlike for WSTS, the outcome of $M_{|\mathrm{Post}|}$ on $(i, \widehat{x})$ is never ∞ since completions are finitely branching.

Let us show that $\widehat{S}$ is not effective. Let I_m be the ideal $\downarrow m \times \mathbb{N}$ and let J_m be the unique ideal such that $I_m \rightsquigarrow J_m$. One can show that for every $m > 0$, there exists $a \in \mathbb{N}$ such that $J_{m-1} = I_a = \downarrow a \times \mathbb{N}$ and

$$J_m = \begin{cases} \downarrow(a+2) \times \mathbb{N} & \text{if } M_m \text{ halts on its encoding,} \\ \downarrow(a+1) \times \mathbb{N} & \text{otherwise.} \end{cases}$$

Assume $\widehat{S}$ to be effective and let us show that we can decide whether M_m halts or not on its encoding. By definition, it is possible to compute the encoding of I_i by computing $\mathsf{IdealDecomp}(\overline{\uparrow\{(i+1,0)\}})$ which is equal to $\downarrow i \times \mathbb{N}$. Therefore, we can compute I_{m-1} and I_m which allows us to compute J_{m-1} and J_m since $\rightsquigarrow$ is computable. We know that $J_{m-1} = I_a = \downarrow a \times \mathbb{N}$ for some $a \in \mathbb{N}$. As explained earlier, we can compute the ideals $I_0, I_1, \ldots$, hence, to determine a, we test $I_i \subseteq J_{m-1}$ for $i \geq 0$ until we reach $I_{a+1} \not\subseteq J_{m-1}$. Note that testing $I_i \subseteq J_{m-1}$ is possible since $\widehat{S}$ is effective. Similary, we know that $J_m = I_{a+b}$ for some $b \in \{1, 2\}$ and we can find the value of b with the same process. If $b = 1$, then M_m halts on its own encoding, otherwise it does not. Therefore, if $\widehat{S}$ was effective, we could decide the halting problem, which is impossible. □

Proposition 36. *There exists a non post-effective WSTS that is completion-post-effective.*

Proof. Let $S = (\mathbb{N}_\omega, \to, \leq)$ be the WSTS such that

- $i \to j$ if $i, j \in \mathbb{N}$ and Turing_i runs on its encoding for more than j steps,
- $i \to \omega$ for every $i \in \mathbb{N}_\omega$.

Note that S is effective. However, S is not post-effective. Indeed, $\mathrm{Post}_S(i)$ is finite if, and only if, Turing_i halts on its encoding. Therefore, if S was post-effective, it would come equipped with a Turing machine powerful enough to decide the halting problem, which would be a contradiction. On the other hand, $\widehat{S}$ is post-effective since $\mathrm{Post}_{\widehat{S}}(I) = \{\mathbb{N}_\omega\}$ for every $I \in \mathsf{Ideals}(\mathbb{N}_\omega)$. □

4.3. Completion-post-effectiveness of concrete classes of WSTS

Let us examine some prominent classes of WSTS, introduced in Sect. 3.2, and let us show that they are completion-post-effective.

4.3.1. Affine nets and Petri nets

Let $S = (\mathbb{N}^d, \to, \leq)$ be an affine net with affine functions F. As noted in Example 18, the ideals of $\mathbb{N}^d$ can be represented by elements of $\mathbb{N}^d_\omega$. This way, we can simply extend each $f \in F$ to $\mathbb{N}^d_\omega$ with the rule $\omega + x = \omega$ for every $x \in \mathbb{N}^d_\omega$. Let us call $\widehat{F}$ the extension of F to $\mathbb{N}^d_\omega$. The completion of S is

$$\widehat{S} = (\mathbb{N}^d_\omega, \rightsquigarrow, \leq)$$

where $I \rightsquigarrow J$ si $J \in \max\{\widehat{f}(I) : \widehat{f} \in \widehat{F}\}$. Let $I \in \mathbb{N}_\omega^d$, then $\mathrm{Post}_{\widehat{S}}(I)$ can easily be computed by computing

$$\left\{\widehat{f}(I) : \widehat{f} \in \widehat{F}\right\}$$

and remove its non maximal elements by applying Prop. 20. Hence, affine nets are completion-post-effective and consequently Petri nets too.

4.3.2. ω-Petri nets

In order to show that ω-Petri nets are completion-post-effective, we need to extend transitions of ω-Petri nets to ideals, and take the downward closure of their images.

Let $S = (\mathbb{N}^d, \rightarrow, \leq)$ be an ω-Petri net with d places. Let t be a transition of S such that its input arc from place i is labelled by $a \in \mathbb{N}_\omega$ and its output arc to place i is labelled by $b \in \mathbb{N}_\omega$. We extend t over ideals, and take the downward closure of its image as follows. Let $I \in \mathsf{Ideals}(\mathbb{N}^d)$. If $a = b = 0$, then t leaves I_i unchanged. Assume that $a \neq 0$ or $b \neq 0$. If $I_i = \mathbb{N}$, then t maps I_i to $\mathbb{N}$; if $I_i = \downarrow n$ for some $n \in \mathbb{N}$, then t maps I_i to

- $\downarrow(n - a + b)$ if $a \in \mathbb{N}$, $b \in \mathbb{N}$, and $n \geq a$,
- $\downarrow(n + b)$ if $a = \omega$, $b \in \mathbb{N}$,
- $\mathbb{N}$ if $a \in \mathbb{N}$, $b = \omega$, and $n \geq a$,
- $\mathbb{N}$ if $a = \omega$, $b = \omega$,
- undefined otherwise.

As in the case of Petri nets, ideals may be represented by elements of $\mathbb{N}_\omega^d$, and in order to compute $\mathrm{Post}_{\widehat{S}}(I)$, we may compute the maximal elements of

$$\{J : J \text{ is obtained by firing some transition } t \text{ from } I\}$$

by applying Prop. 20.

4.3.3. Lossy channel systems

For simplicity, we restrain ourselves to lossy channel systems with a single channel. Our reasoning can be extended to many channels and a proof can be found in another formalism in [35]. Let $S = (Q \times \Sigma^*, \rightarrow, \preceq)$ be a lossy channel system, and let t be one of its transitions. We extend t to ideals and take the downward closure of its image. Let $I \in \mathsf{Ideals}(Q \times \Sigma^*)$, then as explained in Example 19, $I = (p, P_1 P_2 \cdots P_m)$ where each P_i is either $\{\varepsilon, \sigma\}$ for some $\sigma \in \Sigma$, or A^* for some $A \subseteq \Sigma$. If t writes letter $a \in \Sigma$ and goes to state q, then t maps I to

- $(q, P_1 P_2 \cdots P_m)$ if $P_m = A^*$ with $a \in A$,
- $(q, P_1 P_2 \cdots P_m \{\varepsilon, a\})$ otherwise.

If t reads letter $a \in \Sigma$ and goes to state q, then t maps I to

- $(q, P_{i+1}P_{i+2}\cdots P_m)$ if i is the smallest i s.t. $a \in P_i$ and $P_i = \{\varepsilon, a\}$,
- $(q, P_iP_{i+1}\cdots P_m)$ if i is the smallest i s.t. $a \in P_i$ and $P_i \neq \{\varepsilon, a\}$,
- undefined otherwise.

Therefore, we may represent ideals by regular expressions (or finite auomata), and in order to compute $\mathrm{Post}_{\widehat{\mathcal{S}}}(I)$, we may compute the maximal elements of

$$\{J : J \text{ is obtained by going through some transition } t \text{ from } I\}$$

by applying Prop. 20 combined with regular languages inclusion.

5. Decidability and undecidability in infinitely branching WSTS

One of the goals of this paper is to extend the decidability of termination, boundedness and maintainability given by Theorem 15 to the more general case of *infinitely branching* WSTS. Our goal for the coverability problem, which is decidable for infinitely branching WSTS, is to investigate alternative effectiveness hypotheses.

5.1. Termination

We first note that termination is undecidable for infinitely branching WSTS even with many hypotheses.

Theorem 37. *Termination is undecidable for some post-effective and completion-post-effective class of ω^2-WSTS with strong and strict monotonicity, and well partial ordering.*

Proof. We know from [36, 37] that structural termination, *i.e.* deciding whether a WSTS terminates from every initial state $x \in X$, is undecidable for Petri nets with transfers. Petri nets with transfers form a post-effective and completion-post-effective class $\mathcal{C}$ of ω^2-WSTS with strong and strict monotonicity, and well partial ordering. We show that structural termination for $\mathcal{C}$ reduces to termination for some class $\mathcal{D}$.

Let $\mathcal{C}(i) = (X, \to, \leq)$ and let $\mathcal{D}(i) \stackrel{\text{def}}{=} (X \cup \{x_0\}, \to \cup \{(x_0, x) : x \in X\}, \leq)$ be the WSTS obtained from $\mathcal{C}(i)$ by adding a new "initial" element x_0. By definition, $\mathcal{D}(i)$ is also an ω^2-WSTS with strong and strict monotonicity. Note that $\mathcal{D}$ is infinitely branching because of x_0. We claim that the class $\mathcal{D}$ is still post-effective and completion-post-effective. Indeed, encoding x_0 only requires an extra special symbol; the only new ideal is $\{x_0\}$ and can also be encoded with an extra special symbol; and the Turing machines computing $\to$, $\leq$, $\rightsquigarrow$ and $\subseteq$ are easily adapted to handle these two new symbols. Now, note that $\mathcal{C}(i)$ structurally terminates if, and only if, $\mathcal{D}(i)$ terminates from x_0. $\square$

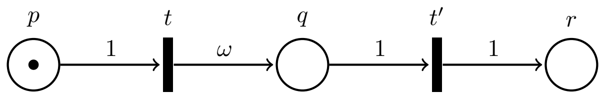

Figure 6: Example of an ω-Petri net that terminates from marking $x_0 = (1, 0, 0)$, but does not terminate in the strong sense.

Recall that strong termination asks whether the length of executions is bounded, whereas termination asks whether there is no infinite excution. When a WSTS is infinitely branching, its termination problem differs in a subtle way from its strong termination problem. For example, consider the ω-Petri net (see Example 5) illustrated in Fig. 6. Consider $x_0 = (1, 0, 0)$, marking respectively p, q and r. As illustrated in Fig. 7, initially, we can only fire t which yields $(0, m, 0)$ for some $m \in \mathbb{N}$. Afterwards, we may only fire t' at most m times. Hence, this ω-Petri net terminates from x_0 since there is no infinite execution, however it does not terminate in the strong sense since the length of executions is unbounded.

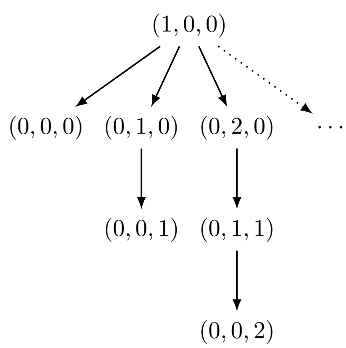

Figure 7: Reachability tree from $x_0 = (1, 0, 0)$ of the ω-Petri net depitcted in Fig. 6.

We show that, in contrast with termination, strong termination is decidable under suitable hypotheses:

Theorem 38. *Strong termination is decidable for any completion-post-effective class of ω^2-WSTS with transitive monotonicity.*

Proof. Let $\mathcal{C}$ be a completion-post-effective class of ω^2-WSTS with transitive monotonicity. It has been shown by Finkel and Schnoebelen [2, Theorem 4.6], that termination, and thus strong termination, is decidable for any post-effective class of finitely branching WSTS with transitive monotonicity. Let $S \in \mathcal{C}$, since S is an ω^2-WSTS, $\widehat{S}$ is a WSTS. Moreover, $\widehat{S}$ has strong monotonicity by Prop. 31 and thus transitive monotonicity. Since $\mathcal{C}$ is completion-post-effective, by definition, $\widehat{\mathcal{C}}$ is post-effective. Therefore, $\widehat{\mathcal{C}}$ is a post-effective class of finitely branching WSTS with transitive monotonicity, and thus strong termination is decidable for $\widehat{\mathcal{C}}$.

Let us see how this implies decidability of strong termination for $\mathcal{C}$. We claim that no bound on the length of executions from x_0 exists in $S \in \mathcal{C}$ if, and only if, no bound on the length of executions from $\downarrow x_0$ exists in $\widehat{S}$. Hence deciding strong termination from x_0 in $S \in \mathcal{C}$ follows from being able to decide strong termination from $\downarrow x_0$ in $\widehat{S}$. Since $\mathcal{C}$ is completion-post-effective, the latter can be decided by obtaining the encoding of $\downarrow x_0$ from x_0.

To see the claim, let $S = (X, \rightarrow, \leq) \in \mathcal{C}$ and suppose that there is no bound on the length of executions in S from $x_0 \in X$. By Prop. 30, there is no $k \geq 0$ that bounds the length of every execution from $\downarrow x_0$ in $\widehat{S}$. In more detail, suppose to the contrary that the longest execution from $\downarrow x_0$ in $\widehat{S}$ has length $k \geq 0$; this is contradicted by Prop. 30 since an execution of length $k' > k$ exists from x_0 in S. This proves the "only if".

Conversely, suppose that there is no bound on the length of executions from $\downarrow x_0$ in $\widehat{S}$. By Prop. 29, no $k \geq 0$ bounds the length of every execution from x_0 in S. This proves the "if". Hence the claim holds and strong termination is decidable for $\mathcal{C}$. $\square$

5.2. Boundedness

Drawing from [24], we know that boundedness is undecidable, even for some post-effective and completion-post-effective classes of finitely branching ω^2-WSTS with strong (but not strict) monotonicity.

It is known that for post-effective classes of finitely branching WSTS with transitive and strict monotonicity, and a well partial ordering, the boundedness problem is decidable [2]. We generalize this result to (possibly) infinitely branching WSTS and we note that the hypothesis of transitive monotonicity was not necessary in the proof of [2]. The proof follows [2] by building a finite reachability tree, with the extra step of testing whether $\mathrm{Post}_S(x)$ is infinite for each new node.

Theorem 39. *Boundedness is decidable for any post-effective class of WSTS with strict monotonicity and well partial ordering.*

Proof. Let $\mathcal{C}$ be a post-effective class of WSTS with strict monotonicity and well partial ordering. Let $S = (X, \rightarrow, \leq) \in \mathcal{C}$, and let $x_0 \in X$. We build a reachability tree T with root c_0 labelled x_0. If $\mathrm{Post}_S(x_0)$ is infinite, then we return "unbounded", otherwise we mark c_0 and for every $x \in \mathrm{Post}_S(x_0)$ we add a child labelled x to c_0. The tree is then built iteratively in the following way. An unmarked node c labelled x is picked,

- if c has an ancestor c' labelled x' such that $x' < x$, we return "unbounded";
- otherwise, if c has an ancestor c' labelled x' such that $x' = x$, we mark c;
- otherwise, if $\mathrm{Post}_S(x)$ is infinite, we return "unbounded";

- otherwise, we mark c and for every $y \in \mathrm{Post}_S(x)$ we add a child labelled y to c.

We prove that the procedure always terminates. First note that T is finitely branching since children are added to a node labelled x only when $\mathrm{Post}_S(x)$ is finite. Suppose to the contrary that T is infinite, then by König's lemma there exists an infinite path labelled $x_0, x_1, \ldots$ in T. Since $\leq$ is a wqo, there exist $i < j$ such that $x_i \leq x_j$. If $x_i < x_j$, then the algorithm returned "unbounded". Otherwise, if $x_i = x_j$, then the node labelled x_j was marked. Therefore, the path is not infinite which is a contradiction.

We prove that the algorithm is correct. We note that since S has strict monotonicity and $\leq$ is a wpo, S is unbounded from x_0 if, and only if, there exists an execution $x_0 \to x_1 \to \ldots \to x_k$ such that $x_i \neq x_j$ for every $i \neq j$, and either $\mathrm{Post}_S(x_k)$ is infinite or $x_m < x_k$ for some $m < k$. Therefore, when the algorithm returns "unbounded", S is indeed unbounded. Note that this property would not hold with a wqo, since $x \neq y$ and $x \leq y$ does not necessarily imply $x < y$ without antisymmetry.

Conversely, suppose that there exists such an execution $x_0 \to x_1 \to \ldots \to x_k$, then it appears in T. If $\mathrm{Post}_S(x_k)$ is infinite, then the algorithm returns "unbounded" which is correct. Otherwise, the algorithm also returns "unbounded" since $x_m < x_k$. □

5.3. Maintainability

We prove that maintainability is undecidable for classes of infinitely branching WSTS even with many hypotheses.

Theorem 40. *Maintainability is undecidable for some post-effective and completion-post-effective class of ω^2-WSTS with strong and strict monotonicity, and well partial ordering.*

Proof. We give a reduction from the (non) termination problem which was proven undecidable under the same hypotheses in Theorem 37.

Let $\mathcal{C}$ be a post-effective and completion-post-effective class of ω^2-WSTS with strong and strict monotonicity, and well partial ordering. Let $\mathcal{C}(i) = (X, \to, \leq)$ and let $x_0 \in X$. Let $(X', \to', \leq')$ be a disjoint copy of $\mathcal{C}(i)$, and let $\mathcal{D}(i) \stackrel{\text{def}}{=} (X \cup X' \cup \{x_{\min}\}, \to'', \leq'')$ be the ordered transition system such that

- $\to'' \stackrel{\text{def}}{=} \to \cup \to' \cup \{(x, x') : x \in X)\}$,
- $\leq'' \stackrel{\text{def}}{=} \leq \cup \leq' \cup \{(x_{\min}, x) : x \in X \cup \{x_{\min}\}\}$.

Note that it can be shown that $\mathcal{D}(i)$ is a ω^2-WSTS with strong and strict monotonicity. Moreover, $\mathcal{D}$ is a post-effective and completion-post-effective class of WSTS. Indeed, encoding the disjoint copy of X can be achieved with an extra special symbol as a prefix to the original encoding of X, moreover $x_{\min}$ can be encoded with an extra symbol. The encoding of the ideals of $X \cup X' \cup \{x_{\min}\}$, *i.e.*, $\{I \cup \{x_{\min}\} : I \in \mathsf{Ideals}(X)\}$ and a disjoint copy of $\mathsf{Ideals}(X)$, can easily be adapted from the original encoding of $\mathsf{Ideals}(X)$. The Turing machines computing $\to$, $\leq$, $\rightsquigarrow$ and $\subseteq$ are also easily adapted to handle these new encodings.

Let $T = \uparrow x_{\min}$ and let us prove that $\mathcal{C}(i)$ does not terminate from x_0 if, and only if, $\mathcal{D}(i)$ is maintainable in T from x_0. Suppose that there exists an infinite $x_0, x_1, \ldots$ such that $x_0 \to x_1 \to \ldots$. By definition, the same sequence is an execution in $\mathcal{D}(i)$. Moreover, $x_{\min} \leq'' x$ for every $x \in X$. Therefore, $x_0 \to''_T x_1 \to''_T \ldots$, hence $\mathcal{D}(i)$ is maintainable in T from x_0. Conversely,

(a) suppose that there exists a finite sequence $x_0, x_1, \ldots, x_k$ such that $x_0 \to''_T x_1 \to''_T \ldots \to''_T x_k$ and $\mathrm{Post}_{\mathcal{D}(i)}(x_k) = \emptyset$, then for every $0 \leq j \leq k$ we have $x_j \in X$ since $X' \cap T = \emptyset$ and $x_{\min}$ is not reachable from x_0. Therefore, $x'_k \in \mathrm{Post}_{\mathcal{D}(i)}(x_k)$. This is a contradiction, hence there is no such sequence;

(b) suppose that there exists an infinite sequence $x_0, x_1, \ldots$ such that $x_0 \to''_T x_1 \to''_T \ldots$, then, again, $x_j \in X$ for every $j \in \mathbb{N}$. Since every element is in X, it implies that $\mathcal{C}(i)$ does not terminate from x_0. $\square$

Recall that weak maintainability asks whether the length of executions staying in some upward closed set T is bounded, whereas maintainability deals with maximal executions staying in T. We note that the maintainability problem slightly differs from the weak maintainability problem. For example, reconsider the ω-Petri net introduced in Sect. 5.1 and illustrated in Fig. 6, but now with initial marking $x_0 = (1,1,0)$. Let $T = \uparrow(0,1,0)$. As illustrated in Fig. 8, there is no maximal execution that remains in T, hence this ω-Petri net is not maintainable in T from x_0. However, it is weakly maintainable since the length of executions staying in T is unbounded.

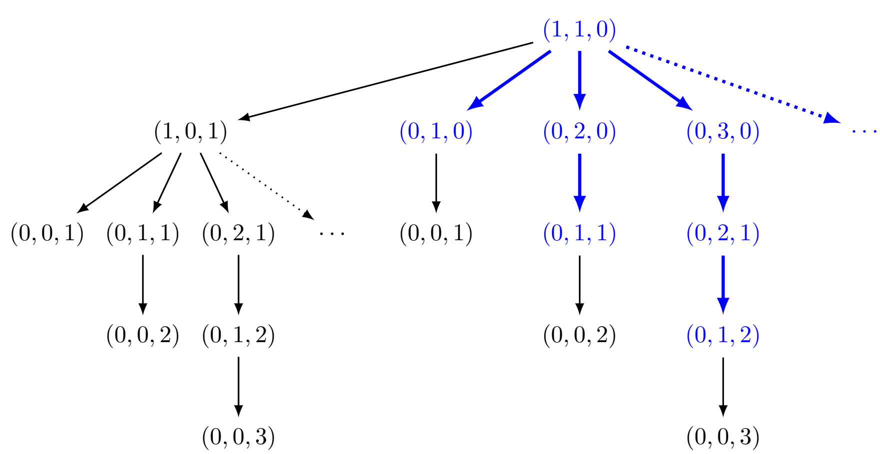

Figure 8: Reachability tree from $x_0 = (1,1,0)$ of the ω-Petri net depitcted in Fig. 6. Executions staying in $T = \uparrow(0,1,0)$ are colored in blue.

We show that, in contrast with maintainability, weak maintainability is decidable under suitable hypotheses:

Theorem 41. *Weak maintainability is decidable for any completion-post-effective class of ω^2-WSTS with strong monotonicity.*

Before proving Theorem 41, we need Prop. 42 and Prop. 43 to relate "covering" executions in a WSTS to "covering" executions in its completion.

Proposition 42. *Let $S = (X, \to, \leq)$ be a WSTS with strong monotonicity, and let $t_1, t_2, \ldots, t_n \in X$. Let $T = \uparrow\{t_1, t_2, \ldots, t_n\}$ and $U = \uparrow_{\widehat{X}} \{\downarrow t_1, \downarrow t_2, \ldots, \downarrow t_m\}$. If $I_0 \rightsquigarrow_U I_1 \rightsquigarrow_U \ldots \rightsquigarrow_U I_k$, then for every $y \in I_k$ there exists an execution $x_0 \to_T x_1 \to_T \ldots \to_T x_k$ such that $x_0 \in I_0$ and $x_k \geq y$.*

Proof. We proceed by induction on k. Assume that $k = 0$ 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 $y \leq x_0$ and $t_i \leq x_0$. Thus, $x_0 \in T$, $x_0 \in I_0$ and $x_0 \geq y$.

Assume that $k > 0$ and $I_0 \rightsquigarrow_U I_1 \rightsquigarrow_U \ldots \rightsquigarrow_U I_k$. By induction hypothesis, for every $y \in I_k$ there exists an execution $x_1 \to_T x_2 \to_T \ldots \to_T x_k$ such that $x_1 \in I_1$ and $x_k \geq y$. Since $x_1 \in I_1 \subseteq \downarrow\mathrm{Post}_S(I_0)$, there exists $x_0 \in I_0$ and $y' \geq x_1$ such that $x_0 \to y'$. 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 $x_0 \leq x_0'$ and $t_i \leq x_0'$. By strong monotonicity, there exists $x_1' \geq y'$ such that $x_0' \to x_1'$. Thus, $x_0' \to_T x_1'$. Moreover, applying strong monotonicity to $x_1 \to_T x_2 \to_T \ldots \to_T x_k$ with $x_1' \geq x_1$, we obtain an execution $x_1' \to_T x_2' \to_T \ldots \to_T x_k'$ such that $x_j' \geq x_j$ for every $1 \leq j \leq k$. Therefore, $x_0' \to_T x_1' \to_T \ldots \to_T x_k'$, $x_0' \in I_0$ and $x_k' \geq y$. $\square$

Proposition 43. *Let $S = (X, \to, \leq)$ be a WSTS and let $t_1, t_2, \ldots, t_n \in X$. Let $T = \uparrow\{t_1, t_2, \ldots, t_n\}$ and $U = \uparrow_{\widehat{X}} \{\downarrow t_1, \downarrow t_2, \ldots, \downarrow t_n\}$. If $x_0 \to_T x_1 \to_T \ldots \to_T x_k$, then for every ideal $I_0 \supseteq \downarrow x_0$ there exists an execution $I_0 \rightsquigarrow_U I_1 \rightsquigarrow_U \ldots \rightsquigarrow_U I_k$ such that $I_k \supseteq \downarrow x_k$.*

Proof. We proceed by induction on k. Assume that $k = 0$ and let I_0 be an ideal such that $I_0 \supseteq \downarrow x_0$. By hypothesis, there exists t_i such that $x_0 \geq t_i$. Thus, $I_0 \supseteq \downarrow x_0 \supseteq \downarrow t_i$, hence $I_0 \in U$.

Assume that $k > 0$ and $x_0 \to_T x_1 \to_T \ldots \to_T x_k$. Let I_0 be an ideal such that $I_0 \supseteq \downarrow x_0$, then $x_1 \in \mathrm{Post}_S(I_0) \subseteq \downarrow\mathrm{Post}_S(I_0)$. Therefore, $I_0 \rightsquigarrow I_1$ for some ideal $I_1 \supseteq \{x_1\}$. Since I_1 is downward closed, $I_1 \supseteq \downarrow x_1$. Moreover, by hypothesis there exists t_i such that $x_0 \geq t_i$. Thus, $I_0 \supseteq \downarrow x_0 \supseteq \downarrow t_i$ and we have $I_0 \in U$. By induction hypothesis, there exists an execution $I_1 \rightsquigarrow_U I_2 \rightsquigarrow_U \ldots \rightsquigarrow_U I_k$ such that $I_k \supseteq \uparrow x_k$. Therefore, $I_0 \rightsquigarrow_U I_1 \rightsquigarrow_U \ldots \rightsquigarrow_U I_k$ and $I_k \supseteq \downarrow x_k$. $\square$

We may now prove Theorem 41 from Prop. 42 and Prop. 43.

Proof of Theorem 41. Let $\mathcal{C}$ be a completion-post-effective class of ω^2-WSTS with strong monotonicity. Let $S = (X, \rightarrow, \leq) \in \mathcal{C}$, let $t_1, t_2, \ldots, t_n \in X$, $T = \uparrow\{t_1, t_2, \ldots, t_n\}$ and let $U = \uparrow_{\widehat{X}} \{\downarrow t_1, \downarrow t_2, \ldots, \downarrow t_n\}$. By Prop. 42 and Prop. 43 there exists an execution $x_0 \rightarrow_T x_1 \rightarrow_T \ldots \rightarrow_T x_k$ if, and only if, there exists an execution $I_0 \rightsquigarrow_U I_1 \rightsquigarrow_U \ldots \rightsquigarrow_U I_k$. Therefore, it suffices to verify, in $\widehat{S}$, whether $\downarrow x_0$ is weakly maintainable in U.

We know from [2] that the maintainability problem is decidable for post-effective classes of finitely branching WSTS with stuttering monotonicity. Note that even though $\widehat{\mathcal{C}}$ is such a class, we still need to adapt their algorithm to the weak maintainability problem which does not coincide.

More specifically, it suffices to build the finite reachability tree of $\widehat{S}$ from $\downarrow x_0$, and verify that it contains a maximal path labelled $I_0, I_1, \ldots, I_k$ such that $I_j \in U$ for every $0 \leq j \leq k$, and such that $I_j \subseteq I_k$ for some $0 \leq j < k$. This can be achieved since $\mathcal{C}$ is completion-post-effective, and thus, by definition, $\widehat{\mathcal{C}}$ is post-effective. □

5.4. Coverability

We now turn to coverability. Existing proofs that coverability is decidable require, in general, upward pre-effectiveness: Abdulla et al. use a backward algorithm [2, 38] that computes a finite basis of $\uparrow \mathrm{Pre}^*(\uparrow x)$ and Geeraerts et al. use a forward algorithm [12] that requires further hypotheses (*i.e.*, restriction to an adequate domain of limits, a *mathematical* hypothesis subsequently shown superfluous [39, 10]). Note that, in general, coverability becomes undecidable for classes of WSTS that are not upward pre-effective. Indeed, as stated in Theorem 15, [20] exhibits a post-effective class of finitely branching WSTS with strong and strict monotonicity for which the problem is undecidable. Moreover, upward pre-effectiveness is not the unique hypothesis ensuring decidability of coverability: for depth-bounded processes [6], a class of WSTS, upward pre-effectiveness is not satisfied and coverability is yet proved decidable by using the EEC algorithm (*i.e.* the Expand, Enlarge and Check of [39]) which is partially reformulated in the ideal completion framework of [10].

We show in the following theorem that coverability can be shown decidable under an alternative hypothesis, namely, it is decidable for completion-post-effective classes of WSTS. Even though some classes are completion-post-effective *and* upward pre-effective [20], our approach relies on evaluating Post on ideals rather than Pre on upward closed sets. Often this is more efficient, *e.g.*, it is much easier to evaluate affine functions over $\mathbb{N}_\omega^d \simeq \mathsf{Ideals}(\mathbb{N}^d)$ than inverting them.

Theorem 44. *Coverability is decidable for any completion-post-effective class of WSTS.*

Proof. Let $\mathcal{C}$ be a class of completion-post-effective $WSTS$. Let $S = (X, \rightarrow, \leq) \in \mathcal{C}$, and $x, y \in X$. We show how to decide whether y is coverable from x using two "semi-procedures". More precisely, we alternately try to determine on one hand that y is coverable, and on the other hand to determine that y is not coverable.

Let us argue that y is coverable from x in S if, and only if, $\downarrow y$ is coverable from $\downarrow x$ in $\widehat{S}$. If y is coverable from x in S, then there exists $y' \geq y$ such that $x \to^* y'$. Therefore, by Prop. 30, there exists an ideal $J \supseteq \downarrow y' \supseteq \downarrow y$ such that $\downarrow x \rightsquigarrow^* J$, hence $\downarrow y$ is coverable from $\downarrow x$ in $\widehat{S}$. Assume that $\downarrow y$ is coverable from $\downarrow x$, then there exists an ideal $J \supseteq \downarrow y$ such that $\downarrow x \rightsquigarrow^* J$. Therefore, by Prop. 29, there exists $x' \in \downarrow x$ and $y' \geq y$ such that $x' \to^* y'$. By monotonicity, $x \to^* y''$ for some $y'' \geq y'$. Since $y'' \geq y' \geq y$, y is coverable from x in S.

This way, in order to determine whether y is coverable, we iteratively build the reachability tree of $\widehat{S}$ from $\downarrow x$. More formally, we compute

$$\downarrow x,\ \mathrm{Post}_{\widehat{S}}(\downarrow x),\ \mathrm{Post}_{\widehat{S}}(\mathrm{Post}_{\widehat{S}}(\downarrow x)),\ \ldots$$

and test whether any of these sets contains an ideal J such that $J \supseteq \downarrow y$. This can be achieved since $\widehat{S}$ is post-effective.

On the other hand, we test whether y is not coverable from x in S. To do so, we note that y is not coverable from x if, and only if, there exists an inductive invariant D such that $x \in D$ and $y \notin D$. An *inductive invariant* is a downward closed set $D \subseteq X$ such that $\downarrow \mathrm{Post}_S(D) \subseteq D$. By a simple induction, we note that any inductive invariant D satisfies $\downarrow \mathrm{Post}_S^*(D) \subseteq D$. Hence, if $x \in D$ and $y \notin D$, then y is not coverable from x in S. More formally, suppose on the contrary that y is coverable, then $y \in \downarrow \mathrm{Post}_S^*(x) \subseteq \downarrow \mathrm{Post}_S^*(D) \subseteq D$, which is a contradiction. Moreover, if y is not coverable from x in S, $\downarrow \mathrm{Post}_S^*(x)$ is such an inductive invariant. Indeed, let us see that $\downarrow \mathrm{Post}_S(\downarrow \mathrm{Post}_S^*(x)) \subseteq \downarrow \mathrm{Post}_S^*(x)$. Let $z \in \downarrow \mathrm{Post}_S(\downarrow \mathrm{Post}_S^*(x))$, there exist $x, y, y', z' \in X$ such that $x \to^* y$, $y \geq y'$, $y' \to^* z'$ and $z' \geq z$, hence, by monotonicity, there exists $z'' \geq z' \geq z$ such that $x \to^* z''$ and therefore $z \in \downarrow \mathrm{Post}_S^*(x)$.

Thus, in order to ascertain that y is not coverable from x in S, it suffices to enumerate inductive invariants D and to test whether $x \in D$ and $y \notin D$. This is made possible by post-effectiveness of $\widehat{S}$. Downward closed sets may be enumerated by their ideal decomposition. In order to test whether D is an inductive invariant, we use the following observation, where the last equivalence follows from Prop. 20:

$$\begin{aligned}
\downarrow \mathrm{Post}_S(D) \subseteq D \iff & \bigcup_{I \in \mathsf{IdealDecomp}(D)} \downarrow \mathrm{Post}_S(I) \ \subseteq \bigcup_{I \in \mathsf{IdealDecomp}(D)} I \\
\iff & \bigcup_{I \in \mathsf{IdealDecomp}(D)} \ \bigcup_{J \in \mathrm{Post}_{\widehat{S}}(I)} J \ \subseteq \bigcup_{I \in \mathsf{IdealDecomp}(D)} I \\
\iff & \forall J \in \mathrm{Post}_{\widehat{S}}(\mathsf{IdealDecomp}(D)), \\
& \qquad \exists I \in \mathsf{IdealDecomp}(D) \ \text{ s.t. } J \subseteq I\ . \qquad (4)
\end{aligned}$$

Since $\widehat{S}$ is post-effective, it is possible to compute $\mathrm{Post}_{\widehat{S}}$, to test ideal inclusion, and therefore to test (4). $\square$

Unfortunately, not all classes of WSTS for which coverability is decidable are completion-post-effective. For completeness, we exhibit a class of WSTS that is neither completion-post-effective, nor upward pre-effective, and yet has its coverability problem decidable. This class, that we will name $\mathcal{F}_1$, is inspired by the more general, but non effective, class of so-called well-structured nets of dimension 1 [20].

We define $\mathcal{F}_1$ as follows. The i^{th} WSTS S of the class, $S = (\mathbb{N}, \rightarrow, \leq)$, is associated to a finite collection of Turing machines $M_1, M_2, \ldots, M_n$, and is a functional WSTS defined by the finite set of non decreasing functions $F = \{f_1, f_2, \ldots, f_n\}$ where each f_j is defined as follows,

$$f_j(x) = \begin{cases} 0 & \text{if } x = 0, \\ f(x-1) & \text{if } x > 0 \text{, and if } M_j \text{, on input } x \in \mathbb{N} \text{, does not halt} \\ & \text{in at most } x \text{ steps,} \\ f(x-1) + y & \text{if } x > 0 \text{, and if } M_j \text{, on input } x \in \mathbb{N} \text{, halts in at} \\ & \text{most } x \text{ steps, and returns } y. \end{cases}$$

Since $\mathcal{F}_1$ is a class of functional WSTS over $\mathbb{N}$, and since each f_j can be computed by executing M_j for a finite amount of time, it is readily seen that $\mathcal{F}_1$ forms a post-effective class of finitely branching ω^2-WSTS with strong monotonicity and well partial ordering. However, $\mathcal{F}_1$ is neither completion-effective, nor upward pre-effective.

Proposition 45. *$\mathcal{F}_1$ is not completion-effective or upward pre-effective.*

Proof. Let Turing'_i be a Turing machine that executes Turing_i and returns 1. Let $S_i = (\mathbb{N}, \rightarrow, \leq)$ be the WSTS of $\mathcal{F}_1$ associated to Turing'_i. If Turing_i does not halt on its encoding, then Turing'_i never halts, hence $\mathrm{Post}_{S_i}(x) = \{0\}$ for every $x \in \mathbb{N}$. If Turing_i halts on its encoding, then there exists $m \in \mathbb{N}$ such that $\mathrm{Post}_{S_i}(x) = \{0\}$ for every $x < m$ and $\mathrm{Post}_{S_i}(x) = \{x - m + 1\}$ for every $x \geq m$. By definition of S_i and $\widehat{S_i}$, we have

$$\begin{aligned} \mathbb{N} \rightsquigarrow \mathbb{N} \quad & \Longleftrightarrow \quad \mathbb{N} \subseteq \downarrow \mathrm{Post}_{S_i}(\mathbb{N}) \\ & \Longleftrightarrow \quad \mathrm{Post}_{S_i}(\mathbb{N}) = \mathbb{N} \\ & \Longleftrightarrow \quad \mathsf{Turing}'_i \text{ outputs 1 for infinitely many inputs } x \in \mathbb{N} \\ & \Longleftrightarrow \quad \mathsf{Turing}_i \text{ halts on its encoding.} \end{aligned}$$

If $\mathcal{F}_1$ was completion-effective, we could produce $\mathsf{IdealDecomp}(\overline{\emptyset}) = \mathbb{N}$ and verify whether $\mathbb{N} \rightsquigarrow \mathbb{N}$. This would yield a procedure to decide the halting problem, which is impossible. Therefore, $\mathcal{F}_1$ is not completion-effective.

By definition of S_i, we also have that $\uparrow \mathrm{Pre}_{S_i}(\uparrow 1) \neq \emptyset$ if, and only if, Turing_i halts on its encoding. Thus, if $\mathcal{F}_1$ was upward pre-effective, it would be possible to decide whether $\uparrow \mathrm{Pre}_{S_i}(\uparrow 1) \neq \emptyset$ and hence to decide whether Turing_i halts on its encoding, which is impossible. Therefore, $\mathcal{F}_1$ is not upward pre-effective. $\square$

Even though $\mathcal{F}_1$ is neither completion-post-effective, nor upward pre-effective, its coverability problem can still be shown decidable.

Proposition 46. *Coverability for $\mathcal{F}_1$ is decidable.*

Proof. Let $S = (\mathbb{N}, \rightarrow, \leq) \in \mathcal{F}_1$ be the functional WSTS whose transition relation is defined by the finite set of non decreasing functions $F = \{f_1, f_2, \ldots, f_n\}$. Let $x_0, x \in \mathbb{N}$. First note that if $x_0 \geq x$, then x is coverable from x_0 and we are done. Therefore, assume that $x_0 < x$.

Let $F^{\leq \ell} = \{f_{i_1} \circ f_{i_2} \circ \cdots \circ f_{i_k} : 1 \leq i_1, i_2, \ldots, i_k \leq n, k \leq \ell\}$ and let $m_\ell = \max\{g(x_0) : g \in F^{\leq \ell}\}$. Consider the infinite sequence $m_0, m_1, \ldots$, let us prove that if two consecutive terms coincide, then the sequence stabilizes. Assume that $m_{\ell+1} = m_\ell$ for some $\ell \in \mathbb{N}$, then by definition $f_i(m_\ell) \leq m_\ell$ for every $i \in [n]$, hence by monotonicity and by induction on p, we may prove that $g(m_\ell) \leq m_\ell$ for every $g \in F^{\leq p}$. Thus, $m_{\ell+p} \leq m_\ell$ for every $p \geq 0$. By construction, $m_{\ell+p} \geq \ell_n$, hence $m_{\ell+p} = m_\ell$ and the sequence stabilizes.

Since the sequence strictly increases until it stabilizes, it suffices to compute m_{x-x_0}. If $m_{x-x_0} \geq x$, then x is coverable from x_0, otherwise it is not. □

What is nevertheless encouraging is that most known useful classes of WSTS can be shown to be completion-post-effective; see [10] for a hierarchy of datatypes, containing integers and words, the cartesian product, the multiset and the concatenation operators and allowing also trees, that permits the construction of various such classes of WSTS. Moreover, [10] shows how the ideals associated with these hierarchy of datatypes can be effectively manipulated.

It is worth mentioning that 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 [40, Corollary 9.6]. More recently, Raskin et al. [12, 39] also used similar enumeration of inductive invariants to provide forward algorithms for deciding coverability in WSTS. However, their techniques use algebraic and additional effectiveness hypotheses, while we appeal to completion-post-effectiveness.

6. Conclusion and further work

In this paper we have extended the decidability results of finitely branching WSTS properties to the case of infinitely branching WSTS. We have also further completed the picture of the decidability status of the four main computational problems (termination, boundedness, coverability and maintainability) of interest for WSTS. This is depicted in Table 1. To make this extension, we have used the completion of well-structured transition systems and along the way, we have simplified the presentation currently found in the literature and we have also extended it to general (*i.e.*, non functional) WSTS. Moreover, we have then established the precise connection between executions in a WSTS and executions in its completion, for both finitely and infinitely branching WSTS obeying an exhaustive list of different monotonicity conditions. The completion of infinitely branching WSTS by means of ideals can be done for ω^2-WSTS and this confirms the central place of ω^2-WSTS among general WSTS.

The theory developed led not only to new decidability results but to reproving some of the well known results in a uniform way. In particular, a new (forward) algorithm was shown to solve the coverability problem for completion-post-effective classes of infinitely branching WSTS (hence fulfilling a simpler condition than the usual upward pre-effectiveness found in the literature).

<table>
<tr><th></th><th>Finitely branching</th><th>Infinitely branching</th></tr>
<tr><td rowspan="2">Termination</td><td>Dec. • post-effective
• transitive monotonicity</td><td rowspan="2">Undec. • post-effective
• completion-post-effective
• strong and strict monotonicity
• ω^2-WSTS
• well partial ordering</td></tr>
<tr><td>Undec. • post-effective
• ω^2-WSTS
• well partial ordering</td></tr>
<tr><td rowspan="2">Strong termination</td><td>Dec. • post-effective
• transitive monotonicity</td><td>Dec. • completion-post-effective
• transitive monotonicity
• ω^2-WSTS</td></tr>
<tr><td colspan="2">Undec. • post-effective
• ω^2-WSTS
• well partial ordering</td></tr>
<tr><td rowspan="2">Boundedness</td><td colspan="2">Dec. • post-effective
• strict monotonicity
• well partial ordering</td></tr>
<tr><td colspan="2">Undec. • post-effective
• strong monotonicity
• well partial ordering</td></tr>
<tr><td rowspan="2">Maintainability</td><td>Dec. • post-effective
• stuttering monotonicity</td><td rowspan="2">Undec. • post-effective
• completion-post-effective
• strong and strict monotonicity
• ω^2-WSTS
• well partial ordering</td></tr>
<tr><td>Undec. • post-effective
• ω^2-WSTS
• well partial ordering</td></tr>
<tr><td rowspan="2">Weak maintainability</td><td>Dec. • post-effective
• stuttering monotonicity</td><td>Dec. • completion-post-effective
• strong monotonicity
• ω^2-WSTS</td></tr>
<tr><td colspan="2">Undec. • post-effective
• ω^2-WSTS
• well partial ordering</td></tr>
<tr><td rowspan="3">Coverability</td><td colspan="2">Dec. • upward pre-effective</td></tr>
<tr><td colspan="2">Dec. • completion-post-effective</td></tr>
<tr><td colspan="2">Undec. • post-effective
• strong and strict monotonicity
• ω^2-WSTS
• well partial ordering</td></tr>
</table>

Table 1: Decidability (in green) or undecidability (in red) of WSTS classes problems. Darker and hached regions identify results new to this paper. See Theorem 15 for references to the sources of results known prior to this paper.

We have uncovered a subtlety that arises in the study of the termination problem for WSTS when the finitely branching hypothesis is dropped: such WSTS having no infinite executions can nonetheless have executions of every finite length. This led to a distinction between weak and strong termination, and to a similar distinction between weak and strong maintainability.

The preliminary version of the present paper first appeared at ICALP 2014. Since then, much recent research [41, 34, 42, 43, 44, 45] was devoted to exploiting the finite decomposition of downward closed sets into a finite set of ideals. We believe that our simple presentation of the theory underlying this decomposition has made applying the latter possible.

We now turn to perspectives and further work. There are other interesting properties (than the four studied here) to decide for infinitely branching WSTS; for instance, being able to decide whether there exists a simulation between an infinitely branching WSTS and a finite automaton would simplify the verification of certain WSTS.

Our treatment of infinitely branching WSTS has unified several results and proofs dealing with parameterized systems based on Petri nets, extended Petri nets with resets and transfers, and affine nets. Future work should consider applying these results to other general parameterized WSTS.

Recursive-parallel systems are shown to be WSTS in [18], for an ordering which uses the multiset ω^2-ordering; this suggests that recursive-parallel systems could also belong to the family of ω^2-WSTS. The completion of ω^2-WSTS also opens the way for designing new forward algorithms. Such algorithms are conceptually simpler than backward algorithms and may in some respects be more efficient as well. It will be interesting to further investigate these algorithmic aspects and, in particular, to see how the efficiencies of backward and forward strategies compare when applied to concrete models.

Acknowledgements. We thank Jean Goubault-Larrecq for allowing us to include his proof of Theorem 21. We would also like to thank the anonymous referees for their thoughtful and critical comments that triggered a partial overhaul of the presentation of our paper.

[1] A. Finkel, Reduction and covering of infinite reachability trees, Information and Computation 89 (2) (1990) 144–179. doi:10.1016/0890-5401(90)90009-7.

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

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

[4] N. Bertrand, G. Delzanno, B. König, A. Sangnier, J. Stückrath, On the decidability status of reachability and coverability in graph transformation systems, in: Proc. 23rd International Conference on Rewriting Techniques and Applications (RTA), 2012, pp. 101–116. doi:10.4230/LIPIcs.RTA.2012.101.

[5] B. König, J. Stückrath, Well-structured graph transformation systems with negative application conditions, in: Proc. 6th International Conference Graph Transformations (ICGT), 2012, pp. 81–95. doi:10.1007/978-3-642-33654-6_6.

[6] T. Wies, D. Zufferey, T. A. Henzinger, Forward analysis of depth-bounded processes, in: Proc. 13th International Conference on Foundations of Software Science and Computational Structures (FOSSACS), 2010, pp. 94–108. doi:10.1007/978-3-642-12032-9_8.

[7] D. Zufferey, T. Wies, T. A. Henzinger, Ideal abstractions for well-structured transition systems, in: Proc. 13th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2012, pp. 445–460. doi:10.1007/978-3-642-27940-9_29.

[8] J. Esparza, A. Finkel, R. Mayr, On the verification of broadcast protocols, in: Proc. 14th Annual IEEE Symposium on Logic in Computer Science (LICS), 1999, pp. 352–359. doi:10.1109/LICS.1999.782630.

[9] A. Kaiser, D. Kroening, T. Wahl, Efficient coverability analysis by proof minimization, in: Proc. 23rd International Conference on Concurrency Theory (CONCUR), 2012, pp. 500–515. doi:10.1007/978-3-642-32940-1_35.

[10] A. Finkel, J. Goubault-Larrecq, Forward analysis for WSTS, part I: completions, in: Proc. 26th International Symposium on Theoretical Aspects of Computer Science (STACS), 2009, pp. 433–444. doi:10.4230/LIPIcs.STACS.2009.1844.

[11] A. Finkel, J. Goubault-Larrecq, Forward analysis for WSTS, part II: complete WSTS, in: Proc. 36th Internatilonal Colloquium on Automata, Languages and Programming (ICALP), 2009, pp. 188–199. doi:10.1007/978-3-642-02930-1_16.

[12] G. Geeraerts, J. Raskin, L. V. Begin, Expand, enlarge and check: New algorithms for the coverability problem of WSTS, Journal of Computer and System Sciences 72 (1) (2006) 180–203. doi:10.1016/j.jcss.2005.09.001.

[13] S. Schmitz, Ph. Schnoebelen, The power of well-structured systems, in: Proc. 24th International Conference Concurrency Theory (CONCUR), 2013, pp. 5–24. doi:10.1007/978-3-642-40184-8_2.

[14] N. Bertrand, Ph. Schnoebelen, Computable fixpoints in well-structured symbolic model checking, Formal Methods in System Design 43 (2) (2013) 233–267. doi:10.1007/s10703-012-0168-y.

[15] S. Schmitz, Ph. Schnoebelen, Multiply-recursive upper bounds with higman's lemma, in: Proc. 38th International Colloquium on Automata, Languages and Programming (ICALP), 2011, pp. 441–452. doi:10.1007/978-3-642-22012-8_35.

[16] G. Cécé, A. Finkel, S. P. Iyer, Unreliable channels are easier to verify than perfect channels, Information and Computation 124 (1) (1996) 20–31. doi:10.1006/inco.1996.0003.

[17] P. Bouyer, N. Markey, J. Ouaknine, Ph. Schnoebelen, J. Worrell, On termination and invariance for faulty channel machines, Formal Aspects of Computing 24 (4-6) (2012) 595–607. doi:10.1007/s00165-012-0234-7.

[18] O. Kouchnarenko, Ph. Schnoebelen, A model for recursive-parallel programs, Electronic Notes in Theoretical Computer Science 5 (1996) 30. doi:10.1016/S1571-0661(05)82512-5.

[19] G. Geeraerts, A. Heußner, M. Praveen, J. Raskin, ω-Petri nets, in: Proc. 34th International Conference on Application and Theory of Petri Nets and Concurrency (ICATPN), 2013, pp. 49–69. doi:10.1007/978-3-642-38697-8_4.

[20] A. Finkel, P. McKenzie, C. Picaronny, A well-structured framework for analysing Petri net extensions, Information and Computation 195 (1-2) (2004) 1–29. doi:10.1016/j.ic.2004.01.005.

[21] G. V. Bochmann, Finite state description of communication protocols, Computer Networks (1976) 2 (4) (1978) 361–372. doi:10.1016/0376-5075(78)90015-6.

[22] D. Brand, P. Zafiropulo, On communicating finite-state machines, Journal of the ACM 30 (2) (1983) 323–342. doi:10.1145/322374.322380.

[23] P. A. Abdulla, B. Jonsson, Verifying programs with unreliable channels, in: Proc. 8th Annual Symposium on Logic in Computer Science (LICS), 1993, pp. 160–170. doi:10.1109/LICS.1993.287591.

[24] C. Dufourd, A. Finkel, Ph. Schnoebelen, Reset nets between decidability and undecidability, in: Proc. 25th International Colloquium on Automata, Languages and Programming (ICALP), 1998, pp. 103–115. doi:10.1007/BFb0055044.

[25] J. Rogers, J. Heinz, G. Bailey, M. Edlefsen, M. Visscher, D. Wellcome, S. Wibel, On languages piecewise testable in the strict sense, in: Revised Selected Papers of the 10th and 11th Biennial Conference on The Mathematics of Language, 2010, pp. 255–265. doi:10.1007/978-3-642-14322-9_19.

[26] P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, B. Jonsson, Using forward reachability analysis for verification of lossy channel systems, Formal Methods in System Design 25 (1) (2004) 39–65. doi:10.1023/B:FORM.0000033962.51898.1a.

[27] J. Goubault-Larrecq, personal communication (October 2013).

[28] J. Goubault-Larrecq, P. Karandikar, K. N. Kumar, The ideal approach to computing closed subsets in well-quasi-orderings, in preparation (2015).

[29] P. Erdős, A. Tarski, On families of mutually exclusive sets, Annals of Mathematics 2 (44) (1943) 315–329. doi:10.2307/1968767.

[30] R. Bonnet, On the cardinality of the set of initial intervals of a partially ordered set, Infinite and finite sets: to Paul Erdős on his 60th birthday (1975) 189–198.

[31] R. Fraïssé, Theory of relations, Studies in Logic and the Foundations of Mathematics 118 (1986) 1–456. doi:10.1016/S0049-237X(08)70963-7.

[32] P. Jančar, A note on well quasi-orderings for powersets, Information Processing Letters 72 (5-6) (1999) 155–160. doi:10.1016/S0020-0190(99)00149-0.

[33] R. Rado, Partial well-ordering of sets of vectors, Mathematika 1 (1954) 89–95. doi:10.1112/S0025579300000565.

[34] J. Leroux, S. Schmitz, Demystifying reachability in vector addition systems, in: Proc. 30$^{\text{th}}$ Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2015, pp. 56–67. doi:10.1109/LICS.2015.16.

[35] P. A. Abdulla, A. Bouajjani, B. Jonsson, On-the-fly analysis of systems with unbounded, lossy FIFO channels, in: Proc. 10$^{\text{th}}$ International Conference on Computer Aided Verification (CAV), 1998, pp. 305–318. doi:10.1007/BFb0028754.

[36] R. Mayr, Undecidable problems in unreliable computations, in: Proc. 4$^{\text{th}}$ Latin American Symposium on Theoretical Informatics (LATIN), 2000, pp. 377–386. doi:10.1007/10719839_37.

[37] C. Dufourd, P. Jančar, Ph. Schnoebelen, Boundedness of reset P/T nets, in: Proc. 26$^{\text{th}}$ International Colloquium on Automata, Languages and Programming (ICALP), 1999, pp. 301–310. doi:10.1007/3-540-48523-6_27.

[38] P. A. Abdulla, K. Cerans, B. Jonsson, Y. Tsay, General decidability theorems for infinite-state systems, in: Proc. 11$^{\text{th}}$ Annual IEEE Symposium on Logic in Computer Science (LICS), 1996, pp. 313–321. doi:10.1109/LICS.1996.561359.

[39] P. Ganty, J. Raskin, L. V. Begin, A complete abstract interpretation framework for coverability properties of WSTS, in: Proc. 7$^{\text{th}}$ International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2006, pp. 49–64. doi:10.1007/11609773_4.

[40] J. K. Pachl, Reachability problems for communicating finite state machines, Tech. Rep. CS-82-12, University of Waterloo (1982).

[41] R. Lazić, S. Schmitz, The ideal view on rackoff's coverability technique, in: Proc. 9th International Workshop on Reachability Problems (RP), 2015, pp. 76–88. doi:10.1007/978-3-319-24537-9_8.

[42] R. Lazić, S. Schmitz, The complexity of coverability in ν-Petri nets, in: Proc. 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2016, pp. 467–476. doi:10.1145/2933575.2933593.

[43] P. Hofman, S. Lasota, R. Lazić, J. Leroux, S. Schmitz, P. Totzke, Coverability trees for Petri nets with unordered data, in: Proc. 19th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS), 2016, pp. 445–461. doi:10.1007/978-3-662-49630-5_26.

[44] J. Leroux, S. Schmitz, Ideal decompositions for vector addition systems, in: Proc. 33rd Symposium on Theoretical Aspects of Computer Science (STACS), 2016, pp. 1:1–1:13. doi:10.4230/LIPIcs.STACS.2016.1.

[45] J. Goubault-Larrecq, S. Schmitz, Deciding piecewise testable separability for regular tree languages, in: Proc. 43rd International Colloquium on Automata, Languages, and Programming (ICALP), 2016, pp. 97:1–97:15. doi:10.4230/LIPIcs.ICALP.2016.97.