Petri Nets with Non-blocking Arcs are Difficult to Analyze

Jean-François Raskin [1,2] and Laurent Van Begin [3,4]

Université Libre de Bruxelles
Blvd du Triomphe, 1050 Bruxelles, Belgium.

Abstract

In this paper, we study the decidability of five problems on a class of extended Petri nets. The study of this class of extended Petri nets is motivated by the problem of parametric verification of multiple copies of processes that can communicate with a *partially non-blocking rendez-vous*. This kind of communications occurs in abstractions of multi-threaded JAVA programs.

Key words: Monotonic Extensions of Petri Nets, Decidability/ Undecidability.

1 Introduction

In parametric verification, we want to verify at once an entire family of systems. For example, some mutual exclusion protocols have been designed to work for any number of (identical) processes. The verification of such protocols for specific number of processes is not satisfactory. We want a proof for any number of those processes. This problem of parametric verification is difficult and has been shown undecidable [AK86] in general. To obtain partial automatic methods, several abstractions have been shown useful. The work in this paper is directly connected to the context of one of these abstractions, the so-called *counting abstraction* [GS92].

When considering counting abstractions, (infinite) Petri nets and their extensions are particularly important. In that context, processes of a parametric system are abstracted by tokens, places are used to count the number of processes in each local state of the parametric system and transitions are used

[1] Email: `jraskin@ulb.ac.be`
[2] This author was partially supported by the FRFC grant 2.4530.02.
[3] Email: `lvbegin@ulb.ac.be`
[4] This author was supported by a "First Europe" grant EPH3310300R0012 of the Walloon Region.

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

to model the dynamics of the processes. Sistla et al [GS92] have shown that Petri nets are well suited to abstract parametric systems where rendez-vous communications are used for synchronizations between processes. When the underlying systems use more "exotic" communication mechanisms, like broadcast communications for example, the model of Petri nets has to be extended, with transfer arcs for instance, see [DRVB02] for more details.

In this paper, we consider a very simple extension of Petri nets that is able to model parametric systems that uses "partially non-blocking" rendez-vous synchronizations in addition to classical (blocking) rendez-vous synchronizations. Partially non-blocking rendez-vous are asymmetric synchronizations where the sending part is *not blocking* (contrary to the usual case) and the receiving part is *blocking* (as in the usual case). To illustrate the notion of *partially non-blocking rendez-vous*, consider Figure 1. This figure represents

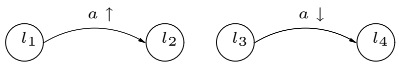

Fig. 1. Example of partially non-blocking rendez-vous.

fragments of two processes. In location l_1, the first process can emit $a\uparrow$, the proposition of a rendez-vous on symbol a and moves to l_2 even if the second process is not present to synchronize on a by emitting $a\downarrow$. If the second process can synchronize then it does. On the other hand, the second process has to synchronize with the first process in order to emit $a\downarrow$ and move from l_3 to l_4, it cannot move alone. So the emission $a\uparrow$ is non-blocking and can occur without a reception part $a\downarrow$ while the reception is blocking and can only occur in the presence of the emission. This is why we call such rendez-vous "partially non-blocking rendez-vous". In this paper, we will define a simple extension of the basic Petri nets that is able to model this kind of communications between processes, we call this extension *Petri nets with non-blocking arcs*.

The study of this simple extension of Petri nets is motivated by previous works by the authors on extensions of Petri nets for modeling communications in multi-threaded programs [DRVB02]. Multi-threaded JAVA programs use instructions like `notify` and `notifyAll` for synchronizations. The instruction `notify` can be modeled by an partially non-blocking rendez-vous and the instruction `notifyAll` can be modeled by a broadcast communication. While transfer nets, that are able to model broadcast communications, have been studied from a theoretical point of view [EFM99,DFS98], this is not the case for extensions of Petri nets modeling partially non-blocking rendez-vous. We study here the decidability of five important problems in the context of Petri nets extended with non-blocking arcs. While those five problems are decidable for Petri nets, we show here that only two of them remain decidable in the extended model.

The rest of the paper is organized as follows. In a second section, we recall some basic notions and notations. In a third section, we introduce Petri nets extended with non-blocking arcs and the five problems that we study. In a fourth section, we show that two of the problems remain decidable on the extended model. In a fifth section, we establish the undecidability of the three other problems.

Finally, a last section draws some conclusions.

2 Preliminaries

Multi-sets.

A multi-set B constructed from a set S of n elements is a function $B : S \to \mathbb{N}$ that assigns to each element s of S the number $B(s)$ of occurrences of s in the multi-set. To denote a multi-set S over $S = \{s_1, \ldots, s_n\}$, we write $\{(s_i, B(s_i)) | B(s_i) > 0\}$. For example, let S be $\{s_1, s_2, s_3\}$ and let B be such that $B(s_1) = 3$, $B(s_2) = 0$, $B(s_3) = 1$, then B is denoted by $\{(s_2, 3), (s_3, 1)\}$. Equivalently, B can be represented as a n-dimensional vector, denoted $\mathbf{vec}(B)$, and defined as follows:

$$\mathbf{vec}(B) = \begin{bmatrix} B(s_1) \\ B(s_2) \\ \ldots \\ B(s_n) \end{bmatrix}$$

Well quasi orderings, well structured transition systems.

A *well quasi ordering* $\preccurlyeq$ on the elements of a set S is a reflexive and transitive relation such that for any infinite sequence $s_1 s_2 \ldots$ where $s_i \in S$ $(i \geq 1)$ there is $i < j$ such that $s_i \preccurlyeq s_j$. In the following we note $s_i \prec s_j$ if $s_i \preccurlyeq s_j$ but $s_j \not\preccurlyeq s_i$. As an example, it is well know that the quasi order $\preccurlyeq$ on elements in $\mathbb{N}^k$ defined as $m \preccurlyeq m'$ if $m_i \leq m'_i$ for any $1 \leq i \leq k$ is a well quasi ordering. In the rest of this paper, we consider this quasi order on vectors of naturals.

A *transition system* is a tuple $\langle L, \to \rangle$ where L is a set of states and $\to \subseteq L \times L$. $\langle l_1, l_2 \rangle \in \to$ is noted $l_1 \to l_2$. A transition system $\langle L, \to \rangle$ is *monotonic* according to the well quasi ordering $\preccurlyeq$ on the elements of L if for all l_1, l_2 in L with $l_1 \preccurlyeq l_2$, if $l_1 \to l'_1$ then there exists $l'_2 \succcurlyeq l'_1$ with $l_2 \to l'_2$. A transition system $\langle L, \to \rangle$ is *strictly monotonic* according to the well quasi ordering $\preccurlyeq$ on the elements of L if it is monotonic and for all l_1, l_2 in L with $l_1 \prec l_2$, if $l_1 \to l'_1$ then there exists $l'_2 \succ l'_1$ with $l_2 \to l'_2$. Systems that are monotonic for a well-quasy order $\preccurlyeq$ are called *well structured transition systems* in [FS01]. For those systems, several general decidability results are known [ACJT96,FS01]. We will use those results in section 4 to derive the decidability of two problems on our extended model of Petri nets.

A *two-counter machine* C, $\mathsf{2CM}$ for short, is a tuple $\langle c_1, c_2, L, \mathsf{Instr} \rangle$ where:

- c_1, c_2 are two counters taking their values in $\mathbb{N}$;

- $L = \{l_1, l_2, \ldots, l_u\}$ is a finite non-empty set of u locations;
- Instr is a function that labels each location $l \in L$ with an instruction that has one of the three following forms:
  - $\cdot$ $l : c_j := c_j + 1;\ \texttt{goto}\ l';$, where $j \in \{1, 2\}$ and $l' \in L$, this is called an increment, and we define $\mathsf{TypeInst}(l) = \texttt{inc}_\texttt{j}$;
  - $\cdot$ $l : c_j := c_j - 1;\ \texttt{goto}\ l';$, where $j \in \{1, 2\}$ and $l' \in L$, this is called a decrement, and we define $\mathsf{TypeInst}(l) = \texttt{dec}_\texttt{j}$;
  - $\cdot$ $l : \texttt{if}\ c_j = 0\ \texttt{then goto}\ l'\ \texttt{else goto}\ l'';$, where $j \in \{1, 2\}$ and $l', l'' \in L$, this is called a zero-test, and we define $\mathsf{TypeInst}(l) = \texttt{zerotest}_\texttt{j}$.

Those instructions have their usual obvious semantics, in particular, decrement can only be done if the value of the counter is strictly greater than zero.

A *configuration* of a 2CM $\langle c_1, c_2, L, \mathsf{Instr}\rangle$ is a tuple $\langle loc, v^1, v^2\rangle$ where $loc \in L$ is the value of the program counter and v^1, respectively v^2, is a natural number that gives the valuation of the counter c_1, respectively c_2. A *computation* γ of a 2CM $\langle c_1, c_2, L, \mathsf{Instr}\rangle$ is either a finite sequence of configurations $\langle loc_1, v_1^1, v_1^2\rangle, \langle loc_2, v_2^1, v_2^2\rangle, \ldots, \langle loc_r, v_r^1, v_r^2\rangle$, or an infinite sequence of configurations $\langle loc_1, v_1^1, v_1^2\rangle, \langle loc_2, v_2^1, v_2^2\rangle\ , \ldots, \langle loc_r, v_r^1, v_r^2\rangle, \ldots$ such that : (i) "Initialization": $loc_1 = l_1$, $v_1^1 = 0$, and $v_1^2 = 0$, i.e. a computation starts in l_1 and the two counters have the value zero; (ii) "Consecution": for each $i \in \mathbb{N}$ such that $1 \leq i \leq |\gamma|$ we have that $\langle loc_{i+1}, v_{i+1}^1, v_{i+1}^2\rangle$ is the configuration obtained from $\langle loc_i, v_i^1, v_i^2\rangle$ by applying the instruction $\mathsf{Instr}(loc_i)$. In the finite case, r is the *length* of the computation γ and we define $\mathsf{final}(\gamma) = \langle loc_r, v_r^1, v_r^2\rangle$. If γ is a computation, γ_i denotes the i^{th} configuration of γ. A configuration $\langle loc, v^1, v^2\rangle$ is *reachable* in the 2CM $\langle c_1, c_2, L, \mathsf{Instr}\rangle$, if there exists a finite computation γ such that $\mathsf{final}(\gamma) = \langle loc, v^1, v^2\rangle$.

The *reachability problem for* 2CM is defined as follows: "Given a 2CM $C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$ and a configuration $\langle loc, v^1, v^2\rangle$ of C, is $\langle loc, v^1, v^2\rangle$ reachable from $\langle l_1, 0, 0\rangle$?". The *boundedness problem for* 2CM is defined as follows: "Given a 2CM $C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$, is there $c \in \mathbb{N}$ such that for all reachable configuration $\langle loc, v_1, v_2\rangle$ in C we have $v_1 + v_2 \leq c$?"

It is well-known that those two problems cannot be answered completely with an algorithm.

Theorem 2.1 (From [Min67]) *The reachability and boundedness problems are undecidable for* 2CM.

3 Petri nets extended with non-blocking arcs

In this section, we introduce formally the class of extended Petri nets that we call *Petri nets with non-blocking arcs.*

Definition 3.1 A *Petri Net with non-blocking arcs* $\mathcal{N}$, PN+NBA for short, is defined by a pair $\mathcal{N} = \langle \mathcal{P}, \mathcal{T}\rangle$ where $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ is a finite set of n places and $\mathcal{T} = \{tr_1, tr_2, \ldots, tr_m\}$ is a finite set of m transitions where each

$tr_i \in \mathcal{T}$ is a tuple $\langle I, O, A\rangle$, where I is a multi-set of input places in $\mathcal{P}$, O is a multi-set of output places in $\mathcal{P}$, and A the non-blocking part of the transition is either the empty set or a singleton $\{\langle p, q\rangle\}$ with $p, q \in \mathcal{P} \setminus \{r \mid (r, i) \in I \text{ or } (r, i) \in O, \text{ and } i \geq 1\}$, p and q are called respectively the source and the target of the non-blocking part.

A marking of a PN+NBA $\mathcal{N} = \langle \mathcal{P}, \mathcal{T}\rangle$ is a function $m : \mathcal{P} \to \mathbb{N}$ that assigns to each place $p \in \mathcal{P}$ a natural number $m(p)$. Equivalently, a marking m can be seen as a n-dimensional vector of natural numbers. In the following, for a marking m and a set of places S, we will write $m(S)$ for $\sum_{p\in S} m(p)$.

Figure 2 shows an example of PN+NBA. Circles represent places and filled rectangles represent transitions. Plain edges from places p to transitions tr are labeled by the number of occurrences of p in the input multi-set of tr and plain edges from transitions tr to places p are labeled by the number of occurrences of p in the output multi-set of tr. Absence of edge from (to) a place p to (from) a transition tr means that there is no occurrence of p in the input (output) multi-set of tr. In the following, when there is only one occurrence of a place into a given multi-set of a transition we will only use edges without labels. Pairs of dashed edges from a place to a transition and from this transition to a place represent the non-blocking part of the transition. Tokens in the places define markings in the usual way.

A transition $tr = \langle I, O, A\rangle$ is *firable* in a marking m iff $m \succcurlyeq \mathbf{vec}(I)$. Note that the non-blocking part is not taken into account to decide if a transition tr is firable in a marking m or not. Given a marking m and a transition $tr = \langle I, O, A\rangle$ that is firable in m, we say that m leads to m' by firing tr, noted $m \to^{tr} m'$ where m' is defined as:

- if $A = \{\langle p, q\rangle\}$ and $m(p) \geq 1 : m' = m - \mathbf{vec}(I) + \mathbf{vec}(O) - \mathbf{vec}(\{(p, 1)\}) + \mathbf{vec}(\{(q, 1)\})$, that is the input places are decremented by their number of occurrences in I, the output places are incremented by their number of occurrences in O and one token moves from the source place to the target place of the non-blocking part.
- otherwise: $m' = m - \mathbf{vec}(I) + \mathbf{vec}(O)$. In that case, either there is no non-blocking part to the transition and the effect of the transition is as in the usual Petri net case or the source of the non-blocking part p is not marked and the non-blocking part has no effect.

A computation η of a PN+NBA $\mathcal{N} = \langle \mathcal{P}, \mathcal{T}\rangle$ is a sequence of markings alternating with transitions $\eta = m_1 tr_1 m_2 tr_2 \ldots tr_{r-1} m_r$ where m_i is a marking for any $i \in \{1, 2, \ldots, r\}$, $tr_j \in \mathcal{T}$ for any $j \in \{1, 2, \ldots, r-1\}$ and we have that $m_1 \to^{tr_1} m_2 \to^{tr_2} \ldots \to^{tr_{r-1}} m_r$. This notion of computation is extended to the infinite case as usual. A sequence of transitions $\sigma = tr_1 tr_2 \ldots tr_r$ is firable in a marking m_1 if there exists a sequence of markings $m_1 m_2 \ldots m_{r+1}$ such that $m_1 tr_1 m_2 tr_2 \ldots tr_r m_{r+1}$ is a computation of $\mathcal{N}$. We note $m \to^{\sigma} m'$ the fact that firing σ from m leads to m'. A marking m' is *reachable* from a marking m in $\mathcal{N}$ iff there exists a sequence of transitions σ of $\mathcal{N}$ such that

$m \rightarrow^{\sigma} m'$. We note $\mathsf{Reach}(\mathcal{N}, m)$ the set of markings that are reachable from m in $\mathcal{N}$, i.e. $\mathsf{Reach}(\mathcal{N}, m) = \{m' | \exists \sigma \in \mathcal{T}^* : m \rightarrow^{\sigma} m'\}$.

A labeled PN+NBA is a tuple $\langle \mathcal{P}, \mathcal{T}, \mathcal{L} \rangle$ where $\mathcal{P}$ and $\mathcal{T}$ are a set of places and a set of transitions as before and $\mathcal{L} : \mathcal{T} \rightarrow \Sigma$ is a labeling function that labels each transition $tr \in \mathcal{T}$ with the label $\mathcal{L}(tr)$ from a finite set of labels Σ. The notion of computation is as before. To each of those computations $\eta = m_1 tr_1 m_2 tr_2 \ldots m_r tr_r \ldots$ we associate the sequence of labels $\mathcal{L}(\eta) = \mathcal{L}(tr_1)\mathcal{L}(tr_2)\ldots\mathcal{L}(tr_n)\ldots$ For a PN+NBA $\mathcal{N}$ and a marking m, we define $\mathcal{L}(\mathcal{N}, m) = \{\mathcal{L}(\eta) | \eta$ is an infinite computation of $\mathcal{N}$ with initial marking $m\}$. The formula of the logic LTL are evaluated over those sequences of labels. Given a set of labels Σ, the formulas of the logic LTL are defined by the following rule:

$$\phi := \lambda | \neg\phi | \phi_1 \vee \phi_2 | \bigcirc \phi | \Box\phi | \Diamond\phi | \phi_1 \mathcal{U} \phi_2$$

where $\lambda \in \Sigma$. We only give the semantics for the $\Box$ and $\Diamond$ operators because they are the only ones that we need in this paper. For $\Lambda \in \Sigma^\omega$ such that $\Lambda = \lambda_1 \ldots \lambda_i \lambda_{i+1} \ldots$, we note Λ^i for the suffix $\lambda_i \lambda_{i+1} \ldots$ of Λ starting at this index i and $\Lambda(i)$ for for the i^{th} element in Λ. Given $\Lambda \in \Sigma^\omega$ and a formula ϕ, we define the satisfaction relation, noted $\models$, as follows :

- if $\phi = \lambda$, then $\Lambda \models \phi$ iff $\Lambda(1) = \lambda$;
- if $\phi = \Diamond\varphi$, then $\Lambda \models \phi$ iff $\exists i \geq 1 : \Lambda^i \models \varphi$;
- if $\phi = \Box\varphi$, then $\Lambda \models \phi$ iff $\forall i \geq 1 : \Lambda^i \models \varphi$.

For a set of infinite sequence of labels M and a formula ϕ, we have $M \models \phi$ if for all $\Lambda \in M$ we have $\Lambda \models \phi$.

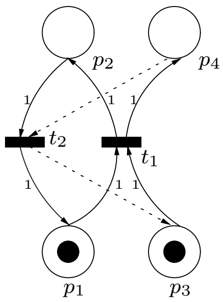

Fig. 2. a Petri net with non-blocking arcs.

The PN+NBA of Figure 2 has two transitions. The transitions t_1 is a classical Petri net transition while t_2 has an non-blocking part. Let us make the hypothesis that the tokens represent processes and places represent local states of processes. In that context, transition t_1 models an usual rendez-vous : “If one process is in its local state p_1 and another is its local state p_3, then the two processes can synchronize and move synchronously to their local states p_2 and p_4, respectively”. In the same context the transition t_2 models a “partially non-blocking” rendez-vous : “If there is one process in p_4 and one in p_2, then the two processes can synchronize and move to p_1 and p_3 respectively. If no process are present in p_4, a process in p_2 does not have to wait and can move to its local state p_1". In that context, the process in p_2 proposes a rendez-vous to processes in p_4. If at least one process is present in p_4 the rendez-vous takes place, otherwise the process in p_2 does not have to wait and can proceed.

Problems

The *marking reachability problem* for a $\mathsf{PN{+}NBA}$ $\mathcal{N}$, is the problem defined as follows: "Given a $\mathsf{PN{+}NBA}$ $\mathcal{N}$ with an initial marking m and a marking m', does m' belong to $\mathsf{Reach}(\mathcal{N}, m)$?". The *marking coverability problem* for a $\mathsf{PN{+}NBA}$ $\mathcal{N}$ is the problem defined as follows: "Given a $\mathsf{PN{+}NBA}$ $\mathcal{N}$ with an initial marking m and a marking m', does there exist a marking m'' that belongs to $\mathsf{Reach}(\mathcal{N}, m)$ and such that $m' \preccurlyeq m''$?". The *boundedness problem* for a $\mathsf{PN{+}NBA}$ $\mathcal{N}$ is the problem defined as follows : "Given a $\mathsf{PN{+}NBA}$ $\mathcal{N}$ and an initial marking m, is $\mathsf{Reach}(\mathcal{N}, m)$ finite ?". The *place boundedness problem* for a $\mathsf{PN{+}NBA}$ $\mathcal{N}$ is the problem defined as follows : "Given a $\mathsf{PN{+}NBA}$ $\mathcal{N}$, an initial marking m and a place p, is there $c \in \mathbb{N}$ such that $\forall m' \in \mathsf{Reach}(\mathcal{N}, m)$ we have $m'(p) \leq c$?" The *action-based* LTL *model checking problem* for a labeled $\mathsf{PN{+}NBA}$ $\mathcal{N}$ is the problem defined as follows : "Given a labeled $\mathsf{PN{+}NBA}$ $\mathcal{N}$, an initial marking m and an action-based LTL formula ϕ, does $\mathcal{L}(\mathcal{N}, m) \models \phi$ hold ?"

It is well-known that those five problems are decidable on Petri nets [KM69,May84,Esp94].

Theorem 3.2 *The marking reachability, marking coverability, boundedness, place boundedness and action-based* LTL *model checking problems are decidable on Petri nets.*

In the next sections we will investigate the decidability of those problems for $\mathsf{PN{+}NBA}$.

4 Decidability results

We give here two positive algorithmic results for the analysis of $\mathsf{PN{+}NBA}$. They are a direct consequence of the strict monotonicity property of that class of extended Petri nets.

Proposition 4.1 *The class* $\mathsf{PN{+}NBA}$ *is strictly monotonic.*

From Proposition 4.1 and [ACJT96,FS01], we deduce the decidability of the coverability problem and the boundedness problem for $\mathsf{PN{+}NBA}$.

Corollary 4.2 *The coverability problem and the boundedness problem are decidable for the class* $\mathsf{PN{+}NBA}$.

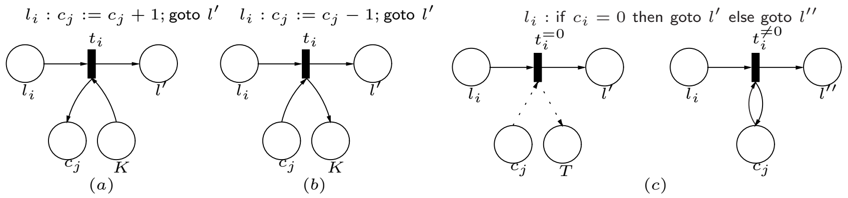

Fig. 3. Simulation of the operations of a 2CM by PN+NBA transitions.

5 Undecidability results

In the previous section, we have seen that the coverability problem and the boundedness problem are decidable for PN+NBA. In this section we show that all the other problems that are decidable for Petri nets become undecidable for PN+NBA.

To establish those undecidability results, we will show that PN+NBA are able to partially simulate the computations of a 2CM. This partial simulation result will allow us to reduce in an uniform way undecidable problems for 2CM to problems for PN+NBA.

5.1 Partial simulations of a 2CM *by a* PN+NBA

Widget.

For any 2CM $C = \langle c_1, c_2, L = \{l_1, l_2, \ldots, l_u\}, \mathsf{Instr}\rangle$, we construct a Petri net with non-blocking arcs $\mathcal{N}_C = \langle \mathcal{P}, \mathcal{T}\rangle$, called the *simulation widget*, defined as follows. The set of places $\mathcal{P}$ is equal to $\{c_1, c_2, l_1, l_2, \ldots, l_u, K, T\}$. The places c_1 and c_2 will be used to keep track of the values of the two counters of C, $l_1, l_2, \cdots, l_u$ called the *control places* will be used to keep track of the program counter of C, K is called the *capacity place*, T is called the *trash*. The use of K and T will be described below. The set of transitions $\mathcal{T}$ is the smallest set of transitions such that for each $l_i \in L$:

- if $\mathsf{Instr}(l_i)$ is of the form $c_j := c_j + 1; \texttt{goto}\ l'$, then $\mathcal{T}$ contains the transition $tr_i = \langle I, O, A\rangle$ with $I = \{(l_i, 1), (K, 1)\}$, $O = \{(c_j, 1), (l', 1)\}$, and $A = \emptyset$.
- if $\mathsf{Instr}(l_i)$ is of the form $c_j := c_j - 1; \texttt{goto}\ l'$, then $\mathcal{T}$ contains the transition $tr_i = \langle I, O, A\rangle$ with $I = \{(l_i, 1), (c_j, 1)\}$, $O = \{l', K\}$, and $A = \emptyset$;
- if $\mathsf{Instr}(l_i)$ is of the form $\texttt{if}\ c_j = 0\ \texttt{then goto}\ l'\ \texttt{else goto}\ l''$ then $\mathcal{T}$ contains two transitions $tr_i^{=0}$ and $tr_i^{\neq 0}$ defined as:
  · $tr_i^{=0} = \langle I, O, A\rangle$ with $I = \{(l_i, 1)\}$, $O = \{(l', 1)\}$, and $A = \{\langle c_j, T\rangle\}$.
  · $tr_i^{\neq 0} = \langle I, O, A\rangle$ with $I = \{(l_i, 1), (c_j, 1)\}$, $O = \{(c_j, 1), (l'', 1)\}$, and $A = \emptyset$.

Figure 3(a) shows the transition that simulates an increment of c_j by moving one token from the capacity place to c_j. Figure 3(b) shows the transition that simulates a decrement of c_j by moving one token from c_j to the capacity place. Figure 3(c) shows the transitions that simulates a zero-test on c_j when c_j is equal to zero (transition $t_i^{=0}$) and when c_j is greater than zero.

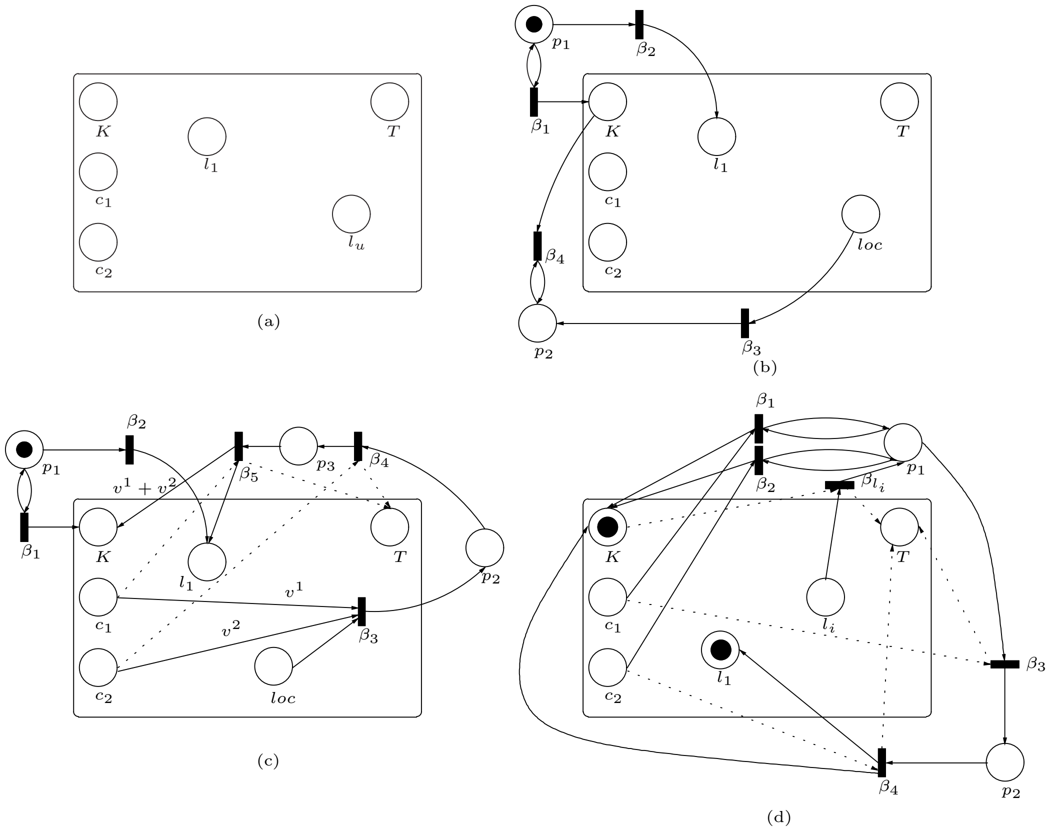

Fig. 4. Construction using the widget.

We note m_k the marking of the places in $\mathcal{P} = \{c_1, c_2, l_1, l_2, \ldots, l_u, K, T\}$ defined as follows: $m_k(l_1) = 1$, for any $l \in \{l_2, l_3, \ldots, l_u\}$, $m_k(l) = 0$, $m_k(c_1) = 0$, $m_k(c_2) = 0$, $m_k(K) = k$, and $m_k(T) = 0$.

Properties of the widget.

Let $C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$ be a 2CM and $\mathcal{N}_C = \langle \mathcal{P}, \mathcal{T}\rangle$ be the simulation widget associated to C as defined above. Let $\gamma = \langle loc_1, v_1^1, v_1^2\rangle\langle loc_2, v_2^2, v_2^2\rangle \ldots$ be the computation of C. We associate to γ a sequence of transitions $tr_1 tr_2 \ldots$ of $\mathcal{N}_C$, such that for all $i \in \mathbb{N}$ such that $1 \leq i \leq |\gamma|$, we have $tr_i = \alpha(\langle loc_i, v_i^1, v_i^2\rangle)$ where α is defined as:

$$\alpha(\langle loc, v_1, v_2\rangle) =$$
$$\begin{cases} tr_k & \text{if } loc = l_k \text{ and } \mathsf{TypeInst}(loc) \neq \mathtt{zerotest_j} \\ tr_k^{=0} & \text{if } loc = l_k \text{ and } \mathsf{TypeInst}(loc) = \mathtt{zerotest_j} \text{ and } v_j = 0. \\ tr_k^{\neq 0} & \text{if } loc = l_k \text{ and } \mathsf{TypeInst}(loc) = \mathtt{zerotest_j} \text{ and } v_j > 0. \end{cases}$$

The sequence of transitions corresponding to γ is denoted by $\alpha(\gamma)$. The function α^{-1} on the transitions of the simulation widget is defined as:

$$\alpha^{-1}(tr_i) = l_i \text{ if } \alpha(\langle l_i, v_1, v_2\rangle) = tr_i \text{ for some } v_1, v_2 \in \mathbb{N}.$$

α^{-1} applied on a sequence of transitions $\sigma = tr_1 \dots tr_n$ of the widget that is firable from m_k ($k \geq 1$), returns a sequence of configurations of C $\gamma = \langle loc_0, v_0^1, v_0^2\rangle \langle loc_1, v_1^1, v_1^2\rangle \dots \langle loc_n, v_n^1, v_n^2\rangle$ such that (i) $loc_0 = l_1$, $v_0^1 = 0$, $v_0^2 = 0$ and (ii) for all $1 \leq i \leq n$, either $\mathsf{TypeInstr}(l_{i-1}) \neq zerotest_j$ and $\langle loc_i, v_i^1, v_i^2\rangle$ is constructed from $\langle loc_{i-1}, v_{i-1}^1, v_{i-1}^2\rangle$ applying $\mathsf{Instr}(l_{i-1})$. Or $\mathsf{Inst}(l_{i-1})$ is of the form $\mathsf{if}\ c_j = 0\ \mathsf{then\ goto}\ l'\ \mathsf{else\ goto}\ l''$ and the following cases holds.

- $tr_i = tr^{=0}$, then $loc_i = l'$, $v_i^1 = v_{i-1}^1$ and $v_i^2 = v_{i-1}^2$, or
- $tr_i = tr^{\neq 0}$ and $loc_i = l''$, $v_i^1 = v_{i-1}^1$, $v_i^2 = v_{i-1}^2$.

We now formalize important properties of the widget by the following lemmas. The proofs of those lemmas are easy but tedious and so given in appendix.

Lemma 5.1 *Let $\gamma = \langle loc_1, v_1^1, v_1^2\rangle\langle loc_2, v_2^1, v_2^2\rangle \dots \langle loc_r, v_r^1, v_r^2\rangle$ be a computation of the* 2CM *$C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$ such that for any $i \in \{1, 2, \dots, r\}$, $v_i^1 + v_i^2 \leq k$. Let $\mathcal{N}_C$ be the simulation widget associated to C. The sequence of transitions $\alpha(\gamma)$ is firable from the marking m_k and firing this sequence of transitions leads to a marking m' defined as follows: $m'(l) = 1$, for $l = loc_r$, $m'(l') = 0$ for any $l' \neq loc_r$, $m'(c_1) = v_r^1$, $m'(c_2) = v_r^2$, $m'(K) = k - v_r^1 - v_r^2$, and $m'(T) = 0$.*

Proof. Given in appendix. □

This lemma formalizes the fact that any computation of a 2CM on which the sum of counters does not exceed k can be faithfully simulated by its associated widget from marking m_k with a computation that does not put tokens in T.

Lemma 5.2 *Let $\sigma = tr_1 tr_2 \dots tr_n$ be a sequence of transitions of the simulation widget $\mathcal{N}_C$ associated to the* 2CM *$C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$. If $m_k \rightarrow^{\sigma} m'$ and $m'(T) = 0$, then $\alpha^{-1}(\sigma)$ is a computation of C with $\mathsf{final}(\alpha^{-1}(\sigma)) = \langle loc, v^1, v^2\rangle$ such that $m'(loc) = 1$, $v^1 = m'(c_1)$ and $v^2 = m'(c_2)$.*

Proof. Given in appendix. □

This second lemma says that any computation of the widget from its initial marking that does not put tokens in T is a simulation of a computation of its associated 2CM.

Lemma 5.3 *Let $\mathcal{N}_C$ be the simulation widget associated to the* 2CM *$C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$. For any marking m such that $m \in \mathsf{Reach}(\mathcal{N}_C, m_k)$, we have that $m(\{c_1, c_2, K, T\}) = k$.*

Proof. Given in appendix. □

This last lemma says that in any reachable marking of the widget, the sum of the tokens in the set of places $\{c_1, c_2, K, T\}$ stays constant.

5.2 Undecidability proofs

We are now equipped to establish the undecidability of the marking reachability, action-based LTL model checking and place boundedness problems.

Theorem 5.4 *The marking reachability problem is undecidable for* PN+NBA.

Proof. Let $C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$ be a 2CM and let $s = \langle loc, v^1, v^2\rangle$ be a configuration of C [5]. Let us show that we can reduce the reachability problem of s in C to the reachability problem between two markings in a PN+NBA.

We construct the PN+NBA $\mathcal{N}' = \langle \mathcal{P}', \mathcal{T}'\rangle$ starting from the simulation widget $\mathcal{N}_C = \langle \mathcal{P}, \mathcal{T}\rangle$ associated to C. To the simulation widget, we add the places and transitions as indicated in figure 4(b). That is, $\mathcal{P}' = \mathcal{P} \cup \{p_1, p_2\}$, $\mathcal{T}' = \mathcal{T} \cup \{\beta_1, \beta_2, \beta_3, \beta_4\}$ and the new transitions are defined as follows: $\beta_1 = \langle I, O, A\rangle$ such that $I = \{(p_1, 1)\}$, $O = \{(K, 1), (p_1, 1)\}$, and $A = \emptyset$; $\beta_2 = \langle I, O, A\rangle$ such that $I = \{(p_1, 1)\}$, $O = \{(l_1, 1)\}$, and $A = \emptyset$; $\beta_3 = \langle I, O, A\rangle$ such that $I = \{(loc, 1)\}$, $O = \{(p_2, 1)\}$ and $A = \emptyset$; $\beta_4 = \langle I, O, A\rangle$ such that $I = \{(K, 1), (p_2, 1)\}$, $O = \{(p_2, 1)\}$, and $A = \emptyset$. We consider the initial marking m such that $m(p_1) = 1$ and for all $p \in \mathcal{P}' \setminus \{p_1\}$, $m(p) = 0$. Furthermore, we consider the marking m_s defined from the configuration s as follows: $m_s(p_1) = 0$, $m_s(p_2) = 1$, $m_s(l) = 0$ for any $l \in L$, $m_s(c_1) = v^1$, $m_s(c_2) = v^2$, $m_s(K) = 0$, and $m_s(T) = 0$. Let us now show that (i) s is reachable in C iff (ii) m_s is reachable from m in $\mathcal{N}'$.

$(i) \to (ii)$. If s is reachable in C then there exists a computation $\gamma = \langle loc_1, v_1^1, v_1^2\rangle, \langle loc_2, v_2^1, v_2^2\rangle, \ldots, \langle loc_r, v_r^1, v_r^2\rangle$ with $s = \langle loc_r, v_r^1, v_r^2\rangle$. Let us note k the maximum of $c_1 + c_2$ along γ. Let us show that we can fire the sequence of transitions $\sigma = \beta_1^k \beta_2 \alpha(\gamma)\beta_3 \beta_4^{k-v_r^1-v_r^2}$ and that $m \to^\sigma m_s$. By firing $\beta_1^k\beta_2$, we put k tokens in the capacity place K and one token in control place l_1. The widget, following Lemma 5.1, is now ready to simulate faithfully γ by firing the sequence of transitions $\alpha(\gamma)$ as K contains enough tokens. As the simulation was faithful, the place c_1 contains v_r^1 tokens and the place c_2 contains v_r^2 tokens. We also know that the place T contains no tokens, and so by Lemma 5.3 the place K contains $k - v_r^1 - v_r^2$ tokens. After we can fire β_3, the control token is moved from the control location loc_r of the widget to the place p_2. So firing $\beta_4^{k-v_r^1-v_r^2}$ leads to the marking m_s.

$(ii) \to (i)$. Let us make the hypothesis that m_s is reachable in $\mathcal{N}'$ with a sequence of transitions σ from m. Let us show that σ must be of the form $\beta_1^*\beta_2\sigma_0\beta_3\beta_4^*$, where σ_0 are transitions of the widget. In m, β_1 and β_2 are the only firable transitions. Once β_2 is fired, place l_1 is marked and the transitions σ_0 of the widget has to be fired. To put one token in p_2, transition β_3 has

[5] In the case of reachability, we may simplify a little bit the construction of the widget by suppressing the capacity place K. However, to keep the proofs uniform and in particular to be able to use lemmas 5.1, 5.2 and 5.3 in all our proofs, we have decided to keep the widget in its full version for this proof.

to be fired. After firing β_3, β_4 is the only firable transition. It remains us to prove that $\alpha^{-1}(\sigma_0)$ is a computation of the 2CM C that reaches s. As $m_s(T)$ contains no token, by Lemma 5.2, we know that the simulation was faithful and so $\alpha^{-1}(\sigma)$ leads to s in C. □

Theorem 5.5 *The action-based* LTL *model checking problem is undecidable for labeled* PN+NBA.

Proof. Let $C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$ be a 2CM and let $s = \langle loc, v^1, v^2\rangle$ be a configuration of C. Let us show that we can reduce the reachability problem of s in C to the action-based LTL model checking problem for a PN+NBA.

We construct the PN+NBA $\mathcal{N}' = \langle \mathcal{P}', \mathcal{T}'\rangle$ starting from the simulation widget $\mathcal{N}_C = \langle \mathcal{P}, \mathcal{T}\rangle$ associated to C. To the simulation widget, we add the places and transitions as indicated in figure 4(c). That is, $\mathcal{P}' = \mathcal{P} \cup \{p_1, p_2, p_3\}$, $\mathcal{T}' = \mathcal{T} \cup \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$ and the new transitions are defined as follows: $\beta_1 = \langle I, O, A\rangle$ such that $I = \{(p_1, 1)\}$, $O = \{(K, 1), (p_1, 1)\}$, and $A = \emptyset$; $\beta_2 = \langle I, O, A\rangle$ such that $I = \{(p_1, 1)\}$, $O = \{(l_1, 1)\}$, and $A = \emptyset$; $\beta_3 = \langle I, O, A\rangle$ such that $I = \{(c_1, v_1), (c_2, v_2), (loc, 1)\}$, $O = \{(p_2, 1)\}$, and $A = \emptyset$; $\beta_4 = \langle I, O, A\rangle$ such that $I = \{(p_2, 1)\}$, $O = \{(p_3, 1)\}$, and $A = \{\langle c_2, T\rangle\}$; $\beta_5 = \langle I, O, A\rangle$ such that $I = \{(p_3, 1)\}$, $O = \{(l_1, 1), (K, v_1 + v_2)\}$, and $A = \{\langle c_1, T\rangle\}$. The labeling function $\mathcal{L}$ is the identity function, that is for any $tr \in \mathcal{T}'$ we have $\mathcal{L}(tr) = tr$. We consider the initial marking m such that $m(p_1) = 1$ and for all $p \in \mathcal{P}' \setminus \{p_1\}$, $m(p) = 0$. Furthermore, we consider the marking m_s defined from the configuration s as follows: $m_s(p_1) = 0$, $m_s(p_2) = 0$, $m_s(p_3) = 0$, $m_s(loc) = 1$, $m_s(l) = 0$ for any $l \neq loc \in L$, $m_s(c_1) = v^1$, $m_s(c_2) = v^2$, $m_s(K) = 0$, and $m_s(T) = 0$. Let us now show that (i) s is reachable in C iff (ii) $\mathcal{L}(\mathcal{N}', m) \not\models \neg\Box\Diamond\beta_3$.

$(i) \rightarrow (ii)$. If s is reachable in C then there exists a computation $\gamma = \langle loc_1, v_1^1, v_1^2\rangle, \langle loc_2, v_2^1, v_2^2\rangle, \ldots, \langle loc_r, v_r^1, v_r^2\rangle$ with $s = \langle loc_r, v_r^1, v_r^2\rangle$. Let us note k the maximum of $c_1 + c_2$ along γ. We now construct from γ a computation σ of $\mathcal{N}'$ such that $\sigma \models \Box\Diamond\beta_3$. We extend the markings m_k $(k \geq 1)$ to $\mathcal{P}'$ such that $m_k(\{p_1, p_2, p_3\}) = 0$. The sequence of transitions $\alpha(\gamma)$ is such that $m \rightarrow^{\beta_1^k\beta_2} m_k \rightarrow^{\alpha(\gamma)} m_s \rightarrow^{\beta_3\beta_4\beta_5} m_k$. By firing $\beta_1^k\beta_2$, we put k tokens in the capacity place K and one token in the control place l_1 to reach the marking m_k. The widget, following Lemma 5.1, is now ready to simulate faithfully γ leading to m_s by firing the sequence of transitions $\alpha(\gamma)$ as K contains enough tokens. After firing β_3, the control token is moved from the control location loc of the widget to the place p_2, v^1 tokens are removed from c_1 and v_2 tokens are removed from c_2. Firing $\beta_4\beta_5$ moves the control token from p_2 to l_1 passing through p_3 and puts $v_1 + v_2$ into K leading to m_k. We conclude that the infinite sequence of transitions $\sigma = \beta_1^k\beta_2(\alpha(\gamma)\beta_3\beta_4\beta_5)^\omega$ is firable from m and satisfies the formula $\Box\Diamond\beta_3$ and so $\mathcal{L}(\mathcal{N}, m) \not\models \neg\Box\Diamond\beta_3$.

$(ii) \rightarrow (i)$. Let us make the hypothesis that there is a sequence of labels associated to a computation of $\mathcal{N}'$ from the marking m and satisfying the formula $\Box\Diamond\beta_3$. Let us show that the infinite sequence of transitions σ corresponding to such a computation must be of the form $\beta_1^*\beta_2\sigma_0\beta_3\beta_4\beta_5\dots\sigma_n\beta_3\beta_4\beta_5\dots$, where each $\sigma_i (i \geq 0)$ is a sequence of transitions of the widget. In fact, β_1 and β_2 are the only firable transitions from m. Once β_2 is fired, place l_1 is marked and a sequence of transitions of the widget σ_0 must be fired. After firing β_3, β_4 followed by β_5 are the only firable transitions, then a sequence of transitions of the widget σ_1 must be fired, etc.

Suppose that s is not reachable and let us derive a contradiction. Assume that we have $m_1 \rightarrow^{\sigma_1} \dots \rightarrow^{\beta_3\beta_4\beta_5} m_{2i-1} \rightarrow^{\sigma_i} m_{2i} \rightarrow^{\beta_3\beta_4\beta_5} m_{2i+1} \rightarrow^{\sigma_{i+1}} \dots$ For each $i \geq 1$, two cases are possible:

1. $m_{2i-1}(c_1) = m_{2i-1}(c_2) = 0$. We consider here two subcases.
   - $(1a)$ $m_{2i}(c_1) = v^1$ and $m_{2i}(c_2) = v^2$. As we suppose that s is not reachable, we have that $\alpha^{-1}(\sigma_i)$ does not correspond to a computation of C and by lemma 5.2, we know that at least one token has been added to the place T. By lemma 5.3, one token has been lost from the set of places $\{c_1, c_2, K\}$. So we can conclude that $m_{2i+1}(\{c_1, c_2, K\}) < m_{2i-1}(\{c_1, c_2, K\})$.
   - $(1b)$ $m_{2i}(c_1) > v^1$ and $m_{2i}(c_2) \geq v^2$, or $m_{2i}(c_1) \geq v^1$ and $m_{2i}(c_2) > v^2$. In that case, after firing the sequence $\beta_3\beta_4\beta_5$, at least one token was added to T from the places c_1 or c_2 and so by lemma 5.3, $m_{2i+1}(\{c_1, c_2, K\}) < m_{2i-1}(\{c_1, c_2, K\})$.

   So in the two subcases, we conclude that we have $m_{2i+1}(\{c_1, c_2, K\}) < m_{2i-1}(\{c_1, c_2, K\})$.
2. $m_{2i-1}(c_1) \neq 0$ or $m_{2i-1}(c_2) \neq 0$. In that case, we start from a marking m_{2i-1} that does not correspond to an initial configuration of the 2CM. We know that it is not possible to add tokens in the set of places $\{c_1, c_2, K\}$ from m_{2i-1} to m_{2i+1}, in fact, we can only move some tokens from $\{c_1, c_2, K\}$ to T. After firing σ_i, two cases are possible.
   - $(2a)$ $m_{2i}(c_1) = v^1$ and $m_{2i}(c_2) = v^2$. In that case, firing $\beta_3\beta_4\beta_5$, we reach a marking m_{2i+1} to which we can apply case 1 above.
   - $(2b)$ $m_{2i}(c_1) > v^1$ and $m_{2i}(c_2) \geq v^2$, or $m_{2i}(c_1) \geq v^1$ and $m_{2i}(c_2) > v^2$. In that case, after firing the sequence $\beta_3\beta_4\beta_5$, at least one token was added to T from the places c_1 or c_2 and so by lemma 3, $m_{2i+1}(\{c_1, c_2, K\}) < m_{2i-1}(\{c_1, c_2, K\})$.

From cases 1 and 2 above, we have that if s is not reachable in C, at least one token is lost (at least one token is put in T) when firing $\sigma_i\beta_3\beta_4\beta_5\sigma_{i+1}\beta_3\beta_4\beta_5$ for any $i \geq 1$. This guarantees, following Lemma 5.3, that the number of tokens in $\{c_1, c_2, K\}$ will reach zero after a finite amount of time. This means that $\mathcal{N}_C$ will not be able to simulate any increment in C and will be blocked. We conclude that σ cannot be infinite and, then, cannot satisfy the formula $\Box\Diamond\beta_3$. This contradicts our hypothesis. $\Box$

Theorem 5.6 *The place boundedness problem is undecidable for* PN+NBA.

Proof. Here, we only sketch the proof. Let $C = \langle c_1, c_2, L, \mathsf{Instr}\rangle$ be a 2CM.

Let us show that we can reduce the boundedness problem for C to the place boundedness problem for a PN+NBA.

From the widget $\mathcal{N}_C$ corresponding to C we construct a PN+NBA $\mathcal{N}'$ as follows. We add the places p_1 and p_2 and the transitions β_1, β_2, β_3 and β_4 as shown in Figure 4(d). Intuitively, while p_1 contains a token the transitions β_1 and β_2 can be fired and move tokens from c_1 and c_2 to the capacity place K. So β_1 and β_2 can be used to reset c_1 and c_2 and put back the tokens in K. When $\beta_3\beta_4$ are fired the control flow token moves from p_1 to l_1 passing through p_2 and one token is added into K. So we extend the simulation capacity of the widget by one. This construction allows us to move all the tokens in $\{c_1, c_2\}$ to K and put the control token into the initial control flow place. If the counters are not set to zero, non-blocking arcs guarantee the lost of at least one token from $\{c_1, c_2, K\}$. Moreover, for each place l_i such that $\mathsf{TypeInst}(l_i) = \mathtt{inc_j}$ we add a transition β_{l_i} that moves the control token into p_1 and moves one token from K to T if there is some tokens in K. We extend m_k $(k \geq 1)$ to $\mathcal{P}'$ such that $m_k(\{p_1, p_2\}) = 0$ and we take m_1 as initial marking. We have that K is unbounded iff C is unbounded.

Suppose that C is unbounded. Starting from m_1, the only way to increment the number of tokens in $\{c_1, c_2, K\}$ is to mimic C until there is no more tokens in K and the next operation to mimics is an increment. Then, firing the transitions β_1 and β_2, the counters are set to zero moving all the tokens from $\{c_1, c_2\}$ to K and one new token is generated into K by firing $\beta_3\beta_4$. This allows us to reach m_2. Applying this strategy from any m_i $(i \geq 2)$ allows us to reach m_{i+1} and leads to the construction of an infinite computation where the number of tokens in $\{c_1, c_2, K\}$ grows infinitely often. As all the tokens in the set $\{c_1, c_2, K\}$ are moved to K at the end of the simulation of C by firing β_1 and β_2, K is unbounded in this computation.

If C is bounded, there is $k \in \mathbb{N}$ such that starting from m_k, it is not possible to faithfully simulates C and then fire β_1 without losing tokens in $\{c_1, c_2, K\}$ by moving tokens to T with non-blocking arcs. This ensures the boundedness of K. $\square$

6 Future Works

Recently, several extensions of the Petri net formalism have been proposed for modeling parametric systems, a.o. Transfer nets [Cia94], Reset nets [Bil91], Multi-transfer nets [DRVB02], and the extension proposed in this paper. We have defined the extension of this paper in order to model partially non-blocking rendez-vous. The other extensions have been proposed for similar reasons related to modeling issues. Nevertheless, a careful analysis of the expressive power of those different extensions of Petri net has not been done so far. We plan to compare formally the expressive power of those extensions by studying the languages that they are able to define.

7 Conclusion

In this paper, we have studied the decidability of five problems for a simple extension of Petri Nets that makes possible the modeling of "partially non-blocking rendez-vous" (necessary to model multi-threaded JAVA programs). The five problems that we have studied are decidable for the basic Petri Net model. We have shown that due to strict monotonicity of the extended model and thanks to general results on well-structured transition systems, the marking coverability and the boundedness problems remain decidable. On the other hand, the three other problems: marking reachability, action-based LTL model-checking and place boundedness become undecidable. Our results are summarized in Table 1.

Problems	PN	$\mathsf{PN}+\mathsf{NBA}$
Marking Reachability	$\surd$	$\times$
Marking Covering	$\surd$	$\surd$
Boundedness	$\surd$	$\surd$
Place Boundedness	$\surd$	$\times$
Action-based LTL	$\surd$	$\times$

Table 1
Summary of the decidability/undecidability results. $\surd$ stands for "decidable", and $\times$ for "undecidable".

The reader interested in our results may want to look at the following related works. The decidability of the five problems considered in this paper for the Petri net models can be found in: for boundedness, place boundedness and covering in [KM69], for reachability in [May84], and action-based LTL model-checking in [Esp94]. Several definition of extended Petri nets can be found in [Cia94] and in [DFS98]. Undecidability results for the class of transfer nets can be found in [DFS98,DJS99,Duf98]. In [May00], similar problems are studied in the context of lossy counter machines. For the practical analysis of models that subsume the class of extended Petri Nets studied here, we refer the reader to [DRVB02].

Acknowledgement

We would like to thank anonymous reviewers for suggesting improvements to the submitted version of this paper and in particular for suggesting the future work on comparing the expressive power of the different extensions of Petri nets proposed in the context of parametric systems verification.

References

[ACJT96] 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.

[AK86] K.R. Apt and D. Kozen. Limits for Automatic program Verification of Finite-State Concurrent Systems. *Information Processing Letters*, 22(6), 1986.

[Bil91] J. Billington. *Extensions to Coloured Petri nets and their applications to Protocols*. PhD thesis, University of Cambridge, 1991.

[Cia94] G. Ciardo. Petri nets with marking-dependent arc multiplicity: properties and analysis. In *Proceeding of the 15th International Conference on Applications and Theory of Petri Nets (ICATPN 94)*, volume 815 of *LNCS*, pages 179–198. Springer, 1994.

[DFS98] 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.

[DJS99] C. Dufourd, P. Jancar, and P. Schnoebelen. Boundedness of Reset P/T nets. In *Proceedings of the 26th International Colloquium on Automata, Languages, and Programming (ICALP'99)*, volume 1644 of *LNCS*, pages 301–310. Springer, 1999.

[DRVB02] G. Delzanno, J-F. Raskin, and L. Van Begin. Towards the Automated Verification of Multithreaded 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.

[Duf98] C. Dufourd. *Réseaux de Petri avec reset/transfert : Décidabilité et indécidabilité*. PhD thesis, ENS de Cachan, 1998.

[EFM99] 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.

[Esp94] J. Esparza. On the Decidabilty of Model Checking for Several mu-calculi and Petri Nets. In *Proceedings of the 19th International Colloquium on Trees in Algebra and Programming*, volume 787 of *LNCS*, pages 115–129, 1994.

[FS01] A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, 2001.

[GS92] S. M. German and A. P. Sistla. Reasoning about Systems with Many Processes. *Journal of ACM*, 39(3):675–735, 1992.

[KM69] R. M. Karp and R. E. Miller. Parallel Program Schemata. *Journal of Computer and System Sciences*, 3:147–195, 1969.

[May84] E.W. Mayr. An algorithm for the general petri net reachability problem. *SIAM Journal of Computing*, 3(13):441–460, 1984.

[May00] R. Mayr. Undecidable Problems in Unreliable Computations. In *Proceedings of the 4th Latin American Symposium on Theoretical Informatics (LATIN'2000)*, volume 1776 of *LNCS*, pages 377–386. Springer, 2000.

[Min67] N.M. Minsky. *Finite and Infinite Machines*. Englewood Cliffs, N.J., Prentice-Hall, 1967.

Appendix

Lemma 5.1 *Let $\gamma = \langle loc_1, v_1^1, v_1^2\rangle\langle loc_2, v_2^2, v_2^2\rangle \dots \langle loc_r, v_r^1, v_r^2\rangle$ be a computation of the* 2CM *C such that for any $i \in \{1, 2, \dots, n\}$, $v_i^1 + v_i^2 \leq k$. Let $\mathcal{N}_C$ be the* PN+NBA *associated to C. The sequence of transitions $\alpha(\gamma)$ is firable from the marking m_k and firing this sequence of transitions leads to a marking m' defined as follows: $m'(l) = 1$, for $l = loc_r$, $m'(l') = 0$ for any $l' \neq loc_r$, $m'(c_1) = v_r^1$, $m'(c_2) = v_r^2$, $m'(K) = k - v_r^1 - v_r^2$, and $m'(T) = 0$.*

Proof. By induction on the length of the computations of C. The basic case ($l = 1$) is obvious. Suppose that the lemma holds for all the computations of size $l < n$.

Let $\gamma = \gamma' \cdot \langle loc_n, v_n^1, v_n^2\rangle$ be a computation of C of size n where $\gamma' = \langle loc_1, v_1^1, v_1^2\rangle \dots \langle loc_{n-1}, v_{n-1}^1, v_{n-1}^2\rangle$. By induction hypothesis, we have that $\alpha(\gamma')$ leads to the marking m' in $\mathcal{N}_C$ such that $m'(l_i) = 1$ if $l_i = loc_{n-1}$, $m'(l_i) = 0$ for all $l_i \in L \setminus \{loc_{n-1}\}$, $m'(c_1) = v_{n-1}^1$, $m'(c_2) = v_{n-1}^2$, $m'(K) = k - v_{n-1}^1 - v_{n-1}^2$ and $m'(T) = 0$. The following cases hold.

1. If $\mathsf{Instr}(loc_{n-1})$ is of the form $c_j := c_j + 1; \mathsf{goto}\ l'$, then we have that $\alpha(\langle loc_{n_1}, v_{n-1}^1, v_{n-1}^2\rangle) = tr$ such that $tr = \langle I, O, \emptyset\rangle$ where $I = \{(loc_{n-1}, 1), (K, 1)\}$ and $O = \{(l', 1), (c_j, 1)\}$. By hypothesis we have $m'(K) > 0$ and we have $m' \to^{tr} m''$ such that $m''(l') = 1$, $m''(l_i) = 0$ for all $l_i \in L \setminus \{l'\}$, $m''(c_1) = v_n^1$, $m''(c_2) = v_n^2$, $m''(K) = k - v_n^1 - v_n^2$ and $m''(T) = 0$.
2. If $\mathsf{Instr}(loc_{n-1})$ is of the form $c_j := c_j - 1; \mathsf{goto}\ l'$, then we have that $\alpha(\langle loc_{n-1}, v_{n-1}^1, v_{n-1}^2\rangle) = tr$ such that $tr = \langle I, O, \emptyset\rangle$ where $I = \{(loc_{n-1}, 1), (c_j, 1)\}$ and $O = \{(l', 1), (K, 1)\}$. As $\langle loc_{n-1}, v_{n-1}^1, v_{n-1}^2\rangle$ has a successor, we have $v_{n-1}^j > 0$ and $m' \to^{tr} m''$ such that $m''(l') = 1$, $m''(l_i) = 0$ for all $l_i \in L \setminus \{l'\}$, $m''(c_1) = v_n^1$, $m''(c_2) = v_n^2$, $m''(K) = k - v_n^1 - v_n^2$ and $m''(T) = 0$.
3. If $\mathsf{Instr}(loc_{n-1})$ is of the form $\mathsf{if}\ c_j = 0\ \mathsf{then\ goto}\ l'\ \mathsf{else\ goto}\ l''$, then if $v_{n-1}^j = 0$ we have $\alpha(\langle loc_{n-1}, v_{n-1}^1, v_{n-1}^2\rangle) = tr^{=0}$ such that $tr^{=0} = \langle I^{=0}, O^{=0}, \{\langle c_j, T\rangle\}\rangle$ where $I^{=0} = \{(loc_{n-1}, 1)\}$ and $O^{=0} = \{(l', 1)\}$. $tr^{=0}$ is firable from m' and we have $m' \to^{tr} m''$ such that $m''(l') = 1$, $m''(l_i) = 0$ for all $l_i \in L \setminus \{l'\}$, $m''(c_1) = v_n^1$, $m''(c_2) = v_n^2$, $m''(K) = k - v_n^1 - v_n^2$ and $m''(T) = 0$. Otherwise if $v_{n-1}^j > 0$ we have $\alpha(\langle loc_{n-1}, v_{n-1}^1, v_{n-1}^2\rangle) = tr^{\neq 0}$ such that $tr^{\neq 0} = \langle I^{\neq 0}, O^{\neq 0}, \{\langle c_j, T\rangle\}\rangle$ where $I^{\neq 0} = \{(loc_{n-1}, 1), (c_j, 1)\}$ and $O^{\neq 0} = \{(l'', 1), (c_j, 1)\}$. $tr^{\neq 0}$ is firable from m' and we have $m' \to^{tr} m''$ such that $m''(l'') = 1$, $m''(l_i) = 0$ for all $l_i \in L \setminus \{l''\}$, $m''(c_1) = v_n^1$, $m''(c_2) = v_n^2$, $m''(K) = k - v_n^1 - v_n^2$ and $m''(T) = 0$.

□

Lemma 5.2 *Let $\sigma = tr_1 tr_2 \dots tr_n$ be a sequence of transitions of the* PN+NBA *$\mathcal{N}_C$ associated to the* 2CM *C. If $m_k \to^{\sigma} m'$ and $m'(T) = 0$, then $\alpha^{-1}(\sigma)$ is a computation of C such that $\mathsf{final}(\alpha^{-1}(\sigma)) = \langle loc, v^1, v^2\rangle)$ where $loc = l$ if $m'(l) = 1$, $v^1 = m'(c_1)$ and $v^2 = l'(c_2)$.*

Proof. By induction on the size of the sequence of transitions. The basic case ($l = 1$) is obvious. Suppose that the lemma holds for all the sequences of transitions of size $l < n$. Let $\sigma = \sigma' \cdot tr_n$ be a sequence of transitions of $\mathcal{N}_C$ of size n where $\sigma' = tr_1 \dots tr_{n-1}$. By induction hypothesis we have that $m \rightarrow^{\sigma'} m'$ and $\mathsf{final}(\alpha^{-1}(\sigma')) = \langle loc, v_1^1, v_1^2 \rangle$ such that $m'(loc) = 1$, $m'(c_1) = v^1$, $m'(c_2) = v^2$ and $m'(T) = 0$. The following cases holds.

1. if $\mathsf{Instr}(\alpha^{-1}(tr_n))$ is of the form $c_j := c_j + 1; \mathsf{goto}\ l'$, then $tr_n = \langle I, O, \emptyset \rangle$ such that $I = \{loc, K\}$ and $O = \{l', c_j\}$. We have $m' \rightarrow^{tr_n} m''$ and $\alpha^{-1}(\sigma)$ is a computation of C with $\mathsf{final}(\alpha^{-1}(\sigma)) = \langle l', v_2^1, v_2^2 \rangle$ such that $m''(l') = 1$, $m''(c_1) = v_2^1$, $m''(c_2) = v_2^2$ and $m''(T) = 0$.
2. if $\mathsf{Instr}(\alpha^{-1}(tr_n))$ is of the form $c_j := c_j - 1; \mathsf{goto}\ l'$, then $tr_n = \langle I, O, \emptyset \rangle$ such that $I = \{(loc, 1), (c_j, 1)\}$ and $O = \{(l', 1), (K, 1)\}$. We have $m' \rightarrow^{tr_n} m''$ and $\alpha^{-1}(\sigma)$ is a computation of C with $\mathsf{final}(\alpha^{-1}(\sigma)) = \langle l', v_2^1, v_2^2 \rangle$ such that $m''(l') = 1$, $m''(c_1) = v_2^1$, $m''(c_2) = v_2^2$ and $m''(T) = 0$.
3. if $\mathsf{Instr}(\alpha^{-1}(tr_n))$ is of the form $\mathsf{if}\ c_j = 0\ \mathsf{then\ goto}\ l'\ \mathsf{else\ goto}\ l''$, then if $m'(c_j) = 0$, tr_n must be such that $tr_n = \langle I, O, \{\langle c_j, T \rangle\} \rangle$ with $I = \{(loc, 1)\}$ and $O = \{(l', 1)\}$. We have $m' \rightarrow^{tr_n} m''$ and $\alpha^{-1}(\sigma)$ is a computation of C with $\mathsf{final}(\alpha^{-1}(\sigma)) = \langle l', v_2^1, v_2^2 \rangle$ such that $m''(l') = 1$, $m''(c_1) = v_2^1$, $m''(c_2) = v_2^2$ and $m''(T) = 0$. Otherwise if $m'(c_j) > 0$, tr must be such that $tr = \langle I, O, \emptyset \rangle$ with $I = \{(loc, 1), (c_j, 1)\}$ and $O = \{(l'', 1), (c_j, 1)\}$, otherwise T would contain one token after firing tr_n. We have $m' \rightarrow^{tr_n} m''$ and $\alpha^{-1}(\sigma)$ is a computation of C with $\mathsf{final}(\alpha^{-1}(\sigma)) = \langle l'', v_2^1, v_2^2 \rangle$ such that $m''(l'') = 1$, $m''(c_1) = v_2^1$, $m''(c_2) = v_2^2$ and $m''(T) = 0$.

□

Lemma 5.3 *Let $\mathcal{N}_C$ be the* PN+NBA *associated to the* 2CM *C. For any marking $m \in \mathsf{Reach}(\mathcal{N}_C, m_k)$, we have that $m(c_1, c_2, K, T) = k$.*

Proof. By induction on the size of the minimal computation of $\mathcal{N}_C$ that allows us to reach m. The basic case ($l = 1$) is obvious. Suppose that the lemma holds for all the markings reachable in i steps from m_k in $\mathcal{N}_C$ with $i < n$. Suppose that m is reachable by firing $n-1$ transitions and we have $m \rightarrow^{tr} m'$ for some transition tr of $\mathcal{N}$. tr can be of the following forms:

1. $tr = \langle I, O, \emptyset \rangle$ with $I = \{(l, 1), (c_j, 1)\}$ and $O = \{(l', 1), (K, 1)\}$ and corresponds to a decrement. In this case we have $m(\{c_1, c_2, K, T\}) = m'(\{c_1, c_2, K, T\})$.
2. $tr = \langle I, O, \emptyset \rangle$ with $I = \{(l, 1), (K, 1)\}$ and $O = \{(l', 1), (c_j, 1)\}$ and corresponds to an increment. In this case we have $m(\{c_1, c_2, K, T\}) = m'(\{c_1, c_2, K, T\})$.
3. $tr = \langle I, O, \{\langle c_j, T \rangle\} \rangle$ with $I = \{(l, 1)\}$ and $O = \{(l'', 1)\}$ and corresponds to a test for zero on c_j. In this case, when $m(c_j) = 0$ or $m(c_j) > 0$, we have $m(\{c_1, c_2, K, T\}) = m'(\{c_1, c_2, K, T\})$.

4. $tr = \langle I, O, \emptyset \rangle$ with $I = \{(l, 1), (c_j, 1)\}$ and $O = \{(l'', 1), (c_j, 1)\}$ and corresponds to a test for zero on c_j when $m(c_j) > 0$. In this case we have $m(\{c_1, c_2, K, T\}) = m'(\{c_1, c_2, K, T\})$.

$\square$