Expand, Enlarge, and Check

New algorithms for the coverability problem of WSTS

Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin

DI, Université Libre de Bruxelles

Abstract. In this paper, we present a general algorithmic schema called "Expand, Enlarge and Check" from which new efficient algorithms for the coverability problem of WSTS can be constructed. We show here that our schema allows us to define forward algorithms that decide the coverability problem for several classes of systems for which the Karp and Miller procedure cannot be generalized, and for which no complete forward algorithms were known. Our results have important applications for the verification of parameterized systems and communication protocols.

1 Introduction

Model-checking is nowadays widely accepted as a powerful technique for the automatic verification of reactive systems that have natural finite state abstractions. However, many reactive systems are only naturally modeled as infinite-state systems. Consequently, a large (and successful) research effort has recently focused on the direct application of model-checking techniques to infinite-state models such as timed automata [5], hybrid automata [19], FIFO channel systems [3, 2, 4], Petri nets [10, 7], broadcast protocols [14], etc.

One of the positive results is the decidability of the *coverability problem*[1] for *well-structured transition systems* (WSTS for short). WSTS enjoy an infinite set of states that is well-quasi ordered by $\leq$ and their transition relation is monotonic w.r.t $\leq$. Examples of such systems are Petri nets and their monotonic extensions [23, 9, 11], broadcast protocols [13], lossy channel systems [3]. The *coverability problem* asks, given two states c_1 and c_2, whether there is $c_3 \geq c_2$ (c_3 covers c_2) that is reachable from c_1.

A general algorithm (i.e. a procedure that always terminates) is known to solve the coverability problem for WSTS [1, 17]. It symbolically manipulates upward-closed sets of states, obtained by unrolling the transition relation in a *backward* fashion. Unfortunately, backward search is seldom efficient in practice [20], and the only complete forward approach known so far is the Karp and Miller algorithm that can only be applied to a small subclass of WSTS: Petri nets.

The Karp and Miller procedure computes, through a combination of a forward exploration strategy and a simple acceleration technique, the so-called *covering set* of the net, which is known to be well-suited to decide the coverability problem. After several attempts to generalize this procedure to WSTS (which have all produced incomplete approaches [15, 13, 16]), it has been shown in [12] that Petri nets form the sole class (among the examples cited above) for which the covering set is constructible in general. However, this set always exists and is usually finitely representable. Our *main contribution* is to make the best of this fact and devise a forward technique that is complete to decide the coverability problem for a large class of WSTS. This

[1] A large class of *safety verification problems* can be reduced to the coverability problem.

class includes, among others, all the monotonic extensions of Petri nets defined in the literature, as well as lossy channel systems.

We present a new schema of algorithm: "Expand, Enlarge and Check" that works by iteratively constructing more and more precise abstractions of the system. These abstractions (made up of reachable states and limit elements) are guaranteed to become precise enough to decide the coverability problem after a finite number of steps. We show how to apply the schema on two classes of WSTS of practical interest: monotonic extensions of Petri nets (that are useful to model parameterized systems [18, 11]) and lossy channels systems (that are useful to model communication protocols [2]). Besides giving the opportunity to define efficient and complete forward algorithms for a large class of WSTS, the abstractions that we define in this paper can also be used to obtain semi-decision procedures for checking more complex properties, like model-checking of LTL formulas.

2 Preliminaries

In this section, we recall some fundamental results about *well-quasi orderings* and *well-structured transition systems* (the systems we analyze here). We show how to *finitely* represent upward- and downward-closed sets of states (which will allow us to devise *symbolic* algorithms), and discuss And-Or graphs (useful to represent abstractions of systems).

Well quasi-orderings and adequate domains of limits A *well quasi ordering* $\leq$ on the elements of a set C (wqo for short) is a *reflexive* and *transitive* relation such that for any infinite sequence $c_0 c_1 \dots c_n \dots$ of elements in C, there exist two indices i and j, such that $i < j$ and $c_i \leq c_j$. In the following, we note $c_i < c_j$ if $c_i \leq c_j$ but $c_j \not\leq c_i$.

Let $\langle C, \leq \rangle$ be a well-quasi ordered set. A $\leq$*-upward closed set* $U \subseteq C$ is such that for any $c \in U$, for any $c' \in C$ such that $c \leq c'$, $c' \in U$. A $\leq$*-downward closed set* $D \subseteq C$ is such that for any $c \in D$, for any $c' \in C$ such that $c' \leq c$, $c' \in D$. It is well-known that any $\leq$-upward closed set $U \subseteq C$ is uniquely determined by its finite sets of minimal elements. Formally, a set of $\leq$*-minimal* elements $\mathsf{Min}(U)$ of a set $U \subseteq C$ is a minimal set such that $\mathsf{Min}(U) \subseteq U$ and $\forall s' \in U : \exists s \in \mathsf{Min}(U) : s \leq s'$. The following proposition is a direct consequence of wqo:

Proposition 1. *Let $\langle C, \leq \rangle$ be a wqo set and $U \subseteq C$ be an $\leq$-upward closed set, then:* $\mathsf{Min}(U)$ *is finite and* $U = \{c \mid \exists c' \in \mathsf{Min}(U) : c' \leq c\}$.

Thus, any $\leq$-upward closed set can be *effectively represented* by its finite set of minimal elements. Downward-closed sets are more difficult to represent effectively. To obtain a finite representation of those sets, we must use well-chosen limit elements $\ell \notin C$ to represent downward closures of infinite increasing chains of elements. Thus, we introduce the notion of *adequate* domain of limits.

Definition 1. Let $\langle C, \leq \rangle$ be a well-quasi ordered set and L be a set of elements disjoint from C, the tuple $\langle L, \sqsubseteq, \gamma \rangle$ is called an *adequate domain of limits* for $\langle C, \leq \rangle$ if the following conditions are satisfied: (L_1: representation mapping) $\gamma : L \cup C \to 2^C$ associates to each element in $L \cup C$ a $\leq$-downward closed set $D \subseteq C$, furthermore, for any $c \in C$, we impose that $\gamma(c) = \{c' \mid c' \leq c\}$. In the following, γ is extended to sets $S \subseteq L \cup C$ in the natural way: $\gamma(S) = \cup_{c \in S} \gamma(c)$; ($\mathsf{L}_2$: top element) There exists a special element $\top \in L$ such that $\gamma(\top) = C$; (L_3: precision order) The elements of $C \cup L$ are ordered by the complete quasi order $\sqsubseteq$, defined as follows: $d_1 \sqsubseteq d_2$ if and only if $\gamma(d_1) \subseteq \gamma(d_2)$; ($\mathsf{L}_4$: completeness) for any downward closed set $D \subseteq C$, there exists a finite set $D' \subseteq C \cup L$ such that $\gamma(D') = D$.

Well-structured transition systems and coverability problem A *transition system* is a tuple $S = \langle C, c_0, \rightarrow\rangle$ where C is a (possibly infinite) set of states, $c_0 \in C$ is the initial state, $\rightarrow \subseteq C \times C$ is a transition relation. In the following, $c \rightarrow c'$ will denote that $\langle c, c'\rangle \in \rightarrow$. For any state c, $\mathsf{Post}(c)$ denotes the set of one-step successors of c, i.e. $\mathsf{Post}(c) = \{c' | c \rightarrow c'\}$. This operator is extended to sets of states $C' \subseteq C$ as follows: $\mathsf{Post}(C') = \{c | \exists c' \in C' : c' \rightarrow c\}$. A *path* of S is a sequence of states $c_1, c_2, \ldots, c_k$ such that $c_1 \rightarrow c_2 \rightarrow \cdots \rightarrow c_k$. A state c' is reachable from a state c, noted $c \rightarrow^* c'$, if we have a path $c_1, c_2, \ldots c_k$ in S with $c_1 = c$ and $c_k = c'$. Given a transition system $S = \langle C, c_0, \rightarrow\rangle$, $\mathsf{Reach}(S)$ denotes the set $\{c \in C \mid c_0 \rightarrow^* c\}$. Finally, we require a transition system to be without deadlock states[2]. That is, for any state $c \in C$, there exists $c' \in C$ such that $c \rightarrow c'$.

Definition 2. A transition system $S = \langle C, c_0, \rightarrow\rangle$ is a *well-structured transition system* for the quasi order $\leq \subseteq C \times C$ if the two following properties hold: (W_1: well-ordering) $\leq$ is a well-quasi ordering and (W_2: monotonicity) for all $c_1, c_2, c_3 \in C$ such that $c_1 \leq c_2$ and $c_1 \rightarrow c_3$, there exists $c_4 \in C$ such that $c_3 \leq c_4$ and $c_2 \rightarrow c_4$.

From now on, $S = \langle C, c_0, \rightarrow, \leq\rangle$ will denote the well-structured transition system $\langle C, c_0, \rightarrow\rangle$ for $\leq$. In the sequel, we need to manipulate WSTS and adequate domain of limits. In particular, we need the following effectiveness properties:

Definition 3. A WSTS $S = \langle C, c_0, \rightarrow, \leq\rangle$ and an adequate domain of limits $\langle L, \sqsubseteq, \gamma\rangle$ are *effective* if the following conditions are satisfied: (E_1) C and L are recursively enumerable; (E_2) for any $c_1, c_2 \in C$, we can decide whether $c_1 \rightarrow c_2$; (E_3) for any two finite subsets $C' \subseteq C$ and $L' \subseteq L$, for any $d \in C' \cup L'$ and any finite subset $D \subseteq C' \cup L'$, we can decide whether $\mathsf{Post}(\gamma(d)) \subseteq \gamma(D)$; ($\mathsf{E}_4$) For any finite subsets $D_1, D_2 \subseteq C \cup L$, we can decide whether $\gamma(D_1) \subseteq \gamma(D_2)$.

Problem 1. The *coverability problem for well-structured transition systems* is defined as follows: "Given a well-structured transition system S and the $\leq$-upward closed set $U \subseteq C$, determine whether $\mathsf{Reach}(S) \cap U = \emptyset$?"

To solve the coverability problem, we use covering sets, defined as follows:

Definition 4. Let $S = \langle C, c_0, \rightarrow, \leq\rangle$ be a WSTS. The *covering set* of S, noted $\mathsf{Cover}(S)$, is the (unique) smallest subset of C which (CS_1) is $\leq$-downward closed and (CS_2) contains $\mathsf{Reach}(S)$.

Property For any WSTS $S = \langle C, c_0, \rightarrow, \leq\rangle$ with an adequate domain of limits $\langle L, \sqsubseteq, \gamma\rangle$ for $\langle C, \leq\rangle$, by property L_4 of Definition 1, there exists a finite subset $\mathsf{CS}(S) \subseteq L \cup C$ such that $\gamma(\mathsf{CS}(S)) = \mathsf{Cover}(S)$. In the following, $\mathsf{CS}(S)$ is called a *coverability set* of the covering set $\mathsf{Cover}(S)$ and it is a finite representation of that set.

Proposition 2 ([15]). *For any* WSTS $S = \langle C, c_0, \rightarrow, \leq\rangle$*, the covering set of* S *is such that for any* $\leq$*-upward closed set* $U \subseteq C$*:* $\mathsf{Reach}(S) \cap U = \emptyset$ *iff* $\mathsf{Cover}(S) \cap U = \emptyset$*.*

[2] Note that this condition is not restrictive since we can always add a transition to a dummy state.

And-Or graph and its avoidability problem An *And-Or graph* is a tuple $G = \langle V_A, V_O, v_i, \Rightarrow\rangle$ where $V = V_A \cup V_O$ is the set of nodes (V_A is the set of "And" nodes and V_O is the set of "Or" nodes), $V_A \cap V_O = \emptyset$, $v_i \in V_O$ is the initial node, and $\Rightarrow \subseteq (V_A \times V_O) \cup (V_O \times V_A)$ is the transition relation such that for any $v \in V_A \cup V_O$, there exists $v' \in V_A \cup V_O$ such that $(v, v') \in \Rightarrow$.

Definition 5. A *compatible unfolding* of an And-Or graph $G = \langle V_A, V_O, v_i, \Rightarrow\rangle$ is an infinite labelled tree $T_G = \langle N, \mathit{root}, B, \Lambda\rangle$ where: (*i*) N is the set of nodes of T_G, (*ii*) $\mathit{root} \in N$ is the root of T_G, (*iii*) $B \subseteq N \times N$ is the transition relation of T_G, (*iv*) $\Lambda : N \to V_A \cup V_0$ is the labelling function of the nodes of T_G by nodes of G that respects the three following compatibility conditions (Λ is extended to sets of nodes in the usual way): (C_1)$\Lambda(\mathit{root}) = v_i$; ($\mathsf{C}_2$) for all $n \in N$ such that $\Lambda(n) \in V_A$, we have that (*a*) for all nodes $v' \in V_O$ such that $\Lambda(n) \Rightarrow v'$, there exists one and only one $n' \in N$ such that $B(n, n')$ and $\Lambda(n') = v'$, and conversely (*b*) for all nodes $n' \in N$ such that $B(n, n')$, there exists $v' \in V_O$ such that $\Lambda(n) \Rightarrow v'$ and $\Lambda(n') = v'$. (C_3) for all $n \in N$ such that $\Lambda(n) \in V_O$, we have that: there exists one and only one $n' \in N$ such that $B(n, n')$, and $\Lambda(n) \Rightarrow \Lambda(n')$.

Problem 2. The *And-Or Graph Avoidability Problem* is defined as follows: "Given an And-Or graph $G = \langle V_A, V_O, v_i, \Rightarrow\rangle$ and a set $E \subseteq V_A \cup V_O$, does there exist $T = \langle N, \mathit{root}, \Lambda, B\rangle$, a compatible unfolding of G such that $\Lambda(N) \cap E = \emptyset$?". When the answer is positive, we say that E is *avoidable* in G.

It is well-known that this problem is complete for *PTIME* [21].

3 A new schema of algorithms

In this section, we introduce our new schema of algorithms to decide the coverability problem for WSTS. We first explain, in subsection 3.1, how to build an *abstraction* of a given WSTS, w.r.t. a given finite set of reachable states $C' \subseteq C$ and a given finite set of limit elements $L' \subseteq L$. These abstractions are *And-Or graphs* whose nodes are annotated by downward-closed sets of states of a WSTS. We show in subsection 3.2 that any unfolding of this And-Or graph is able to *simulate* the behaviours of its associated WSTS (Proposition 3). Moreover, if the downward-closed sets that are used to annotate the And-Or graph are *precise enough* (in a sense that we make clear in Theorem 2), then the And-Or graph can be used to decide *negative instances* of the coverability problem. Based on those results, we propose a new algorithmic schema to decide the coverability problem of WSTS. It works by iteratively constructing abstractions of the WSTS which become more and more precise. In parallel, it also explores, in a breadth-first fashion, the set of reachable states of the system (to be able to decide the *positive instances* of the problem). Thus, after a finite number of steps either a concrete trace to a *covered state* will be found, or *precise enough abstraction* will be computed to prove that no covered state can ever be reached. This informal statement is formalized in Theorem 3. The algorithm by itself is presented in subsection 3.3.

3.1 The And-Or Graph $\mathsf{Abs}(S, C', L')$

Definition 6. Given a WSTS $S = \langle C, c_0, \to, \leq\rangle$, an adequate domain of limits $\langle L, \sqsubseteq, \gamma\rangle$ for $\langle C, \leq\rangle$, a finite subset $C' \subseteq C$ with $c_0 \in C'$, and a finite subset $L' \subseteq L$ with $\top \in L'$, the And-Or graph $G = \langle V_A, V_O, v_i, \Rightarrow\rangle$, noted $\mathsf{Abs}(S, C', L')$, is defined as follows: (A_1) $V_O = C' \cup L'$; (A_2) $V_A = \{S \in 2^{L' \cup C'} \setminus \{\emptyset\} \mid \nexists d_1 \neq d_2 \in S : d_1 \sqsubseteq d_2\}$; ($\mathsf{A}_3$) $v_i = c_0$; ($\mathsf{A}_{4.1}$) $(n_1, n_2) \in \Rightarrow$ with

$n_1 \in V_A, n_2 \in V_O$ if and only if $n_2 \in n_1$; ($\mathsf{A}_{4.2}$) for any $n_1 \in V_O, n_2 \in V_A : (n_1, n_2) \in \Rightarrow$ if and only if (i) *successor covering:* $\mathsf{Post}(\gamma(n_1)) \subseteq \gamma(n_2)$, (ii) *preciseness:* $\nexists n \in V_A : \mathsf{Post}(\gamma(n_1)) \subseteq \gamma(n) \subset \gamma(n_2)$.

The following lemma states that the And-Or graph can be constructed for any WSTS and adequate domain of limits that are effective.

Lemma 1. *Given a* WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$ *and an adequate domain of limits* $\langle L, \sqsubseteq, \gamma \rangle$ *for* $\langle C, \leq \rangle$ *that are effective, a finite subset* $C' \subseteq C$ *with* $c_0 \in C'$*, and a finite subset* $L' \subseteq L$ *with* $\top \in L'$*,* $\mathsf{Abs}(S, C', L')$ *is effectively constructible.*

Notice that in $\mathsf{Abs}(S, C', L')$ all the nodes have at least one successor. Indeed, for all $n \in V_A$, since $n \neq \emptyset$ (following point $\mathsf{A}_{4.1}$ and point A_2 of Definition 6), n has at least one successor. Since And-nodes are subsets of limits that may contain the $\top$ element, with $\gamma(\top) = C$ (following point L_2 of Definition 1), we can always approximate for any $n \in V_O$ the (non-empty) set of successors of $\gamma(n)$, hence we are guaranteed to have at least one successor of n (point $\mathsf{A}_{4.2}$ of Definition 6).

Given a WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$, an associated And-Or graph $\mathsf{Abs}(S, L', C') = \langle V_A, V_O, v_i, \Rightarrow \rangle$, and an $\leq$-upward-closed set of states $U \subseteq C$, we note $\mathsf{Abs}(U)$ the set of nodes $v \in V_A \cup V_O$ such that $\gamma(v) \cap U \neq \emptyset$, that is, the set of nodes whose associated downward-closed set of states intersects with U. It is easy to show that this subset of nodes can be effectively computed for any effective WSTS with adequate domain of limits.

Degenerated case If an And-Or graph is such that any Or-node has exactly one successor, the And-Or graph is said to be *degenerated*. In that case, the avoidability problem is equivalent to the (un)reachability problem in a plain graph. From the definition of $\mathsf{Abs}(S, C', L')$, we can easily see that the And-Or graph will be degenerated if for any $d \in C' \cup L'$, there exists a *unique* minimal set $\gamma(D)$ such that $D \in V_A$ and $\mathsf{Succ}(\gamma(d)) \subseteq \gamma(D)$. This motivates the next definition:

Definition 7. Given a WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$ and an adequate domain of limits $\langle L, \sqsubseteq, \gamma \rangle$ for $\langle C, \leq \rangle$, we say that a pair $\langle C', L' \rangle$, where $C' \subseteq C$ with $c_0 \in C$ and $L' \subseteq L$ with $\top \in L'$, is *perfect* if for any $d \in C' \cup L'$, there exists a unique minimal set $D \subseteq C' \cup L'$ such that (i) $\mathsf{Post}(\gamma(d)) \subseteq \gamma(D)$ and (ii) there is no $D' \subseteq C' \cup L'$ with $\mathsf{Post}(\gamma(d)) \subseteq \gamma(D') \subset \gamma(D)$.

Lemma 2. *Given a* WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$*, an adequate domain of limits* $\langle L, \sqsubseteq, \gamma \rangle$ *for* $\langle C, \leq \rangle$*, a finite subset* $C' \subseteq C$ *with* $c_0 \in C'$*, and a finite subset* $L' \subseteq L$ *with* $\top \in L'$ *such that* $\langle C', L' \rangle$ *is perfect, then* $\mathsf{Abs}(S, C', L')$ *is a degenerated And-Or graph.*

3.2 Properties of $\mathsf{Abs}(S, C', L')$

In this section, we prove important properties of $\mathsf{Abs}(S, C', L')$. Roughly speaking, we prove now that the abstraction we have defined above is *adequate* for any pair $\langle C', L' \rangle$ such that $c_0 \in C'$ and $\top \in L'$ (Theorem 1) and *complete* (Theorem 2) for some pair $\langle C', L' \rangle$. To establish those results, we first show that $\mathsf{Abs}(S, C', L')$ can simulate for any $\langle C', L' \rangle$ such that $c_0 \in C'$ and $\top \in L'$ its underlying WSTS (the proof of this proposition is given in Appendix A):

Proposition 3 (Simulation). *Given a* WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$ *with an adequate domain of limits* $\langle L, \sqsubseteq, \gamma \rangle$ *for* $\langle C, \leq \rangle$*, the following holds for any* $C' \subseteq C$ *with* $c_0 \in C'$ *and* $L' \subseteq L$ *with* $\top \in L'$*: for any path* $c_0 c_1 \ldots c_k$ *of* S *and any unfolding* $T = \langle N, root, B, \Lambda \rangle$ *of* $\mathsf{Abs}(S, C', L')$ *there exists a path* $n_0 n_1 \ldots n_{2k}$ *of* T *with* $n_0 = root$ *and such that* $c_i \in \gamma(\Lambda(n_{2i}))$ *for* $0 \leq i \leq k$.

Theorem 1 (the proof is given in Appendix A) states the *adequacy* of the And-Or graph to decide the coverability problem.

Theorem 1 (Adequacy). *Given a* WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$*, an adequate domain of limits* $\langle L, \sqsubseteq, \gamma \rangle$ *for* $\langle C, \leq \rangle$*, and an upward-closed set* $U \subseteq C$*, the following holds for any* $C' \subseteq C$ *with* $c_0 \in C'$ *and* $L' \subseteq L$ *with* $\top \in L'$*:* if $\mathsf{Abs}(U)$ *is avoidable in* $\mathsf{Abs}(S, C', L')$, then $\mathsf{Reach}(S) \cap U = \emptyset$.

Finally, we prove the *completeness* of our approach. Intuitively, the next theorem puts forward that, when the pair $\langle C', L' \rangle$ is *precise enough*, $\mathsf{Abs}(S, C', L')$ allows us to decide *negative instances* of the coverability problem (the proof is given in Appendix A).

Theorem 2 (Completeness). *Given a* WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$*, an adequate domain of limits* $\langle L, \sqsubseteq, \gamma \rangle$ *for* $\langle C, \leq \rangle$ *and an upward closed set* $U \subseteq C$*, the following holds for any* $C' \subseteq C$ *with* $c_0 \in C'$ *and* $L' \subseteq L$ *with* $\top \in L'$ *such that* $\mathsf{CS}(S) \subseteq C' \cup L'$*:* if $\mathsf{Reach}(S) \cap U = \emptyset$ then $\mathsf{Abs}(U)$ *is avoidable in* $\mathsf{Abs}(S, C', L')$*.*

3.3 The new algorithmic schema

We have now at our disposal all the necessary results to propose a new schema of algorithms that decide the coverability problem on effective WSTS (in the sense of Definition 3). Let $S_0, S_1, \ldots, S_n \ldots$ be an infinite sequence of finite sets of reachable states of S such that (i) $\forall i \geq 0 : S_i \subseteq S_{i+1}$, (ii) $\forall c \in \mathsf{Reach}(S) : \exists i \geq 0 : c \in S_i$, and (iii) $c_0 \in S_0$. Let $L_0, L_1, \ldots, L_n, \ldots$ be a infinite sequence of finite sets of limits such that (i) $\forall i \geq 0 : L_i \subseteq L_{i+1}$, (ii) $\forall \ell \in L : \exists i \geq 0 : \ell \in L_i$ and (iii) $\top \in L_0$. The schema is given at Figure 1 and its proof of correctness is stated in Theorem 3 (the proof is given in Appendix A).

```
i := 0;
while (true) do
    "Expand" Compute S_i;
    "Enlarge" Compute L_i;
    "Check" if ∃c_1, ..., c_k : c_0 → ... → c_k with c_j ∈ S_i for all 0 ≤ j ≤ k and c_k ∈ U then return
    "Reachable";
    else if Abs(U) is avoidable in Abs(S, S_i, L_i) then return "Unreachable";
```

Fig. 1: **Abstract algorithm** Its inputs are a finite representation of a WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$ with the adequate limit domain $\langle L, \sqsubseteq, \gamma \rangle$ for $\langle C, \leq \rangle$ and a finite representation of upward-closed set of states $U \subseteq C$.

Theorem 3. *For any* WSTS S *with adequate domain of limits* $\langle L, \sqsubseteq, \gamma \rangle$ *that are effective, for any upward-closed set* U *represented by* $\mathsf{Min}(U)$*, Algorithm at Fig. 1 terminates after a finite amount of time and returns "Reachable" if* $\mathsf{Reach}(S) \cap U \neq \emptyset$*, "Unreachable" otherwise.*

Remark 1. Note that Theorem 3, that states the adequation and completeness of our algorithmic schema for the coverability problem of effective WSTS, is not in contradiction with the result of [12] which establishes that there does not exist a procedure that always terminates and returns a coverability set for a large class of WSTS, including ours. Indeed, to establish the correctness of our algorithm, we only need to ensure that a coverability set will be included at some point in the sequence of S_i's and L_i's. Nevertheless, given a pair $\langle S_i, L_i\rangle$, it is not possible to establish algorithmically that this pair contains a coverability set. Also, given a particular set upward-closed U, our algorithm may terminate before reaching a pair $\langle S_i, L_i\rangle$ that contains a coverability set, because the set U is reachable or because the abstraction constructed from a pair $\langle S_j, L_j\rangle$, with $j < i$, is sufficiently precise to prove that U is not reachable.

Remark 2. Note that the constraints on the sequence of L_i's computed by the algorithm of Fig. 1 may be relaxed. Indeed, those constraints ensure that the algorithm eventually considers a set of limits which allows to construct a graph that is precise enough to decide negative instances of the coverability problem. However, following Theorem 2, it is sufficient to ensure that there exists $i \geq 0$ such that $S_i \cup L_i$ contains a coverability set. Hence, only the limits of a coverability set must appear in the sequence of L_i's.

4 Application to Self-modifying Petri nets

Let us show how to apply the approach proposed in the previous section to solve the coverability problem for a large subclass of *Self-modifying Petri nets* [24] (SMPN), a general extension of Petri nets that includes almost all the monotonic extensions of Petri nets defined in the literature and for which, so far, there was no complete forward procedure.

In subsection 4.1, we present our subclass of SMPN, called *strongly monotonic self-modifying Petri nets*. In subsection 4.2, we instantiate the schema of algorithm presented in subsection 3.3 to the case of strongly monotonic SMPN. We first define the set of limits we will consider and how to construct the sequences of S_i's and L_i's. Then, we show that in this particular case, the And-Or graph one obtains is *degenerated* (Corollary 1). Finally, we deduce a simpler algorithm, that contains a decision procedure for the classical graph reachability problem instead of the avoidability problem in an And-Or graph.

4.1 Self-modifying Petri nets

A *Self-Modifying Petri net* [24], SMPN for short, is a tuple $\langle P, T, D^-, D^+, \mathbf{m}_0\rangle$. $P = \{p_1, \ldots, p_{k_P}\}$ is a finite set of places. A *marking* is a function $\mathbf{m} : P \to \mathbb{N}$ that assigns a natural value to each place. In the following, markings are also seen as tuples in $\mathbb{N}^{k_P}$ where the ith dimension is the value assigned to place p_i. $T = \{t_1, \ldots, t_{k_T}\}$ is a finite set of transitions. For any $1 \leq i \leq k_T$ and any $1 \leq j \leq k_P$, $D^-_{ij} : \mathbb{N}^{k_P} \to \mathbb{N}$ and $D^+_{ij} : \mathbb{N}^{k_P} \to \mathbb{N}$ describe respectively the input and output effect of transition t_i on place p_j. Namely, D^-_{ij} and D^+_{ij} are functions of the marking $\mathbf{m}$ of the form $\alpha + \sum_{k=1..k_P} \beta_k \cdot \mathbf{m}(p_k)$ where $\alpha \in \mathbb{N}$ and $\beta_k \in \mathbb{N}$ for all $1 \leq k \leq k_P$. $\mathbf{m}_0$ is the initial marking of the SMPN.

We define the quasi order $\preccurlyeq \subseteq \mathbb{N}^{k_P} \times \mathbb{N}^{k_P}$ on markings such that $\langle m_1, \ldots, m_{k_P}\rangle \preccurlyeq \langle m'_1, \ldots, m'_{k_P}\rangle$ if $m_i \leq m'_i$ for all $1 \leq i \leq k_P$. It is well-known that $\preccurlyeq$ is a well-quasi ordering.

A transition t_i is firable from a marking $\mathbf{m}$ if $\mathbf{m}(p_j) \geq D^-_{ij}(\mathbf{m})$ for all $p_j \in P$. Firing t_i from $\mathbf{m}$ leads to a marking $\mathbf{m}' \in \mathbb{N}^{k_P}$, noted $\mathbf{m} \to_{t_i} \mathbf{m}'$, such that, for any $p_j \in P : \mathbf{m}'(p_j) = \mathbf{m}(p_j) + D^+_{ij}(\mathbf{m}) - D^-_{ij}(\mathbf{m})$. Given a set S of markings and a transition t_i, $\mathsf{Post}(S, t_i) = \{\mathbf{m}' \mid \exists \mathbf{m} \in S : \mathbf{m} \to_{t_i} \mathbf{m}'\}$.

A SMPN $\mathcal{P}$ defines a transition system $\mathcal{T}_{\mathcal{P}} = \langle \mathbb{N}^{k_P}, \mathbf{m}_0, \to\rangle$ where $\to \subseteq \mathbb{N}^{k_P} \times \mathbb{N}^{k_P}$ is a transition relation and is such that we have $\langle \mathbf{m}, \mathbf{m}'\rangle \in \to$, noted $\mathbf{m} \to \mathbf{m}'$, if and only if there exists $t_i \in T$ such that t_i is firable from $\mathbf{m}$ and $\mathbf{m} \to_{t_i} \mathbf{m}'$.

A SMPN $\mathcal{P}$ is *$\preccurlyeq$-monotonic* when the underlying transition system $\mathcal{T}_{\mathcal{P}}$ satisfies the monotonicity property for $\preccurlyeq$. A SMPN $\mathcal{P}$ is *strongly monotonic* when for every transition t_i and markings $\mathbf{m}_1, \mathbf{m}_2$ and $\mathbf{m}_3$, the following holds: if $\mathbf{m}_1 \rightarrow_{t_i} \mathbf{m}_3$ and $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$, there exists $\mathbf{m}_4$ such that $\mathbf{m}_2 \rightarrow_{t_i} \mathbf{m}_4$ and $\mathbf{m}_3 \preccurlyeq \mathbf{m}_4$. Obviously, all the strongly monotonic SMPN are $\preccurlyeq$-monotonic.

We say that a transition t is *unfirable*, whenever there exists no marking $\mathbf{m}$ such that t is enabled in $\mathbf{m}$. In the following, we assume that the SMPN's we consider do not contain unfirable transitions. The following lemma (the proof is given in Appendix A) defines the syntactical subclass of SMPN's that are strongly monotonic.

Lemma 3. *Given a SMPN $\mathcal{P} = \langle P, T, D^-, D^+, \mathbf{m}_0 \rangle$ without unfirable transitions, $\mathcal{P}$ is strongly monotonic if and only if for all $t_i \in T, p_j \in P : D^-_{ij} = \alpha$ with $\alpha \in \mathbb{N}$ or $D^-_{ij} = \mathbf{m}(p_j)$.*

Although strongly monotonic SMPN is a sub-class of SMPN, it remains a general class of monotonic systems. Indeed, almost all the monotonic extensions of Petri nets studied in the literature are syntactical sub-classes of strongly monotonic SMPN, i.e. sub-classes defined by imposing constraints on the linear expressions defining the effect of transitions. Examples of such extensions are Petri nets with transfers [9], with reset [6] and Post self-modifying Petri nets [24]. On the other hand, the other monotonic extensions of Petri nets are not syntactical sub-classes of strongly monotonic SMPN, but we can construct (in polynomial time) a strongly monotonic SMPN with the same set of places that is equivalent to the original net with respect to the coverability problem. Examples of such extensions are Petri nets with non-blocking arcs [22] and Lossy Petri nets [8]. So the algorithm that we propose in the next section is a forward algorithm that decides the coverability problem for all monotonic extensions of Petri nets proposed in the literature.

4.2 A forward algorithm to decide the coverability problem for strongly monotonic SMPN

Domain of Limits We will consider the domain of limits $\langle \mathcal{L}, \preccurlyeq_e, \gamma(.) \rangle$ where $\mathcal{L} = (\mathbb{N} \cup \{+\infty\})^k \setminus \mathbb{N}^k$, $\preccurlyeq_e \subseteq (\mathbb{N} \cup \{+\infty\})^k \times (\mathbb{N} \cup \{+\infty\})^k$ is such that $\langle m_1, \ldots, m_k \rangle \preccurlyeq_e \langle m'_1, \ldots, m'_k \rangle$ if and only if $\forall 1 \leq i \leq k : m_i \leq m'_i$ where $c < +\infty$ for all $c \in \mathbb{N}$. $\gamma(.)$ is defined as: $\gamma(\mathbf{m}) = \{\mathbf{m'} \in \mathbb{N}^k \mid \mathbf{m'} \preccurlyeq_e \mathbf{m}\}$. In the following, tuples in $\mathcal{L}$ are called extended markings. It is well-known, see for instance [25], that the following lemma holds.

Lemma 4. *$\langle \mathcal{L}, \preccurlyeq_e, \gamma(.) \rangle$ is an adequate domain of limits for $\langle \mathbb{N}^k, \preccurlyeq \rangle$.*

Notice that in this case the $\top$ element such that $\gamma(\top) = \mathbb{N}^k$ is the marking that assigns $+\infty$ to all the places.

Given a strongly monotonic SMPN $\mathcal{P}$, we extend the underlying transition relation from markings to extended markings by assuming that $+\infty + +\infty = +\infty$, $c \cdot +\infty = +\infty$ for all $c \in \mathbb{N} \setminus \{0\}$, $0 \cdot +\infty = 0$, $+\infty + c = +\infty$ for all $c \in \mathbb{Z}$.

Since our algorithm requires the WSTS and its associated domain of limits to be effective (Definition 3), we state the following lemma (proof omitted):

Lemma 5. *Any strongly monotonic SMPN $\mathcal{P}$ with the adequate domain of limits $\langle \mathcal{L}, \preccurlyeq_e, \gamma(.) \rangle$ are effective.*

The following definition explains how we construct the S_i's and L_i's. Following Definition 6, this is sufficient to define the And-Or graphs built by our verification algorithm.

Definition 8. *The sequences of S_i's and L_i's are defined as follows:* (D_1) $S_i = \{0,\ldots,i\}^k \cup \{\mathbf{m}_0\}$, *i.e.* S_i *is the set of markings where each place is bounded by* i *(plus the initial marking);* (D_2) $L_i = \{\mathbf{m} \in \{0,\ldots i,+\infty\}^k \mid \mathbf{m} \notin \mathbb{N}^k\}$.

It is easy to see that (i) for all $i \geq 0 : S_i \subset S_{i+1}$ and $L_i \subset L_{i+1}$, (ii) for any $\mathbf{m} \in \mathbb{N}^k$, there exists $i \in \mathbb{N}$ such that for all $j \geq i : \mathbf{m} \in S_j$, (iii) for any $\mathbf{m} \in \mathcal{L}$, there exists $i \in \mathbb{N}$ such that for all $j \geq i : \mathbf{m} \in L_j$, and (iv) $\mathbf{m}_0 \in S_0$ and $\top \in L_0$.

Degenerated And-Or graph Let us show that in the present case, one obtains a *degenerated* And-Or graph. For this purpose, we prove, following Lemma 2, that the pairs $\langle S_i, L_i\rangle$ are *perfect* pairs (the proof is given in Appendix A).

Lemma 6. *Given a* SMPN $\mathcal{P} = \langle P, T, D^-, D^+\rangle$ *with the adequate domain of limits* $\langle \mathcal{L}, \preccurlyeq_e, \gamma(.)\rangle$ *and the sets* $S_i \subseteq \mathbb{N}^{k_P}$ *and* $L_i \subseteq \mathcal{L}$ *constructed following Definition 8, any pair* $\langle S_i, L_i\rangle$ *is a perfect pair.*

From Lemma 6 and Lemma 2, the following corollary holds.

Corollary 1. *Given a strongly monotonic* SMPN *net* $\mathcal{P}$ *with the adequate domain of limits* $\langle \mathcal{L}, \preccurlyeq_e, \gamma(.)\rangle$ *and the sets* $S_i \subseteq \mathbb{N}^{k_P}$ *and* $L_i \subseteq \mathcal{L}$ *constructed following Definition 8,* $\mathsf{Abs}(\mathcal{P}, S_i, L_i)$ *is a degenerated And-Or graph.*

Algorithm for the coverability problem Let $\mathsf{Abs}(\mathcal{P}, i)$ be the graph (degenerated And-Or graph) $\mathsf{Abs}(\mathcal{P}, S_i, L_i)$ constructed from $\mathcal{P}$, S_i and L_i. We note $\Rightarrow$ its transition relation. We define $\mathsf{Reach}_{\mathsf{exact}}(\mathsf{Abs}(\mathcal{P}, i))$ as the set $\{\mathbf{m} \mid \mathbf{m}_0 \Rightarrow \mathbf{m}_1 \Rightarrow \ldots \Rightarrow \mathbf{m}_n \text{ with } \forall 1 \leq j \leq n : \mathbf{m}_j \in S_i, \mathbf{m}_n = \mathbf{m}\}$ and $\mathsf{Reach}(\mathsf{Abs}(\mathcal{P}, i))$ as the set $\{\mathbf{m} \mid \mathbf{m}_0 \Rightarrow \mathbf{m}_1 \Rightarrow \ldots \Rightarrow \mathbf{m}_n \text{ with } \forall 1 \leq j \leq n : \mathbf{m}_j \in S_i \cup L_i, \mathbf{m}_n = \mathbf{m}\}$. By applying the schema presented in Section 3 to strongly monotonic self-modifying Petri nets, we obtain the algorithm at Fig. 2. Remark that this algorithm is *incremental*: one can compute $\mathsf{Reach}_{\mathsf{exact}}(\mathsf{Abs}(\mathcal{P}, i+1))$ by extending $\mathsf{Reach}_{\mathsf{exact}}(\mathsf{Abs}(\mathcal{P}, i))$ for all $i \geq 0$. Similarly, one can construct $\mathsf{Reach}(\mathsf{Abs}(\mathcal{P}, i))$ from $\mathsf{Reach}_{\mathsf{exact}}(\mathsf{Abs}(\mathcal{P}, i))$.

$i \leftarrow 1$;
while *(***true***)* **do**
  if $\exists \mathbf{m} \in \mathsf{Reach}_{\mathsf{exact}}(\mathsf{Abs}(\mathcal{P}, i)), \mathbf{m}' \in G_U : \mathbf{m} \preccurlyeq \mathbf{m}'$ **then return** *"Reachable"*;
  else
    if $\nexists \mathbf{m} \in \mathsf{Reach}(\mathsf{Abs}(\mathcal{P}, i)), \mathbf{m}' \in G_U : \mathbf{m} \preccurlyeq_e \mathbf{m}'$ **then return** *"Unreachable"*;
    else $i \leftarrow i + 1$;

Fig. 2: **A forward algorithm for** SMPN Its inputs are $\mathcal{P}$, a strongly monotonic self-modifying Petri system and G_U, the set of minimal element of the $\preccurlyeq$-upward closed set U.

Theorem 4. *The algorithm of Fig. 2 returns "Reachable" if* $\mathsf{Reach}(\mathcal{C}) \cap U \neq \emptyset$, *"Unreachable" otherwise.*

We built a first (naive) prototype that implements the algorithm at Fig. 2. We report on its performance in Appendix B. Although rather rough, our prototype already performs better than its counterpart based on a backward approach.

5 Application to Lossy Channel Systems

To show the generality of our new approach, we apply our schema of algorithm to *lossy channel systems*, which are systems made up of automata extended with FIFO channels that may lose messages. We recall the model, define an adequate domain of limits, show how to construct the sets S_i's and L_i's and discuss the construction of the And-Or graph.

A *Lossy Channel System*, LCS for short, is a tuple $\mathcal{C} = \langle Q, q_i, F, \Sigma, T\rangle$ where Q is a finite set of locations, $q_i \in Q$ is the initial location, F is a finite set of channels, Σ is a finite alphabet, $T \subseteq Q \times Op \times Q$ where $Op : F \to \bigcup_{a\in\Sigma}\{?a, !a\}$. A state is a pair $\langle q, W\rangle$ where $q \in S$, $W : F \to \Sigma^*$. In the following, $\mathcal{S}_\mathcal{C}$ will denote the set of states of the LCS $\mathcal{C}$. We define the order $\precsim$ on states in $\mathcal{S}_\mathcal{C}$ such that for any $s = \langle q, W\rangle, s' = \langle q', W'\rangle : s \precsim s'$ if and only if $q = q'$ and $W(c)$ is a (not necessarily contiguous) subword of $W'(c)$ for all $c \in F$, i.e $W(c)$ is obtained from $W'(c)$ by deleting characters. It is well-known that $\precsim$ is a well-quasi order (see for instance [2]). A transition $t = \langle q_1, Op, q_2\rangle \in T$ is firable from state $\langle q, W\rangle$ if $q = q_1$ and for all $c \in F : Op(c) = ?a$ implies that $W(c) = a \cdot \sigma$ where $\sigma \in \Sigma^*$. Given a transition $t = \langle q_1, Op, q_2\rangle$ firable from state $\langle q, W\rangle$, $\delta(\langle q, W\rangle, t) = \langle q', W\rangle$ such that $q' = q_2$ and for all $c \in F$, either $Op(c) = ?a$, $W(c) = a\cdot\sigma$ and $W'(c) = \sigma$, or $Op(c) = !a$, $W(c) = \sigma$ and $W'(c) = \sigma\cdot a$. When t is firable from s, firing t from s leads to any state s', noted $s \rightarrowtail_t s'$, such that $s' \precsim \delta(s,t)$. Given a set S of states and a transition t, $\mathsf{Post}(S,t) = \{s' \mid \exists s \in S : s \rightarrowtail_t s'\}$. A LCS $\mathcal{C} = \langle Q, q_i, F, \Sigma, \to\rangle$ defines a transition system $\langle \mathcal{S}_\mathcal{C}, s_0, \to\rangle$ where $s_0 = \langle q_i, W_i\rangle$ such that $W_i(c) = \varepsilon$ for all $c \in F$ and for all $s_1, s_2 \in \mathcal{S}_\mathcal{C} : s_1 \to s_2$ if and only if $\exists t \in T : s_1 \rightarrowtail_t s_2$. It is well-known that transition relations defined by LCS are $\precsim$-monotonic. In the following, we always consider a LCS $\mathcal{C} = \langle Q, q_i, F, \Sigma, T\rangle$.

Domain of limits Let $L(\Sigma)$ be the set of downward closed regular expressions ($\mathsf{dc\text{-}re}$) $\{(a_1 + \ldots + a_n)^* \mid \forall 1 \le i \le n : a_i \in \Sigma, \forall a_i, a_j : i \ne j$ implies that $a_i \ne a_j\} \cup \{(a+\varepsilon) \mid a \in \Sigma\} \cup \{\varepsilon\}$. A simple regular expression (sre) is either a $\mathsf{dc\text{-}re}$ or an expression $a_1 \cdot \ldots \cdot a_n$ where $\forall 1 \le i \le n : a_i$ is a $\mathsf{dc\text{-}re}$. The size of a sre is the number of $\mathsf{dc\text{-}re}$ that compose it. The set of limits is the set $\mathcal{L}(\Sigma, Q) = \{\langle q, E\rangle \mid q \in Q, E : C \to L(\Sigma)^*$ assigns a sre to each channel[3]$\} \cup \{\top\}$. For $\langle q, E\rangle \in \mathcal{L}(\Sigma, Q)$: $[\![\langle q, E\rangle]\!]$ denotes the set of pairs $\langle q, W\rangle \in \mathcal{S}_\mathcal{C}$ such that $W(c)$ is a word in the language generated by the regular expression $E(c)$ for all $c \in C$. We define the function $\gamma : \mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q) \to 2^{\mathcal{S}_\mathcal{C}}$ such that (i) for all $\langle q, W\rangle \in \mathcal{S}_\mathcal{C} : \gamma(\langle q, W\rangle) = \{\langle q, W'\rangle \mid \langle q, W'\rangle \precsim \langle q, W\rangle\}$, (ii) $\gamma(\top) = \{\langle q, W\rangle \mid q \in Q, W(c) \in \Sigma^*$ for all $c \in C\}$ and (iii) for all $\langle q, E\rangle \in \mathcal{L}(\Sigma, Q) \setminus \{\top\}$: $\gamma(\langle q, E\rangle) = [\![\langle q, E\rangle]\!]$. We define $\overline{\sqsubseteq} : (\mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)) \times (\mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q))$ as follows : $c_1 \overline{\sqsubseteq} c_2$ if and only if $\gamma(c_1) \subseteq \gamma(c_2)$. The following Theorem holds (the proof is given in Appendix A):

Theorem 5. *$(\mathcal{L}(\Sigma, Q), \overline{\sqsubseteq}, \gamma)$ is an adequate domain of limits for $(\mathcal{S}_\mathcal{C}, \precsim)$.*

Moreover, the following theorem says that any LCS $\mathcal{C}$ with the adequate domain of limits $(\mathcal{L}(\Sigma, Q), \overline{\sqsubseteq}, \gamma)$ are effective. The proof can be found in Appendix A.

Theorem 6. *Any LCS $\mathcal{C}$ with the adequate domain of limits $(\mathcal{L}(\Sigma), \overline{\sqsubseteq}, \gamma)$ are effective.*

Construction of the S_i's and the L_i's We construct the sequences of the S_i's and L_i's as follows. $S_i = \{\langle q, W\rangle \in \mathcal{S}_\mathcal{C} \mid q \in Q, \forall c \in C : W(c) = \varepsilon$ or $W(c) = a_1 \cdot \ldots \cdot a_n$ with $\forall 1 \le j \le n : a_j \in \Sigma, n \le i\}$, i.e. S_i is the set of states where the contents of the channels are words of size at most

[3] We also require that E does not assign ε to all the channels because we require in Definition 1 that the set of limits is disjoint from $\mathcal{S}_\mathcal{C}$.

i. Similarly, $L_i = \{\langle q, E\rangle \in \mathcal{L}(\Sigma, Q) \mid q \in Q, \forall c \in C : E(c) = \varepsilon$ or $E(c) = e_1 \cdot \ldots \cdot e_n$ with $\forall 1 \leq j \leq n : e_j \in L(\Sigma), n \leq i\} \cup \{\top\}$, i.e. L_i is the set of limits that assign sre of size of most i to channels (plus the $\top$ element).

It is easy to see that (i) $S_i \subseteq S_{i+1}$ and $L_i \subseteq L_{i+1}$ for all $i \geq 0$, (ii) for all $c \in \mathcal{S}_\mathcal{C}$ there exists $i \geq 0$ such that $c \in S_i$ and for all $\ell \in \mathcal{L}(\Sigma, Q)$ there exists $i \geq 0$ such that $\ell \in L_i$, (iii) $\langle q_i, W_i\rangle \in S_0$ where $\forall c \in C : W_i(c) = \varepsilon$ and (iv), by construction $\top \in L_0$.

Construction of the And-Or graph In order to construct the And-Or graph, we need to construct the set of Or-nodes (point A_1), the set of And-nodes (point A_2) and the transition relation between nodes (points $\mathsf{A}_{4.1}$ and $\mathsf{A}_{4.2}$). The two first points are obvious. Let us focus on the construction of the transition relation. Given the two sets S_i and L_i as defined above, the successors of And-nodes are computed as follows. For any And-node $n \in 2^{S_i \cup L_i} \setminus \{\emptyset\}$, we have $\langle n, n'\rangle \in \Rightarrow$ if and only if $n' \in n$. In order to define the successors of an Or-node, we need the following functions. Let $\widetilde{\mathsf{Post}}(.,.) : (\mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)) \times T \rightarrow \mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)$ be the partial function defined in [2] such that $\widetilde{\mathsf{Post}}(\ell, t) = \ell'$ and $\widetilde{\mathsf{Post}}(\gamma(\ell), t) = \gamma(\ell')$ if t is firable from $\gamma(\ell)$, otherwise $\widetilde{\mathsf{Post}}(\ell, t)$ is undefined. In other words, $\widetilde{\mathsf{Post}}(\gamma(\ell), t)$ returns the element ℓ' in $\mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)$ such that $\gamma(\ell')$ is the set of successors of $\gamma(\ell)$ by firing t. The partial function $\mathsf{App}(\ell, t, i) : (\mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)) \times T \times \mathbb{N} \rightarrow 2^{\mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)}$ is such that $\mathsf{App}(\ell, t, i)$ is defined if t is firable from $\gamma(\ell)$. In that case, $\mathsf{App}(\ell, t, i) = \widetilde{\mathsf{Post}}(\ell, t)$ if $\widetilde{\mathsf{Post}}(\ell, t) \in S_i \cup L_i$, otherwise $\mathsf{App}(\ell, t, i) = \{\ell' \in S_i \cup L_i \mid \widetilde{\mathsf{Post}}(\ell, t) \overline{\sqsubseteq} \ell', \neg\exists \ell'' \in S_i \cup L_i : \widetilde{\mathsf{Post}}(\ell, t) \overline{\sqsubseteq} \ell'' \overline{\sqsubseteq} \ell'\}$. If t is unfirable from $\gamma(\ell)$, $\mathsf{App}(\ell, t, i)$ is undefined for all $i \geq 0$. In other words, when $\mathsf{App}(\ell, t, i)$ is defined, it returns $\widetilde{\mathsf{Post}}(\ell, t)$ if it is in the set $S_i \cup L_i$ of states and limits that we consider during the construction of the graph, otherwise it returns the set of elements $\ell' \in S_i \cup L_i$ such that $\gamma(\ell')$ is a best overapproximation of $\gamma(\widetilde{\mathsf{Post}}(\ell, t))$. Notice that we can always construct $\mathsf{App}(\ell, t, i)$ since $\widetilde{\mathsf{Post}}(\ell, t)$ is constructible [2], S_i and L_i are finite and $\overline{\sqsubseteq}$ is decidable. Let $\mathsf{Firable}(\ell) = \{t_1, \ldots, t_{k_\ell}\}$ be the set of k_ℓ transitions that are firable from $\gamma(\ell)$ and $\mathfrak{Post}(\ell, i) = \{S \subseteq S_i \cup L_i \mid \exists c_1, \ldots, c_{k_\ell} \in S_i \cup L_i : \forall 1 \leq j \leq k_\ell : t_j \in \mathsf{Firable}(\ell), c_j \in \mathsf{App}(\ell, t_j, i), c \in S$ if and only if $\exists 1 \leq j \leq k_\ell : c = c_j\}$, i.e. $\mathfrak{Post}(\ell, i)$ is the set of sets of elements in $S_i \cup L_i$ that represent an over-approximation of the successors of $\gamma(\ell)$. Sets in $\mathfrak{Post}(\ell, i)$ satisfy the *covering* property of point $\mathsf{A}_{4.2}$, but they may not be minimal, i.e. they could contain two elements that are ordered, and they may not represent most precise overapproximations of the set of successors. For any $n \in V_O$, we define the set $\mathsf{Succ}(n, i)$ of successor And-nodes of n such that $\mathsf{Succ}(n, i) = \{S \subseteq S_i \cup L_i \mid \exists S' \in \mathfrak{Post}(n, i) : S \subseteq S', \gamma(S) = \gamma(S'), \forall c_1, c_2 \in S : c_1 \neq c_2$ implies $c_1 \overline{\not\sqsubseteq} c_2, \nexists S'' \in \mathfrak{Post}(n, i) : \gamma(S'') \subset \gamma(S)\}$. That is $\mathsf{Succ}(n, i)$ is the set of most precise and minimal approximations of the set of successors of $\gamma(n)$. That set is constructible since $\mathfrak{Post}(\ell, i)$ is constructible and, following Theorem 6 and so E_4 of Definition 3, $\gamma(S) \subseteq \gamma(S')$ is decidable for any finite $S, S' \subseteq \mathcal{S}_\mathcal{C} \cup \mathcal{L}(\Sigma, Q)$.

6 Conclusion

In this paper, we have defined a new approach to solve the coverability problem of WSTS, which we call "Expand, Enlarge and Check". When applied to a large class of monotonic counter systems (the strong monotonic Self-modifying Petri nets), our approach produces an algorithm that uses forward analysis to decide the coverability problem. Up to now, such a forward approach was known only for Petri nets (the Karp and Miller algorithm), a restricted subclass of strong monotonic SMPN. We have demonstrated the generality of our approach by showing how to apply the algorithmic schema to lossy channel systems. Preliminary implementation results are encouraging and confirm that forward verification algorithms are usually more efficient than backward verification algorithms (see Appendix B).

Future works The most important future work will be to go on with the implementation for SMPN. We also plan to build a prototype to verify LCS and Timed Petri nets.

References

1. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General Decidability Theorems for Infinite-state Systems. In *Proceedings of the 11th Annual Symposium on Logic in Comuter Science (LICS'96)*, pages 313–321. IEEE Computer Society Press, 1996.
2. P.A. Abdulla, A Bouajjani, and B Jonsson. On-the-Fly Analysis of Systems with Unbounded, Lossy FIFO Channels. In *Proceedings of the 10th International Conference on Computer Aided Verification (CAV'98)*, volume 1427 of *LNCS*, pages 305–318. Springer, 1998.
3. P.A. Abdulla and B. Jonsson. Verifying Programs with Unreliable Channels. In *Proceedings of the 8th IEEE International Symposium in Logic in Computer Science (LICS'93)*, pages 160–170. IEEE Computer Society Press, 1993.
4. Parosh Abdulla, Aurore Annichini, and Ahmed Bouajjani. Symbolic verification of lossy channel systems: Application to the bounded retransmission protocol. In *Proc. 5th Intern. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99)*, number 1579 in LNCS, pages 208–222. Springer-Verlag, 1999.
5. R. Alur and D.L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–236, 1994.
6. T. Araki and T. Kasami. Some decision problems related to the reachability problem for petri nets. *Theoretical Computer Science*, 3(1):85–104, 1977.
7. S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. FAST: Fast acceleration of symbolic transition systems. In *Proceedings of the 15th International Conference on Computer Aided Verification (CAV'03)*, will be published in LNCS. Springer, 2003.
8. A. Bouajjani and R. Mayr. Model Checking Lossy Vector Addition Systems. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, volume 1563 of *LNCS*, pages 323–333. Springer, 1999.
9. G. Ciardo. Petri nets with marking-dependent arc multiplicity: properties and analysis. In *Proceedings of the 15th International Conference on Applications and Theory of Petri Nets (ICATPN 94)*, volume 815 of *LNCS*, pages 179–198. Springer, 1994.
10. G. Delzanno, J.-F. Raskin, and L. Van Begin. Covering Sharing Trees: Efficient Data Structurs for the Automated Verification of Parametrized Systems. *accepted for publication in Software Tools for Technology Transfer Manuscript*, 2002.
11. G. Delzanno, J.-F. Raskin, and L. Van Begin. Towards the Automated Verification of Multi-threaded Java Programs. In *Proceedings of the International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS 2002)*, volume 2280 of *LNCS*, pages 173–187. Springer, 2002.
12. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset Nets Between Decidability and Undecidability. In *In Proceedings of the 25th International Colloquium on Automata, Languages, and Programming (ICALP'98)*, volume 1443 of *LNCS*, pages 103–115. Springer, 1998.
13. E. A. Emerson and K. S. Namjoshi. On Model Checking for Non-deterministic Infinite-state Systems. In *Proceedings of the 13th Annual Symposium on Logic in Computer Science (LICS '98)*, pages 70–80. IEEE Computer Society Press, 1998.
14. J. Esparza, A. Finkel, and R. Mayr. On the Verification of Broadcast Protocols. In *Proceedings of the 14th Annual Symposium on Logic in Computer Science (LICS'99)*, pages 352–359. IEEE Computer Society Press, 1999.

15. A. Finkel. Reduction and Covering of Infinite Reachability Trees. *Information and Computation*, 89(2):144–179, 1990.
16. A. Finkel, J.-F. Raskin, M. Samuelides, and L. Van Begin. Monotonic Extensisions of Petri Nets : Forward and Backward Search Revisited. In *Proceedings of the 4th international workshop on verification of infinite-state systems (INFINITY 2002)*, volume 68 of *ENTCS*. Elsevier, 2002.
17. A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.
18. S. M. German and A. P. Sistla. Reasoning about Systems with Many Processes. *Journal of ACM*, 39(3):675–735, 1992.
19. T. A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Symposium on Logic in Computer Science (LICS '96)*, page 278. IEEE Computer Society, 1996.
20. T. A. Henzinger, O. Kupferman, and S. Qadeer. From *pre*historic to *post*modern symbolic model checking. *Formal Methods in System Design*, 23(3):303–327, 2003.
21. Neil Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and System Sciences*, 22(3):384–406, 1981.
22. J.-F. Raskin and L. Van Begin. Petri Nets with Non-blocking Arcs are Difficult to Analyse. In *Proceedings of the 5th International Workshop on Verification of Infinite-state Systems (INFINITY 2003)*, volume 96 of *ENTCS*. Elsevier, 2003.
23. W. Reisig. *Petri Nets. An introduction.* Springer, 1986.
24. R. Valk. On the computational power of extended petri nets. In *Proceedings of the 7th symposium on Mathematical Foundations of Computer Science*, volume 64 of *LNCS*, pages 527–535. Springer, 1978.
25. L. Van Begin. *Efficient Verification of Counting Abstractions for Parametric systems.* PhD thesis, Université Libre de Bruxelles, Belgium, 2003.

A Proofs

Proof of Proposition 3 Given a WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$ with an adequate domain of limits $\langle L, \sqsubseteq, \gamma \rangle$ for $\langle C, \leq \rangle$, the following holds for any $C' \subseteq C$ with $c_0 \in C'$ and $L' \subseteq L$ with $\top \in L'$: for any path $c_0 c_1 \dots c_k$ of S and any unfolding $T = \langle N, root, B, \Lambda \rangle$ of $\mathsf{Abs}(S, C', L')$ there exists a path $n_0 n_1 \dots n_{2k}$ of T with $n_0 = root$ and such that $c_i \in \gamma(\Lambda(n_{2i}))$ for $0 \leq i \leq k$.

Proof. Let $c_0, \dots, c_k$ be a path of S. For any unfolding, we will show, by induction on the length k of the path in S, that there exists a path $n_0 n_1 \dots n_{2k}$ of the unfolding such that $c_i \in \gamma(\Lambda(n_{2i}))$ for all i such that $0 \leq i \leq k$.
Base case: The base case is trivial since $\Lambda(root) = c_0$ following A_3 and C_1.
Induction step: Suppose that there exists a path $P = n_0, \dots, n_{2i}$ $(i < k)$ of the unfolding, such that $c_j \in \gamma(\Lambda(n_{2j}))$ for all j such that $1 \leq j \leq i$. Let us show that there exists a path $n_0 \dots n_{2(i+1)}$ of the unfolding, where $c_j \in \gamma(\Lambda(n_{2j}))$ for all j such that $1 \leq j \leq i+1$. Since $c_i \rightarrow c_{i+1}$, from point $\mathsf{A}_{4.2}$ of Definition 6 we have that all the And-nodes $v = \{d_1, \dots, d_\ell\}$ in $\mathsf{Abs}(S, C', L')$ with $\Lambda(n_{2i}) \Rightarrow v$ are such that $c_{i+1} \in \gamma(d_j)$ for some j such that $1 \leq j \leq \ell$. Hence, following C_3, all the successors of n_{2i} in the unfolding are nodes n with $\Lambda(n) = \{d_1, \dots, d_\ell\}$ such that $c_{i+1} \in \gamma(d_j)$ for some j. Moreover, following $\mathsf{A}_{4.1}$ and C_2, n has a successor n' such that $\Lambda(n') = d_j$, i.e. $c_{i+1} \in \gamma(\Lambda(n'))$. We conclude that the path P extended with the nodes n and n' is such that the $i+1$ Or-nodes n_j are such that $c_j \in \gamma(\Lambda(n_j))$.

Proof of Theorem 1 Given a WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$, an adequate domain of limits $\langle L, \sqsubseteq, \gamma \rangle$ for $\langle C, \leq \rangle$, and an upward-closed set $U \subseteq C$, the following holds for any $C' \subseteq C$ with $c_0 \in C'$ and $L' \subseteq L$ with $\top \in L'$: *if* $\mathsf{Abs}(U)$ is avoidable in $\mathsf{Abs}(S, C', L')$, *then* $\mathsf{Reach}(S) \cap U = \emptyset$.

Proof. Suppose that it is not the case, that is $\mathsf{Abs}(U)$ is avoidable in $\mathsf{Abs}(S, C', L')$ but $\mathsf{Reach}(S) \cap U \neq \emptyset$. Hence, there exists a path $c_0, \ldots, c_k$ in S with $c_k \in U$. From Proposition 3, we have that for any unfolding $T = \langle N, root, B, \Lambda \rangle$ of $\mathsf{Abs}(S, C', L')$, there exists a path $n_0 \ldots n_{2k}$ in T with $n_0 = root$ such that $c_i \in \gamma(\Lambda(n_{2i}))$ for all i such that $0 \leq i \leq k$. Hence, $\Lambda(N) \cap \mathsf{Abs}(U) \neq \emptyset$ and we obtain a contradiction.

Proof of Theorem 2 Given a WSTS $S = \langle C, c_0, \rightarrow, \leq \rangle$, an adequate domain of limits $\langle L, \sqsubseteq, \gamma \rangle$ for $\langle C, \leq \rangle$ and an upward closed set $U \subseteq C$, the following holds for any $C' \subseteq C$ with $c_0 \in C'$ and $L' \subseteq L$ with $\top \in L'$ such that $\mathsf{CS}(S) \subseteq C' \cup L'$: *if* $\mathsf{Reach}(S) \cap U = \emptyset$ *then* $\mathsf{Abs}(U)$ is avoidable in $\mathsf{Abs}(S, C', L')$.

Proof. Suppose that it is not the case, i.e. there exists $\mathsf{CS}(S) \subseteq C' \cup L'$, $\mathsf{Reach}(S) \cap U = \emptyset$ and $\mathsf{Abs}(U)$ is not avoidable in $\mathsf{Abs}(S, C', L')$. We will show by induction that in that case we can construct an unfolding having every node n such that $\gamma(\Lambda(n)) \subseteq \gamma(\mathsf{CS}(S))$. Since $\gamma(\mathsf{CS}(S)) \cap U = \emptyset$, we conclude that $\mathsf{Abs}(U)$ is avoidable and we obtain a contradiction.

Base case: Notice that $root = c_0$ following C_1 and A_3, and $c_0 \in \gamma(\mathsf{CS}(S))$ following condition CS_2 of Definition 4. Moreover, by CS_2 we also conclude that $\mathsf{Post}(c_0) \subseteq \gamma(\mathsf{CS}(S))$. Hence, following $\mathsf{A}_{4.2}$, there exists $v \in V_A$ (the set of And-nodes) with $v_i \Rightarrow v$ and $\gamma(v) \subseteq \gamma(\mathsf{CS}(S))$ since v satisfies the preciseness property of $\mathsf{A}_{4.2}$ and $\mathsf{CS}(S)$ covers the successors of v_i. We choose such an And-node v and add one successor node n to $root$ such that $\Lambda(n) = v$.

Induction step: Suppose that we can construct $2k$ layers of the unfolding such that for all the nodes n of the $2k$ first layers we have that $\gamma(n) \subseteq \gamma(\mathsf{CS}(S))$. Let us show that we can construct $2k + 2$ layers such that for all the nodes n of the $2k + 2$ first layers we have that $\gamma(n) \subseteq \gamma(\mathsf{CS}(S))$.

By induction hypothesis, all the And-nodes n in the $2k$-th layer are such that $\Lambda(n) = \{d_1, \ldots, d_\ell\}$ and $\gamma(\Lambda(n)) \subseteq \gamma(\mathsf{CS}(S))$. Since, following $\mathsf{A}_{4.1}$, all the successors nodes v of $\Lambda(n)$ in $\mathsf{Abs}(S, C', L')$ are such that $v \in \Lambda(n)$, we have that $\gamma(v) \subseteq \gamma(\mathsf{CS}(S))$. We conclude, following C_2, that all the Or-nodes n' of the $2k + 1$-th layer are such that $\gamma(\Lambda(n')) \subseteq \gamma(\mathsf{CS}(S))$.

Since following W_2, S is monotonic, if $c \rightarrow c''$ with $c \sqsubseteq c'$, then there exists c''' such that $c'' \sqsubseteq c'''$ and $c' \rightarrow c'''$. Moreover, all the nodes n of the $2k + 1$-th layer are such that if $c \in \gamma(\Lambda(n))$, then there exists $c' \in \mathsf{Reach}(S)$ with $c \sqsubseteq c'$. Indeed, if it is not the case, since $\gamma(\Lambda(n)) \subseteq \gamma(\mathsf{CS}(S))$, $\mathsf{CS}(S)$ is not the minimal downward closed set that contains $\mathsf{Reach}(S)$. But following Definition 4 it is not the case and we obtain a contradiction.

Hence, for all the nodes n of the $2k + 1$-th layer we have $\mathsf{Post}(\gamma(\Lambda(n))) \subseteq \gamma(\mathsf{CS}(S))$ and there exists following $A_{4.2}$ an And-node v with $\gamma(v) \subseteq \gamma(\mathsf{CS}(S))$ and $\Lambda(n) \Rightarrow v$ since v satisfies the preciseness property of $\mathsf{A}_{4.2}$ and $\mathsf{CS}(S)$ covers the successors of $\gamma(\Lambda(n))$. So, we choose such a node v and add one successor n' to n such that $\Lambda(n') = v$. That allows us to conclude that we can construct the $2k + 2$-th first layers of the unfolding such that all the nodes n are such that $\gamma(\Lambda(n)) \subseteq \mathsf{CS}(S)$.

Proof of Theorem 3 For any WSTS S with adequate domain of limits $\langle L, \sqsubseteq, \gamma \rangle$ that are effective, for any upward-closed set U represented by $\mathsf{Min}(U)$, Algorithm at Fig. 1 terminates after a finite amount of time and returns "Reachable" if $\mathsf{Reach}(S) \cap U \neq \emptyset$, "Unreachable" otherwise.

Proof. First, notice that S_i is finite for all $i \geq 0$ and the transition relation $\rightarrow$ is decidable (following E_2), hence we can decide if there exists a path leading to U where only states in S_i appear (this is also possible because $\leq$ is decidable). Moreover, the And-Or graph is constructible, following Lemma 1, and it is easy to show that the set of Or-nodes $\mathsf{Abs}(U)$ is constructible. Hence, we can effectively test whether $\mathsf{Abs}(U)$ is avoidable in $\mathsf{Abs}(S, S_i, L_i)$ (remember that the avoidability problem is decidable, since it is PTIME-complete).

It remains to prove that the algorithm returns an answer after a finite number of iterations of the loop.

If $\mathsf{Reach}(S) \cap U \neq \emptyset$, we have from Theorem 1 that $\mathsf{Abs}(U)$ is not avoidable in $\mathsf{Abs}(S, S_i, L_i)$ for all $i \geq 0$. Moreover, since for all $c \in \mathsf{Reach}(S)$ there exists j such that $c \in S_{j'}$ for all $j' \geq j$, there exists $i \geq 0$ such that we have $c_0 \rightarrow \ldots \rightarrow c_k$ with $c_j \in S_i$ for all j such that $0 \leq j \leq k$ and $c_k \in U$. We conclude that Algorithm 1 returns "Reachable" if $\mathsf{Reach}(S) \cap U \neq \emptyset$.

If $\mathsf{Reach}(S) \cap U = \emptyset$, we know that there exists $i \geq 0$ and a finite coverability set $\mathsf{CS}(S)$ such that $\mathsf{CS}(S) \subseteq S_i \cup L_i$. Hence, from Proposition 2 we have that $\mathsf{Abs}(U)$ is avoidable in $\mathsf{Abs}(S, S_i, L_i)$ and we conclude that Algorithm 1 returns "Unreachable" if $\mathsf{Reach}(S) \cap U = \emptyset$.

Proof of Lemma 3 Given a SMPN $\mathcal{P} = \langle P, T, D^-, D^+, \mathbf{m}_0 \rangle$ without unfirable transitions, $\mathcal{P}$ is strongly monotonic if and only if for all $t_i \in T, p_j \in P : D^-_{ij} = \alpha$ with $\alpha \in \mathbb{N}$, or $D^-_{ij} = \mathbf{m}(p_j)$.

Proof. $\Rightarrow$ Suppose that it is not the case, that is $\mathcal{P}$ is strongly monotonic and there exist $t_i \in T, p_j \in P$ such that D^-_{ij} is not of the form α with $\alpha \in \mathbb{N}$ or $\mathbf{m}(p_j)$. Let $D^-_{ij} = \sum_{p_k \in P} \beta_k \cdot \mathbf{m}(p_k) + \alpha$. We consider two cases:

1. $\beta_j > 1$ or $\beta_j = 1$ and $\alpha > 0$. In both cases, t_i is unfirable, which contradicts the hypothesis.
2. $\beta_j = 0$ or $\beta_j = 1$ and $\alpha = 0$. Since D^-_{ij} is not of the form α or $\mathbf{m}(p_j)$, there is $k' \neq j$ such that $\beta_{k'} > 0$. By hypothesis, t_i is firable from at least one marking $\mathbf{m}$. Let us construct the marking $\mathbf{m}'$ as follows: $\forall p_k \neq p_{k'} \in P : \mathbf{m}'(p_k) = \mathbf{m}(p_k)$, and $\mathbf{m}'(p_{k'}) = \mathbf{m}(p_{k'}) + \mathbf{m}(p_j) + 1$. By construction, $\mathbf{m} \preccurlyeq \mathbf{m}'$ but t_i is not firable from $\mathbf{m}'$. Indeed, for t_i to be firable we should have $\mathbf{m}'(p_j) = \mathbf{m}(p_j) \geq D^-_{ij}(\mathbf{m}') \geq \beta_{k'} \cdot \big(\mathbf{m}(p_{k'}) + \mathbf{m}(p_j) + 1\big)$. Since $\beta_{k'} > 0$, this is not possible. We conclude that $\mathcal{P}$ is not strongly monotonic.

In both cases, we obtain a contradiction.

$\Leftarrow$ We proceed by contradiction. Suppose that $\mathcal{P}$ is not strongly monotonic but for all $t_i \in T, p_j \in P : D^-_{ij} = \alpha$ with $\alpha \in \mathbb{N}$ or $D^-_{ij} = p_j$. Hence there exists three markings $\mathbf{m}_1, \mathbf{m}_2$ and $\mathbf{m}_3$ and a transition t_i such that $\mathbf{m}_1 \rightarrow_{t_i} \mathbf{m}_3$, $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$ and there do not exist a marking $\mathbf{m}_4$ such that $\mathbf{m}_2 \rightarrow_{t_i} \mathbf{m}_4$ and $\mathbf{m}_3 \preccurlyeq \mathbf{m}_4$.

Since $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$ and $\mathbf{m}_1(p_j) \geq D^-_{ij}(\mathbf{m}_1)$ for all $p_j \in P$, $\mathbf{m}_2(p_j) \geq D^-_{ij}(\mathbf{m}_2)$ for all $p_j \in P$. As a consequence, t_i is firable from $\mathbf{m}_2$. Suppose that $\mathbf{m}_2 \rightarrow_{t_i} \mathbf{m}_4$.

Let $\mathbf{m}'_k$ ($k \in \{1, 2\}$) be such that $\mathbf{m}'_k(p_j) = \mathbf{m}_k(p_j) - D^-_{ij}(\mathbf{m}_k)$ for all $p_j \in P$. Since $\mathbf{m}_1 \preccurlyeq \mathbf{m}_2$, $\mathbf{m}'_1 \preccurlyeq \mathbf{m}'_2$. Moreover, we have that $D^+_{ij}(\mathbf{m}_1) \leq D^+_{ij}(\mathbf{m}_2)$ for all j such that $1 \leq j \leq |P|$. Since $\mathbf{m}_3(p_j) = \mathbf{m}'_1(p_j) + D^+_{ij}(\mathbf{m}_1)$ and $\mathbf{m}_4(p_j) = \mathbf{m}'_2(p_j) + D^+_{ij}(\mathbf{m}_2)$ for all $p_j \in P$, we conclude that $\mathbf{m}_3 \preccurlyeq \mathbf{m}_4$ and we obtain a contradiction.

Proof of Lemma 6 To establish Lemma 6, we first define a new notation and establish two auxiliary lemmata. Given a (possibly extended) marking $\mathbf{m}$ over set of places P and $k \in \mathbb{N}$, we define $\mathsf{Bound}(\mathbf{m}, k) : (\mathbb{N} \cup \{+\infty\})^{|P|} \rightarrow (\mathbb{N} \cup \{+\infty\})^{|P|}$ such that for all place $p_i \in P$: $\mathsf{Bound}(\mathbf{m}, k)(p_i) = \mathbf{m}(p_i)$ if $\mathbf{m}(p_i) \leq k$, $\mathsf{Bound}(\mathbf{m}, k)(p_i) = +\infty$ otherwise.

The first lemma shows that the way we have extended the transition relation is well-suited in the following sense. Let $\mathbf{m}$ and $\mathbf{m}'$ be two (extended) markings such that $\mathbf{m} \rightarrow_t \mathbf{m}'$ for some transition t. Then $\gamma(\mathbf{m}')$ is the most precise downward closed overapproximation for $\mathsf{Post}(\gamma(\mathbf{m}), t)$.

Lemma 7. *Let $\mathcal{P}$ be a strongly monotonic* SMPN *with set of transitions T and $\mathbf{m}, \mathbf{m}'$ be two (possibly extended) markings. If $\mathbf{m} \to_{t_i} \mathbf{m}'$ for some $t_i \in T$, then $\gamma(\mathbf{m}')$ has the two following properties:* [covering] $\mathsf{Post}(\gamma(\mathbf{m}), t_i) \subseteq \gamma(\mathbf{m}')$ *and* [preciseness] *there is no $S \subseteq \mathcal{L} \cup \mathbb{N}^{k_P}$ such that* $\mathsf{Post}(\gamma(\mathbf{m}), t_i) \subseteq \gamma(S) \subset \gamma(\mathbf{m}')$.

Proof. **(Covering)** Suppose that the covering property is not verified. In this case, there exist four (possibly extended) markings $\mathbf{m}, \mathbf{m}', \mathbf{n}$ and $\mathbf{n}'$, and a transition $t_i \in T$ such that such that $\mathbf{m} \to_{t_i} \mathbf{m}'$, $\mathbf{n} \to_{t_i} \mathbf{n}'$, $\mathbf{n} \in \gamma(\mathbf{m})$ and $\mathbf{n}' \notin \gamma(\mathbf{m}')$. Hence, there exists $p_j \in P$ such that $\mathbf{n}'(p_j) > \mathbf{m}'(p_j)$.

Following Lemma 3, the effect of transition t_i on place p_j for a marking $\mathbf{l}$, i.e. $D^+_{ij}(\mathbf{l}) - D^-_{ij}(\mathbf{l})$, may be of two forms. Either $D^+_{ij}(\mathbf{l}) - D^-_{ij}(\mathbf{l}) = \sum_{p_k \in P} \beta_k \cdot \mathbf{l}(p_k) + \alpha$ or $D^+_{ij}(\mathbf{l}) - D^-_{ij}(\mathbf{l}) = \sum_{p_k \in P} \beta_k \cdot \mathbf{l}(p_k) + \alpha - \mathbf{l}(p_j)$ with $\beta_k \in \mathbb{N}$ for all k and $\alpha \in \mathbb{Z}$. Hence, either $\mathbf{n}'(p_j) = \mathbf{n}(p_j) + \sum_{p_k \in P} \beta_k \cdot \mathbf{n}(p_k) + \alpha$ and $\mathbf{m}'(p_j) = \mathbf{m}(p_j) + \sum_{p_k \in P} \beta_k \cdot \mathbf{n}(p_k) + \alpha$, or $\mathbf{n}'(p_j) = \sum_{p_k \in P} \beta_k \cdot \mathbf{n}(p_k) + \alpha$ and $\mathbf{m}'(p_j) = \sum_{p_k \in P} \beta_k \cdot \mathbf{m}(p_k) + \alpha$. In both cases, since $\mathbf{n} \in \gamma(\mathbf{m})$, $\mathbf{n}(p_k) \leq \mathbf{m}(p_k)$ for all $p_k \in P$, hence $\sum_{p_k \in P} \beta_k \cdot \mathbf{n}(p_k) + \alpha \leq \sum_{p_k \in P} \beta_k \cdot \mathbf{m}(p_k) + \alpha$. We conclude that $\mathbf{n}'(p_j) \leq \mathbf{m}'(p_j)$ and we obtain a contradiction.

(Preciseness) In order to establish the preciseness property, we prove that if $\mathbf{m} \to_{t_i} \mathbf{m}'$, then any marking $\mathbf{n} \in \gamma(\mathbf{m}')$ is covered by a marking $\mathbf{n}' \in \mathsf{Post}(\gamma(\mathbf{m}), t_i)$. This clearly implies that the set $\gamma(\mathbf{m}')$ is the minimal downward closed set that contains $\mathsf{Post}(\gamma(\mathbf{m}), t_i)$, since for any downward closed set $D \subset \gamma(\mathbf{m}')$, there exists at least one marking $\mathbf{n} \in \mathsf{Post}(\gamma(\mathbf{m}), t_i)$ that is not in D. The proof is by contradiction. Suppose that it is not the case, thus there exists $\mathbf{n} \in \gamma(\mathbf{m}')$ such that there is no $\mathbf{n}'' \in \mathsf{Post}(\gamma(\mathbf{m}), t_i)$ with $\mathbf{n} \preccurlyeq \mathbf{n}''$.

Let c be such that $c > \max\{|\alpha_1|, \ldots, |\alpha_{k_P}|\}$ where α_j is the constant term in $D^+_{ij} - D^-_{ij}$. We first construct the marking $\mathbf{n}'$ in the following manner: $\mathbf{n}'(p_j) = \mathbf{m}(p_j)$ if $\mathbf{m}(p_j) \in \mathbb{N}$; otherwise $\mathbf{n}'(p_j) > \max\{\mathbf{n}(p_k) \mid p_k \in P\} + c$. By construction, $\mathbf{n}' \in \gamma(\mathbf{m})$ and t_i is firable from $\mathbf{n}'$. Let $\mathbf{n}' \to_{t_i} \mathbf{n}''$. From the covering property, $\mathbf{n}'' \in \gamma(\mathbf{m}')$. Let us show that $\mathbf{n} \preccurlyeq \mathbf{n}''$.

For all $p_j \in P$, two cases hold following Lemma 3 again:

- $D^+_{ij}(\mathbf{m}) - D^-_{ij}(\mathbf{m}) = \sum_{p_k \in P} \beta_k \cdot \mathbf{m}(p_k) + \alpha_j - \mathbf{m}(p_j)$ with $\beta_k \in \mathbb{N}$ for all k and $\alpha_j \in \mathbb{Z}$. Either $\beta_k > 0$ implies that $\mathbf{m}(p_k) \in \mathbb{N}$ for all k. In that case, $\mathbf{n}''(p_j) = \mathbf{m}'(p_j)$. Hence, $\mathbf{n}(p_j) \leq \mathbf{n}''(p_j)$. Or there is some $p_k \in P$ such that $\beta_k > 0$ and $\mathbf{m}(p_k) = +\infty$. By construction, $\mathbf{n}''(p_j) > \max\{\mathbf{n}(p_k) \mid p_k \in P\}$, hence $\mathbf{n}(p_j) < \mathbf{n}''(p_j)$;
- $D^+_{ij}(\mathbf{m}) - D^-_{ij}(\mathbf{m}) = \sum_{p_k \in P} \beta_k \cdot \mathbf{m}(p_k) + \alpha_j$ with $\beta_k \in \mathbb{N}$ for all k and $\alpha_j \in \mathbb{Z}$. By using a similar reasoning than in the previous case, we obtain that $\mathbf{n}(p_j) \leq \mathbf{n}''(p_j)$.

We conclude that $\mathbf{n} \preccurlyeq \mathbf{n}''$ and we obtain a contradiction.

The second auxiliary lemma is easy to establish and its proof will be omitted.

Lemma 8. *Given any $i \in \mathbb{N}$, let S_i and L_i be constructed following Definition 8 and $\mathbf{m} \in S_i \cup L_i$, there does not exist a finite set $S \subseteq S_i \cup L_i$ such that $\gamma(\mathbf{m}) \subseteq \gamma(S)$ and $\gamma(\mathsf{Bound}(\mathbf{m}, i)) \not\subseteq \gamma(S)$.*

We can now give the proof of Lemma 6, that we first recall.

Lemma 6 Given a SMPN $\mathcal{P} = \langle P, T, D^-, D^+ \rangle$ with the adequate domain of limits $\langle \mathcal{L}, \preccurlyeq_e, \gamma(.) \rangle$ and the sets $S_i \subseteq \mathbb{N}^{k_P}$ and $L_i \subseteq \mathcal{L}$ constructed following Definition 8, any pair $\langle S_i, L_i \rangle$ is a perfect pair.

Proof. Let us first define $\mathfrak{Post}(\mathbf{m}, i)$ as the set of maximal elements of $\cup_{t_k \in T} \mathsf{Bound}(\mathsf{Post}(\mathbf{m}, t_k), i)$. Following the definition of a perfect pair (Definition 7), we show that for any (extended) marking $\mathbf{m} \in S_i \cup L_i$, $\mathfrak{Post}(\mathbf{m}, i)$ is the unique, minimal and *most precise* subset of $S_i \cup L_i$ to cover $\mathsf{Post}(\gamma(\mathbf{m}))$.

From Lemma 8 and Lemma 7, we have that for any $i \geq 0$ and $\mathbf{m} \in S_i \cup L_i : \mathfrak{Post}(\mathbf{m}, i)$ is such that $\mathsf{Post}(\gamma(\mathbf{m})) \subseteq \gamma(\mathfrak{Post}(\mathbf{m}, i))$. Moreover, for all $\mathbf{m}_1, \mathbf{m}_2 \in \mathfrak{Post}(\mathbf{m}, i) : \mathbf{m}_1 \neq \mathbf{m}_2$ implies $\mathbf{m}_1 \not\preccurlyeq_e \mathbf{m}_2$. Let us prove that there does not exist $L \subseteq L_i \cup S_i$ such that $L \neq \mathfrak{Post}(\mathbf{m}, i), \mathsf{Post}(\gamma(\mathbf{m})) \subseteq \gamma(L), \forall \mathbf{m}_1, \mathbf{m}_2 \in L : \mathbf{m}_1 \neq \mathbf{m}_2$ implies $\mathbf{m}_1 \not\preccurlyeq_e \mathbf{m}_2$, and $\nexists L' \subseteq S_i \cup L_i : \mathsf{Post}(\gamma(\mathbf{m})) \subseteq \gamma(L') \subset \gamma(L)$. If such a set L does not exist, we conclude that there does not exist $L' \subseteq S_i \cup L_i : \mathsf{Post}(\gamma(\mathbf{m})) \subseteq L' \subset \mathfrak{Post}(\mathbf{m}, i)$, hence $\gamma(\mathfrak{Post}(\mathbf{m}, i))$ is a most precise downward-closed overapproximation of $\mathsf{Post}(\gamma(\mathbf{m}))$, and $\mathfrak{Post}(\mathbf{m}, i)$ is the unique most precise downward-closed overapproximation of $\mathsf{Post}(\gamma(\mathbf{m}))$. Hence, we conclude that any pair $\langle S_i, L_i \rangle$ is a perfect pair.

We exhibit a reasoning by contraposition. Suppose that there exists $L \subseteq S_i \cup L_i$ such that $L \neq \mathfrak{Post}(\mathbf{m}, i), \mathsf{Post}(\gamma(\mathbf{m})) \subseteq \gamma(L), \forall \mathbf{m}_1, \mathbf{m}_2 \in L : \mathbf{m}_1 \neq \mathbf{m}_2$ implies $\mathbf{m}_1 \not\preccurlyeq_e \mathbf{m}_2, \nexists L' \subseteq L_i \cup S_i : \mathsf{Post}(\gamma(\mathbf{m})) \subseteq \gamma(L') \subset \gamma(L)$. From Corollary 6 of [25], we have that for any downward-closed set $D \subseteq \mathbb{N}^k$, there exists an unique $L' \subseteq (\mathbb{N} \cup \{+\infty\})^k$ such that for all $\mathbf{m}_1, \mathbf{m}_2 \in L'$: $\mathbf{m}_1 \neq \mathbf{m}_2$ implies $\mathbf{m}_1 \not\preccurlyeq_e \mathbf{m}_2$, and $\gamma(L') = D$. Since $\mathfrak{Post}(\mathbf{m}, i) \neq L$, $\gamma(\mathfrak{Post}(\mathbf{m}, i)) \neq \gamma(L)$. Hence, $\gamma(\mathfrak{Post}(\mathbf{m}, i)) \not\subseteq \gamma(L)$, since $\gamma(\mathfrak{Post}(\mathbf{m}, i)) \not\subset \gamma(L)$ by hypothesis, and there exists $\mathbf{n} \in \mathfrak{Post}(\mathbf{m}, i)$ such that for all $\mathbf{n}' \in L$ we have $\mathbf{n} \not\preccurlyeq_e \mathbf{n}'$. Suppose that $\mathbf{n} = \mathsf{Bound}(\mathsf{Post}(\gamma(\mathbf{m}), t_j), i)$ for some $t_j \in T$. From Lemma 7 and Lemma 8, we have that there does not exist $S \subseteq S_i \cup L_i$ such that $\mathsf{Post}(\gamma(\mathbf{m}), t_j) \subseteq \gamma(S)$ and $\gamma(\mathbf{n}) \not\subseteq \gamma(S)$. Hence, since $\gamma(\mathsf{Post}(\gamma(\mathbf{m}), t_j)) \subseteq \gamma(L)$, we have that $\gamma(\mathbf{n}) \subseteq \gamma(L)$, i.e. there exists $\mathbf{n}' \in L : \mathbf{n} \preccurlyeq_e \mathbf{n}'$ and we obtain a contradiction.

Proof of Theorem 5 $(\mathcal{L}(\Sigma, Q), \overline{\sqsubseteq}, \gamma)$ is an adequate domain of limits for $(\mathcal{S}_{\mathcal{C}}, \precsim)$.

Proof. (i) It is easy to show that for any $\langle q, w \rangle \in \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$, $\gamma(w)$ is $\precsim$-downward closed (see [2]), (ii) the element $\top$ is such that $\gamma(\top)$ is the whole set of states $\mathcal{S}_{\mathcal{C}}$, (iii) by definition $w_1 \overline{\sqsubseteq} w_2$ if and only if $\gamma(w_1) \subseteq \gamma(w_2)$ for all $w_1, w_2 \in \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$, and (iv) from Theorem 1 of [2] we deduce that if $S \subseteq \mathcal{S}_{\mathcal{C}}$ is $\precsim$-downward closed, then there exists $S' \subseteq \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$ such that S' is finite and $\gamma(S') = S$.

Proof of Theorem 6 Any LCS $\mathcal{C}$ with the adequate domain of limits $(\mathcal{L}(\Sigma), \overline{\sqsubseteq}, \gamma)$ are effective.

Proof. (i) it is easy to show that $\mathcal{S}_{\mathcal{C}}$ and $\mathcal{L}(\Sigma)$ are recursively enumerable, (ii) it is shown in [2] that the transition relation of LCS is decidable, (iii) it is shown in [2] how to compute an operator that returns, given $w \in \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$, $w' \in \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$ such that $\gamma(w') = \mathsf{Post}(\gamma(w))$. By using that operator and since $\overline{\sqsubseteq}$ is decidable following [2], we conclude that we can decide whether $\mathsf{Post}(\gamma(w)) \subseteq \gamma(w')$ for any $w, w' \in \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$, (iv) as noticed in the previous point, an algorithm is given in [2] to decide whether $w_1 \overline{\sqsubseteq} w_2$ for any $w_1, w_2 \in \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$. Moreover, for any $S_1, S_2 \subseteq \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$, $\gamma(S_1) \subseteq \gamma(S_2)$ if and only if for all $c \in S_1$, there exists $c' \in S_2$ such that $c \precsim c'$ (see [2] for proofs). Hence, we can decide for any finite sets $S_1, S_2 \subseteq \mathcal{S}_{\mathcal{C}} \cup \mathcal{L}(\Sigma)$ whether $\gamma(S_1) \subseteq \gamma(S_2)$.

B Practical results

We have built a (naive) prototype that implements the algorithm presented at Fig. 2 and applied it to the verification of counting abstractions of multi-threaded JAVA programs [11, 25].

We compare those results with the execution times obtained by applying the standard backward algorithm for well-structured transition systems [1]. The comparison is fair since we have built both prototypes around the same data-structure called Covering Sharing Trees (see [25]). The results are reported in Table 1.

case study	**P**	**T**	Post	Pre
Consprod	18	16	0.37s	30.75s
Consprod2	18	16	0.36s	1.36s
delegate_notifyAll	50	54	1.82s	>10h
example_lea	48	44	17.14s	>10h
lea_basic_approach	16	14	0.36s	0.35s
lea_conflict_set	30	37	9.8s	121.45s
Queued_busy_flag	82	105	10m23s	>10h
simple_Java_example	30	37	38.87s	6.96s
transthesis	90	118	47m	37m

Table 1. Results obtained on a Pentium 4 3GHz with 4GB of memory. **P** : number of places, **T** : number of transitions, Post : execution time of algorithm at Figure 2, Pre : execution time of the classical backward approach [1].