A Model Checking Decision Procedure for Sequential Recursive Petri Nets

Serge Haddad[1] and Denis Poitrenaud[2]

[1] LAMSADE - UPRESA 7024, Université Paris IX, Dauphine
Place du Maréchal De Lattre de Tassigny, 75775 Paris cedex 16
[2] LIP6 - UMR 7606, Université Paris VI, Jussieu
4, Place Jussieu, 75252 Paris cedex 05

Abstract. Recursive Petri nets (RPNs) have been introduced to model systems with dynamic structure. Whereas this model is a strict extension of Petri nets and context-free grammars (w.r.t. the language criterion), reachability in RPNs remains decidable. However the kind of model checking which is decidable for Petri nets becomes undecidable for RPNs. In this work, we introduce a submodel of RPNs called sequential recursive Petri nets (SRPNs) and we study its theoretical features. First we show that we can decide whether a RPN is a sequential one. Then, we analyze the language aspects proving that the SRPN languages still strictly include the union of Petri nets and context-free languages. Moreover the family of languages of SRPNs is closed under intersection with regular languages (unlike the one of RPNs). This property is the starting point for the model checking of the action-based linear time logic which is also shown to be decidable. To the best of our knowledge, this is the first time such a result is obtained for a model strictly including Petri nets and context-free grammars.

1 Introduction

In the area of verification theory, a great attention has been recently paid on infinite state systems. In contrast to finite state systems where theoretical and practical developments mainly focus on complexity reduction [Hol90], an essential topic in infinite state systems is to find a trade-off between expressivity of the models and decidability of verification [HM96]. As the model checking of temporal logic formula is one of the most general approach for verification, it has been intensively studied in the framework of infinite-state systems.

Context-free grammars (also called context-free processes) have led to complementary works. In [Wal96] , it is shown that the model checking of branching time μ-calculus formula is decidable and that it is DEXPTIME-complete. When restricting the temporal logic formula to the linear time logic LTL, one obtains polynomial time algorithms [BEM97,FWW97].

In [Esp97], model checking for Petri nets has been studied. The branching temporal logic as well as the state-based linear temporal logic are undecidable even for restricted logics. Fortunately, the model checking for action-based linear temporal logic is decidable. The case of infinite sequences may be reduced to the search of repetitive sequences studied in [Yen92] (an EXPSPACE-complete problem) and the case of finite sequences may be reduced to the reachability problem [May81]. Recently, in [Bou98] the reachability problem for Petri nets is also shown to be EXPSPACE-complete. Thus the model checking complexity is also EXPSPACE-complete.

It seems interesting to combine context-free grammars and Petri nets and to look for decidable properties. Indeed, for two such models - the process rewrite systems [May97] and the recursive Petri nets (RPNs) [HP99b] - the reachability problem is decidable (and, due to [Bou98], EXPSPACE-complete). However, for both these two models, the model checking of action-based temporal logic becomes undecidable. It remains undecidable even for restricted models such as those presented in [BH96]. So (to the best of our knowledge) for any existing model strictly including Petri nets and context-free grammars, the action-based linear time model checking is undecidable.

In this work, we present a submodel of RPNs called sequential recursive Petri nets (SRPNs) and we give some decision procedures including the model checking. Roughly speaking, in recursive Petri nets some transitions emulate concurrent procedure calls by initiating a new token game in the net. The return mechanism is ensured by reachability conditions. A state of a RPN is then a tree of "token games".

A recursive Petri net is sequential if there are firable transitions only in the last initiated token game. Such a definition is behavioral and our first result is that we can decide whether a RPN is a SRPN. We then study the language family of SRPNs and we show that this family strictly includes the union of Petri nets and context-free languages. Moreover, unlike RPNs, this family is closed under intersection with regular languages.

In the last part of the paper, building on this result, we focus on the model checking for an action-based linear time logic. The case of finite (maximal) sequences is handled by a straightforward adaptation of the closure result. The case of infinite sequence is more tricky and requires to distinguish w.r.t. the asymptotic behavior of the depth of token games in an infinite sequence. Based on this analysis, we obtain an EXPSPACE upper bound for the decision procedure.

Due to the space restrictions, only sketches of proof are given in the paper. However in appendix, we give complete proofs for the main propositions. This appendix will be omitted in the final version.

2 Sequential Recursive Petri Nets

2.1 Recursive Petri nets

A RPN has the same structure as an ordinary one except that the transitions are partitioned into two categories: *elementary transitions* and *abstract transitions*. Moreover a *starting marking* is associated to each abstract transition and a effectively semilinear set of *final markings* is defined. The semantics of such a net may be informally explained as follows. In an ordinary net, a thread plays the token game by firing a transition and updating the current marking (its internal state). In a RPN there is a dynamical tree of threads (denoting the fatherhood relation) where each thread plays its own token game. The step of a RPN is thus a step of one of its threads. If the thread fires an elementary transition, then it updates its current marking using the ordinary firing rule. If the thread fires an abstract transition, it consumes the input tokens of the transition and generates a new child which begins its token game with the starting marking of the transition. If the thread reaches a final marking, it may terminate aborting its whole descent of threads and producing (in the token game of its father) the output tokens of the abstract transition which gave birth to him. In case of the root thread, one obtains an empty tree.

Definition 1 (Recursive Petri nets). A *recursive Petri net* is defined by a tuple $N = \langle P, T, W^-, W^+, \Omega, \Upsilon \rangle$ where

- P is a finite set of places, T is a finite set of transitions.
- A transition of T can be either elementary or abstract. The sets of elementary and abstract are respectively denoted by T_{el} and T_{ab} (with $T = T_{el} \uplus T_{ab}$ where $\uplus$ denotes the disjoint union).
- W^- and W^+ are the pre and post flow functions defined from $P \times T$ to $\mathbb{N}$.
- Ω is a labeling function which associates to each abstract transition an ordinary marking (i.e. an element of $\mathbb{N}^P$) called the starting marking of t.
- Υ is an effectively semilinear set of final markings (any usual syntax can be accepted for its specification).

Definition 2 (Extended marking). An *extended marking tr* of a recursive Petri net $N = \langle P, T, W^-, W^+, \Omega, \Upsilon \rangle$ is a labeled tree $tr = \langle V, M, E, A \rangle$ where

- V is the set of vertices,
- M is a mapping $V \to \mathbb{N}^P$,
- $E \subseteq V \times V$ is the set of edges and
- A is a mapping $E \to T_{ab}$.

A *marked recursive Petri net* $\langle N, tr_0 \rangle$ is a recursive Petri net N associated to an initial extended marking tr_0.

We denote by $v_0(tr)$ the root node of the extended marking tr. The empty tree is denoted by $\perp$. Any ordinary marking m can be seen as an extended marking, denoted by $\lceil m \rceil$, consisting of a single node. For a vertex v of an extended marking, we denote by *pred*(v) its (unique) predecessor in the tree (defined only if v is different from the root) and by $Succ(v)$ the set of its direct and indirect successors including v ($\forall v \in V, Succ(v) = \{v' \in V \mid (v, v') \in E^*\}$ where E^* denotes the reflexive and transitive closure of E). An *elementary step* of a RPN may be either a firing of a transition or a closing of a subtree (called a *cut step* and denoted by τ).

Definition 3. A transition t is *enabled* in a vertex v of an extended marking tr (denoted by $tr \xrightarrow{t,v}$) if $\forall p \in P, M(v)(p) \geq W^-(p, t)$ and a cut step is *enabled* in v (denoted by $tr \xrightarrow{\tau,v}$) if $M(v) \in \Upsilon$

Definition 4. The *firing* of an enabled elementary step t from a vertex v of an extended marking $tr = \langle V, M, E, A \rangle$ leads to the extended marking $tr' = \langle V', M', E', A' \rangle$ (denoted by $tr \xrightarrow{t,v} tr'$) depending on the type of t.

- $t \in T_{el}$
  - $V' = V$, $E' = E$, $\forall e \in E, A'(e) = A(e)$, $\forall v' \in V \setminus \{v\}$, $M'(v') = M(v')$
  - $\forall p \in P, M'(v)(p) = M(v)(p) - W^-(p,t) + W^+(p,t)$
- $t \in T_{ab}$
  - $V' = V \cup \{v'\}$, $E' = E \cup \{(v,v')\}$, $\forall e \in E, A'(e) = A(e)$, $A'((v,v')) = t$
  - $\forall v'' \in V \setminus \{v\}, M'(v'') = M(v'')$, $\forall p \in P, M'(v)(p) = M(v)(p) - W^-(p,t)$
  - $M'(v') = \Omega(t)$

  where v' is a fresh identifier absent in V
- $t = \tau$
  - $V' = V \setminus Succ(v)$, $E' = E \cap (V' \times V')$, $\forall e \in E', A'(e) = A(e)$
  - $\forall v' \in V' \setminus \{pred(v)\}, M'(v') = M(v')$
  - $\forall p \in P, M'(pred(v))(p) = M(pred(v))(p) + W^+(p, A(pred(v), v))$

  Let us notice that if v is the root of the tree then the firing of τ leads to to empty tree $\perp$.

The depth of an extended marking is recursively defined as 0 for $\perp$, 1 for a unique vertex and, for the general case, the maximum depth of the direct subtrees of the root incremented by one. For an extended marking tr, its depth is denoted by $depth(tr)$. A firing sequence is defined as usual: a sequence $\sigma = tr_0(t_0, v_0)tr_1(t_1, v_1)\ldots(t_{n-1}, v_{n-1})tr_n$ is a firing sequence (denoted by $tr_0 \xrightarrow{\sigma} tr_n$) iff $tr_i \xrightarrow{t_i,v_i} tr_{i+1}$ for $i \in [0, n-1]$. We define the depth of σ as the maximal depth of tr_1, tr_2, $\ldots$, tr_n. In the sequel, for sake of simplicity, σ will be often denoted by $\sigma = t_0 t_1 \ldots t_{n-1}$

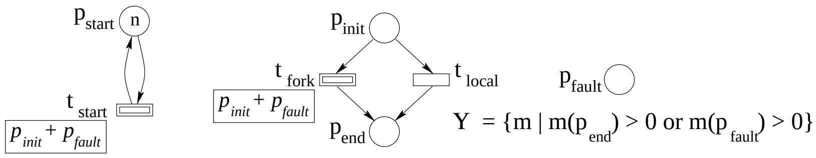

Fig. 1. a simple recursive Petri net

The figure 1 shows the modeling of n similar transactions (represented by n tokens in p_{start}). We represent an abstract transition by a double border rectangle and its initial marking is indicated in a frame. A transaction is started by the firing of the transition t_{start}. When initialized, the transaction may proceed locally by firing t_{local} or starts a new process by firing t_{fork}. Each process may achieve by reaching p_{end} or abort since p_{fault} is always marked. In the latter case, the nested processes are also stopped due to the cut mechanism.

A firing sequence of this RPN is presented in the figure 2 for $n = 2$. The arcs of the trees composing the visited extended markings are labeled by the abstract transition t_{start} for the thin ones and by t_{fork} for the bold ones. The thread in

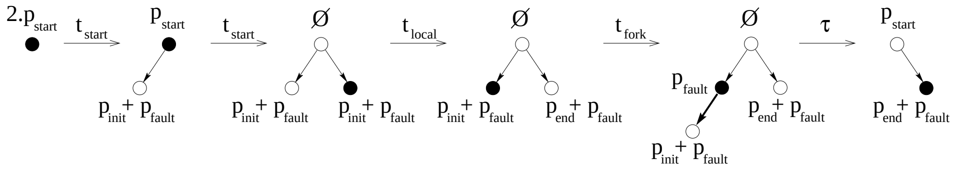

Fig. 2. a firing sequence

which the following step is fired is represented in black. One can notice that each firing of abstract transition leads to the creation of a new node in the tree whereas the firing of the last cut step prunes a subtree not reduced to one node.

2.2 Sequential Recursive Petri Nets

In a recursive Petri net, there are two kinds of parallelism between activities: concurrent firings inside the same node and concurrent firings in different nodes. In order to model "sequential call" with abstract transitions, the second kind of parallelism must be forbidden. This is the aim of the next definition.

Definition 5 (Sequential Recursive Petri Nets). Let $\langle N, tr_0 \rangle$ be a marked recursive Petri net. $\langle N, tr_0 \rangle$ is a *sequential recursive Petri net* if the following conditions hold:

- tr_0 is a tree composed by only one node,
- Each reachable extended marking of N from tr_0 satisfies
  - each node has at most one successor,
  - there is no enabled step in a node different to the leaf.

The first condition is imposed for sake of simplicity but is not a theoretical restriction. As an example, the net of Fig. 1 is a SRPN iff n is equal to one. We could have chosen an alternative syntactical definition (with an additional control place) but the present one leads to the next statement.

Proposition 6 (SRPN class belonging). *Let $\langle N, tr_0 \rangle$ be a marked RPN, one can decide whether $\langle N, tr_0 \rangle$ is a SRPN.*

Sketch of Proof. A RPN is not a SRPN iff there is a node within a reachable extended marking where one can fire simultaneously an abstract transition and any other step (a property defined by an effectively semilinear set of markings). We proceed in two stages. We compute all the starting markings of a node in a reachable extended marking (there are only a finite number). Then, for any such marking, we look in this node whether we can reach the above semilinear set. The effectiveness of these two steps is deduced from the decision procedure for the reachability problem of RPN (see the appendix for more details).

3 Language Properties

We denote by $\mathcal{L}(N, tr_0, Tr_f)$ (where Tr_f is a finite extended marking set) the set of firing sequences (mapped on $(T \cup \tau)^*$) of N from tr_0 to an extended marking of Tr_f. This set is called the language of N. More generally, the languages we will consider are defined via a labeling function. A *labeled marked recursive Petri net* is a marked recursive Petri net and a labeling function h defined from the transition set $T \cup \{\tau\}$ to an alphabet Σ plus λ (the empty word). h is extended to sequences and then to languages. The language of a labeled marked recursive Petri net $\langle\langle N, tr_0\rangle, \Sigma, h\rangle$ for a finite extended marking set Tr_f is defined by $h(\mathcal{L}(N, tr_0, Tr_f))$.

We now study the properties of the languages generated by labeled SRPNs. These languages are defined for a given finite set of terminal extended markings. For sake of simplicity, we impose that such sets are composed by extended marking limited to a single node. One can remark that this condition is not a theoretical restriction. The first result concerning the languages generated by SRPNs is about their relation with Petri net and context-free languages.

Theorem 7 (Strict inclusion). *SRPN languages strictly include the union of context-free and Petri net languages*

We prove that SRPN languages are closed under intersection with regular languages. For a SRPN and an automaton (see appendix for definition and notation), both labeled on a same alphabet, we define a product SRPN resulting of their composition and demonstrate that its language is the intersection of their respective languages.

The product SRPN is constructed from the places of the original one by adding a place set Q which corresponds to the states of the automaton. As usual, the elementary transitions are synchronized with the ones of the automaton using these new places. For each extended arc $q \stackrel{a}{\Longrightarrow} q'$ (with $a \in \Sigma \cup \{\lambda\}$) of the automaton and for each elementary transition t such that $h(t) = a$, an elementary transition $t.q.q'$, having $W^-(t) + q$ as pre-condition and $W^+(t) + q'$ as post-condition, is added. When an abstract transition is fired a new node appears and, due to the SRPN definition, the token game is limited to this node. Then, we have to predict the state reached by the automaton when the opened branch will be closed. The abstract transitions constructed in the product SRPN are denoted $t.q.q'.q''$ where the prefix $t.q.q'$ expresses the same conditions as for the elementary transitions (excepted that t is an abstract transition of the original net). For each state $q'' \in Q$ such an abstract transition is added (the prediction is non deterministic). To ensure that the predicted state is effectively reached when the cut step closing the branch is fired, a set of places $\overline{Q}$ (complementary to Q) is used. The firing of an abstract transition $t.q.q'.q''$ leads to the creation of a new node for which its starting marking has the place $\overline{q''}$ marked. Using these places, the effectively semilinear set of final markings is built in order to ensure that the predicted state is effectively reached. Let us notice that this composition corresponds to a weak synchronization as some transitions of the SRPN can be labeled by λ.

Definition 8 (Product SRPN). Let $A = \langle \Sigma, Q, \Delta, q_0 \rangle$ be an automaton and $S = \langle\langle N, \lceil m_0 \rceil\rangle, \Sigma, h\rangle$ a labeled SRPN. The product RPN of A and S is a labeled marked RPN $\langle\langle N', \lceil m'_0 \rceil\rangle, \Sigma, h'\rangle$ defined by

- $P' = P \cup Q \cup \overline{Q}$, $m'_0 = m_0 + q_0$
-
- $T'_{el} = \{t.q.q' \mid (t \in T_{el}) \wedge (q, q' \in Q) \wedge (q \overset{h(t)}{\Longrightarrow} q')\}$
- $\forall t.q.q' \in T'_{el}$,
  - $h'(t.q.q') = h(t)$, ${W'}^{-}(t.q.q') = W^{-}(t) + q$, ${W'}^{+}(t.q.q') = W^{+}(t) + q'$
- $T'_{ab} = \{t.q.q'.q'' \mid (t \in T_{ab}) \wedge (q, q', q'' \in Q) \wedge (q \overset{h(t)}{\Longrightarrow} q')\}$
- $\forall t.q.q'.q'' \in T'_{ab}$,
  - $h'(t.q.q'.q'') = h(t)$
  - ${W'}^{-}(t.q.q'.q'') = W^{-}(t) + q$, ${W'}^{+}(t.q.q'.q'') = W^{+}(t) + q''$
  - $\Omega'(t.q.q'.q'') = \Omega(t) + q' + \overline{q''}$
- $\Upsilon' = \{m + q + \overline{q'} \mid (m \in \Upsilon) \wedge (q, q' \in Q) \wedge (q \overset{h(\tau)}{\Longrightarrow} q')\}$
- $h'(\tau) = h(\tau)$

It is clear that the constructed RPN is a SRPN as the initial marking is a tree limited to a single node and the pre and post conditions of the initial SRPN are preserved and enriched by the automaton flow. In Fig. 3 of appendix, the behavior of this product is illustrated and commented. The next theorem shows the soundness of this building.

Theorem 9 (SRPN product property). *Let $A = \langle \Sigma, Q, \delta, q_0 \rangle$ be an automaton, $F \subseteq Q$ a set of final states, $S = \langle\langle N, \lceil m_0 \rceil\rangle, \Sigma, h\rangle$ a labeled SRPN and M_f a set of terminal markings. Let $\langle\langle N', \lceil m'_0 \rceil\rangle, \Sigma, h'\rangle$ be the product SRPN of A and S and $M'_f = \{\lceil m + q \rceil \mid \lceil m \rceil \in M_f \wedge q \in F\}$. The following equality holds*

$$h'(\mathcal{L}(N', \lceil m'_0 \rceil, M'_f)) = h(\mathcal{L}(N, \lceil m_0 \rceil, M_f)) \cap \mathcal{L}(A, F)$$

Corollary 10 (SRPN closure). *The family of SRPN languages is closed under intersection with regular languages.*

The SRPN closure property gives the starting point for the decidability of the model checking problem. Moreover, in [HP99a], it is demonstrated that the RPN languages are not closed under intersection with regular ones leading to the next corollary.

Corollary 11 (SRPN versus RPN). *The family of SRPN languages is strictly included in the family of RPN languages.*

4 Model Checking

The model checking that we investigate is the action based linear-time μ-calculus applied to SRPNs. The usual verification method consists to check the existence of a sequence of the system fulfilling the negation of the formula. Depending on the kind of the sequence, different semantics have been defined. We will study the main ones: finite sequences, maximal finite sequences (leading to a deadlock), infinite sequences, divergent sequences (infinite sequences ended by a non observable subsequence). As a linear-time μ-calculus formula is equivalently represented by a Büchi automaton, we limit ourselves to this representation.

4.1 Finite and maximal finite sequences

When the searched sequences are finite, Büchi automata are nothing else than ordinary automata. A slight adaptation of the product of a SRPN and an automaton makes possible the reduction of the model-checking problem to a reachability problem for the product SRPN. In case of maximal finite sequences, adaptation is still possible although more intricate (see the appendix for details).

Theorem 12 (Acceptance of finite sequences). *Let $A = \langle \Sigma, Q, \delta, q_0 \rangle$ be an automaton, $F \subseteq Q$ a set of final states and $S = \langle \langle N, \lceil m_0 \rceil \rangle, \Sigma, h \rangle$ a labeled SRPN. The existence of a finite firing sequence σ of S such that $h(\sigma) \in \mathcal{L}(A, F)$ is decidable.*

Theorem 13 (Acceptance of maximal finite sequences). *Let $A = \langle \Sigma, Q, \delta, q_0 \rangle$ be an automaton, $F \subseteq Q$ a set of final states and $S = \langle \langle N, \lceil m_0 \rceil \rangle, \Sigma, h \rangle$ a labeled SRPN. The existence of a finite firing sequence σ of S such that σ leads to a deadlock of N and $h(\sigma) \in \mathcal{L}(A, F)$ is decidable.*

4.2 Infinite and divergent sequences

We are looking for an infinite firing sequence of the SRPN accepted by a Büchi automaton. We will perform two independent searches depending on a characteristic of the sequence: the asymptotic behavior of the depth of the sequence. Let $\sigma = \lceil m_0 \rceil \xrightarrow{t_1, v_1} tr_1 \xrightarrow{t_2, v_2} \ldots tr_{i-1} \xrightarrow{t_i, v_i} tr_i \ldots$ be an infinite sequence, we define $dinf(\sigma) = \liminf_{i \to \infty} depth(tr_i)$ (defined by $\lim_{i \to \infty} inf_{j \geq i} \{depth(tr_j)\}$). $dinf(\sigma)$ always exists but it can be either finite or infinite.

In case of a finite value, there exists a strictly increasing sequence of indexes $i_1, \ldots, i_k, \ldots$ such that:

- beyond i_1 the set of indexes $\{i_1, i_2, \ldots, i_k, \ldots\}$ is exactly the indexes for which the depth of the visited extended markings is equal to $dinf(\sigma)$ $(\forall i \geq i_1, depth(tr_i) = dinf(\sigma) \Leftrightarrow i \in \{i_1, i_2, \ldots, i_k, \ldots\})$
- beyond i_1 the depth of the visited extended markings will be greater or equal than $dinf(\sigma)$ $(\forall i \geq i_1, depth(tr_i) \geq dinf(\sigma))$
- i_1 is the first index from which the depth of the visited extended markings will be no more less than $dinf(\sigma)$ $(\forall i < i_1, \exists j \geq i, depth(tr_j) < dinf(\sigma))$

So σ will be decomposed as $\lceil m_0 \rceil \xrightarrow{\sigma_0} tr_{i_1} \xrightarrow{\sigma_1} \ldots tr_{i_k} \xrightarrow{\sigma_k} tr_{i_{k+1}} \ldots$ where σ_0 ends with the firing of an abstract transition leading to an extended marking of depth $dinf(\sigma)$ (with the creation of a new node) and σ_k is either a firing of an elementary transition in this node or a sequence beginning by the firing of an abstract transition in this node and ended by a corresponding cut step.

In case of an infinite value, there exists a strictly increasing sequence of indexes $i_1, \ldots, i_k, \ldots$ such that:

- k is the depth of the extended marking tr_{i_k} $(\forall k, depth(tr_{i_k}) = k)$
- beyond i_k the depth of the visited extended markings will be greater or equal than k $(\forall i \geq i_k, depth(tr_i) \geq k)$

– i_k is the first index from which the depth of visited extended markings will be no more less than k ($\forall i < i_k, \exists j \geq i, depth(tr_j) < k$)

So σ will be decomposed as $\lceil m_0 \rceil = tr_{i_1} \xrightarrow{\sigma_1} tr_{i_2} \xrightarrow{\sigma_2} \ldots tr_{i_k} \xrightarrow{\sigma_k} tr_{i_{k+1}} \ldots$ where σ_k begins by a firing in an extended marking of depth k, ends with the firing of an abstract transition leading to an extended marking of depth $k+1$ and such that all the extended markings visited by σ_k have a depth greater or equal than k.

In order to build infinite sequences from the decompositions shown above, we must be able to check the existence of some finite firing subsequences beginning and ending in the same node of the two extended markings and corresponding to paths of the Büchi automaton. Moreover, we want to distinguish two cases depending on the visit of an accepting state of the automaton. The checking of the existence of such finite sequences may be done similarly as the model-checking of finite sequences.

We are now in position to explain the two main procedures. Looking for a sequence σ with $dinf(\sigma)$ finite, we first compute the couples of starting markings and automaton states reachable by a firing sequence. We build an ordinary Petri net representing an abstract view of sequences of the SRPN (recognized by the automaton) where the successive extended markings visited by the sequence are infinitely often reduced to a single node. Then, for each couple as initial marking of this Petri net, we look for an infinite sequence visiting a subset of transitions infinitely often (this can be done by the algorithm of [Yen92]).

Looking for a sequence σ with $dinf(\sigma)$ infinite, we build a graph where the nodes are the computed couples of the first procedure and an edge denotes that one node has been reached from the other one by a sequence increasing by one the depth of the visited extended markings and such that the intermediate subsequences never decrease the depth below its initial value. The edges are partitioned depending on the visit by the sequence of an accepting state of the Büchi automaton. Then the existence of an accepting infinite sequence is equivalent to the existence of some kind of strongly connected component.

Although we will not prove it in the paper, the complexity of these procedures is EXSPACE thus, due to the lower bound for Petri nets, the model-checking problem is EXSPACE-complete. The case of divergent sequences is handled similarly (see the appendix for proof of Th. 14).

Theorem 14 (Acceptance of infinite sequences). *Let $A = \langle \Sigma, Q, \delta, q_0 \rangle$ be an automaton, $F \subset Q$ a set of accepting states and $S = \langle \langle N, \lceil m_0 \rceil \rangle, \Sigma, h \rangle$ a labeled SRPN. The existence of an infinite sequence σ of $\langle N, \lceil m_0 \rceil \rangle$ such that $h(\sigma)$ is an infinite word recognized by a path $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots$ of A satisfying $|\{i \mid q_i \in F\}| = \infty$ is decidable.*

Theorem 15 (Acceptance of divergent sequences). *Let $A = \langle \Sigma, Q, \delta, q_0 \rangle$ be an automaton, $F \subset Q$ a set of accepting states and $S = \langle \langle N, \lceil m_0 \rceil \rangle, \Sigma, h \rangle$ a labeled SRPN. The existence of an infinite sequence σ of $\langle N, \lceil m_0 \rceil \rangle$ such that $h(\sigma)$ is a finite word recognized by a path $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n$ of A with $q_n \in F$ is decidable.*

5 Conclusion

In this work, we have introduced sequential recursive Petri nets and studied their theoretical features. At first we have shown how to decide whether a RPN is a SRPN. Then, we have studied the language family of SRPNs and proved that this family strictly includes the union of Petri nets and context-free languages. Moreover, unlike RPNs, this family is closed under intersection with regular languages. In the last part of the paper, we have focused on the model checking for an action-based linear time logic and obtained an EXPSPACE upper bound for the decision procedure.

An important characteristic of SRPNs is their capability to generate infinite in-degree transition systems. Such a feature makes possible to model dynamic systems which can be handled neither by process algebra nor by Petri nets. So, we plan to study with SRPNs fault tolerant systems and similar ones which require this capability.

References

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model-checking. In *Proc. of CONCUR'97*, 1997.

[BH96] A. Bouajjani and P. Habermehl. Constraint properties, semi-linear systems, and Petri nets. In *Proc. of CONCUR'96*, volume 1119 of *Lecture Notes in Computer Science*. Springer Verlag, 1996.

[Bou98] Z. Bouziane. A primitive recursive algorithm for the general Petri net. In *Proc. 39th IEEE Symp. Foundations of Computer Science*, 1998.

[Esp97] J. Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, 34:85–107, 1997.

[FWW97] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems. In *Proc. of INFINITY'97*, 1997.

[HM96] Y. Hirshfeld and F. Moller. Decidability results in automata and process theory. In *Logics for Concurrency: Structure versus Automata*, volume 1043 of *Lecture Notes in Computer Science Tutorial*, pages 102–148. Springer Verlag, 1996.

[Hol90] G. J. Holzmann. *Design and Validation of Computer Protocols.* Prentice Hall, November 1990.

[HP99a] S. Haddad and D. Poitrenaud. Decidability and undecidability results for recursive Petri nets. Technical Report 019, LIP6, Paris VI University, Paris, France, September 1999.

[HP99b] S. Haddad and D. Poitrenaud. Theoretical aspects of recursive Petri nets. In *Proc. 20th Int. Conf. on Applications and Theory of Petri nets*, volume 1639 of *Lecture Notes in Computer Science*, pages 228–247, Williamsburg, VA, USA, June 1999. Springer Verlag.

[May81] E.W. Mayr. An algorithm for the general Petri net reachability problem. In *Proc. 13th Annual Symposium on Theory of Computing*, pages 238–246, 1981.

[May97] R. Mayr. *Decidability and Complexity of Model Checking Problems for Infinite-State Systems.* PhD thesis, TU-München, 1997.

[Wal96] I. Walukiewicz. Pushdown processes: Games and model checking. In *Int. Conf. on Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 62–74. Springer Verlag, 1996.

[Yen92] H-C. Yen. A unified approach for deciding the existence of certain Petri net paths. *Information and Computation*, 96:119–137, 1992.

A Appendix

Automaton of Sect. 3

An automaton is a tuple $A = \langle \Sigma, Q, \Delta, q_0 \rangle$ where Σ is an alphabet, Q a finite set of states, $\Delta \subseteq Q \times \Sigma \times Q$ a transition relation and $q_0 \in Q$ an initial state. As usual, we denote by $q \xrightarrow{a} q'$ that $(q, a, q') \in \Delta$. Moreover, the extension of $\longrightarrow$ to sequences over Σ is denoted by $\Longrightarrow$ and is defined as follows:

- $\forall q \in Q, q \overset{\lambda}{\Longrightarrow} q$
- $\forall q, q' \in Q, q \overset{\omega a}{\Longrightarrow} q' \Leftrightarrow \exists q'', q \overset{\omega}{\Longrightarrow} q'' \wedge q'' \xrightarrow{a} q'$

For an automaton $A = \langle \Sigma, Q, \Delta, q_0 \rangle$ and a state set $F \subseteq Q$, we denote by $\mathcal{L}(A, F)$ the set of sequences $\{\omega \in \Sigma^* \mid \exists q \in F, q_0 \overset{\omega}{\Longrightarrow} q\}$.

Proof of Prop. 6 (SRPN class belonging)

Proof. Alg. A.1 decides if a given marked RPN belongs to the SRPN class. In this algorithm, the ordinary net N_{elem} is constructed from the RPN in the following way: each abstract transition is removed and for each closable abstract transition, an elementary transition (having the same pre and post sets) is added. An abstract transition t is said closable if $\perp$ is reachable from the extended marking composed by a single node corresponding to the starting marking $\Omega(t)$. An algorithm for the computation of the closable abstract transitions can be found in [HP99b]. In the algorithm A.1, a set $\uparrow Pre(t)$ denotes the effectively semilinear set of ordinary markings in which the transition t is enabled ($\uparrow Pre(t) = \{m \mid \forall p \in P, m(p) \geq W^-(p,t)\}$).

Now, we prove the correctness of the algorithm A.1. Let (N, tr_0) be a RPN which is not a SRPN. Then either tr_0 is not an extended marking composed with a single node or there exists a firing sequence $\sigma = t_1.t_2 \ldots t_n$ leading to an extended marking $tr_n = \langle V, M, E, A \rangle$ such that $(\forall v \in V, |Succ(v)| \leq 1) \wedge (\exists v \in V, |Succ(v)| = 1 \wedge (\exists t \in T, \forall p \in P, M(v) \geq W^-(p,t)) \vee (M(v) \in \Upsilon))$.

The first test realized by the algorithm detects that the initial extended marking has most than one node. Then, we have to demonstrate that the second case is well detected by the remainder of the algorithm.

Let σ be a minimal sequence satisfying these conditions. Let t_i be the abstract transition for which its firing has led to the creation of the successor node of v. Because σ is minimal, t_i is the last transition fired in σ at the level of v. Moreover, because t_i is an abstract transition, this firing only consumes tokens in $M(v)$. We can deduce that either t and t_i are concurrent or $(M(v) + W^-(t_j)) \in \Upsilon$. This condition is detected by the algorithm if the set *Examine* contains the ordinary marking from which the thread of v has began. This ordinary marking can be either the initial marking of $v_0(tr_0)$ or the starting marking associated to the abstract transition for which its firing has led to the creation of the node v. It is clear that, by construction, this marking belongs to *Examine*. $\square$

Algorithm A.1 SRPN class belonging

boolean **SRPN**(RPN N, extended marking tr)
begin
  if $Succ(v_0(tr)) \neq \emptyset$ **then**
    return *false*;
  fi;
  $Enable = \emptyset$;
  $Examine = \emptyset$;
  $ToExamine = M(v_0(tr))$;
  while $ToExamine \neq \emptyset$ **do**
    $m = Pick(ToExamine)$;
    $Examine = Examine \cup \{m\}$;
    forall $t \in T_{ab} \setminus Enable$ **do**
      if $Reachable(N_{elem}, m, \uparrow Pre(t))$ **then**
        $Enable = Enable \cup \{t\}$;
        if $\Omega(t) \notin Examine$ **then**
          $ToExamine = ToExamine \cup \{\Omega(t)\}$;
        fi;
      fi;
    od;
  od;
  forall $m \in Examine$ **do**
    if $Reachable(N_{elem}, m, \bigcup_{t \in T_{ab}}(\uparrow Pre(t)) + (\bigcup_{t' \in T}(\uparrow Pre(t')) \cup \Upsilon))$ **then**
      return *false*;
    fi;
  od;
  return *true*;
end

Proof of Th. 7 (Strict inclusion)

Proof. It is obvious that any PN is a SRPN. Moreover, in [HP99b], it is demonstrated that any context-free language can be simulated by a RPN. We can remark that the proposed construction of the RPN corresponding to a context-free language leads to a SRPN. In the same paper, it is shown that RPN languages strictly include the union of context-free and Petri net languages. The proof of this result exhibits a RPN for which its language is neither PN nor context-free language. We can remark that this RPN is a SRPN. Then, we can conclude that the language family of SRPN strictly includes the union of the context-free and PN languages. □

Illustration of the product SRPN behavior (Def. 8)

The use of the complementary places $\overline{Q}$ is illustrated in Fig. 3. A sequence of a SRPN and a path in an automaton as well as the sequence of the product SRPN corresponding to the synchronization of both are presented. In the product SRPN, we have $t'_1 = t_1.q_0.q_1.\overline{q_3}$ and $t'_2 = t_2.q_1.q_1.\overline{q_2}$. When an abstract transition is fired,

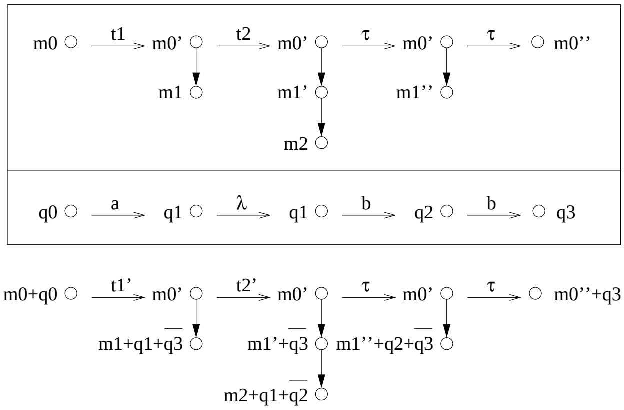

Fig. 3. synchronization of a SRPN with an automaton versus a product SRPN

the automaton state reaches by the cut step closing the opened branch is predicted and coded in an element of $\overline{Q}$. The effectively semilinear set Υ' ensures that the good predicted state is effectively reached by the firing of the cut step.

When the abstract transition $t1$ is fired the automaton moves from $q0$ to $q1$ and it is predicted that the opened branch will be closed by a cut step leading the automaton to the state $q3$ (the place $\overline{q3}$ is marked in the leaf node). This prediction is realized at the end of the given sequence.

From this example, it is clear that the product SRPN can make some bad predictions. Moreover, bad predictions cannot lead to terminal markings which defined the language of the product. However, the existence of a good prediction insures that a word of the intersection of the automaton and the SRPN languages will be produced by a firing sequence of the product.

Proof of Th. 9 (SRPN product property)

Proof. First, we demonstrate that to each word ω of $h'(\mathcal{L}(N', \lceil m'_0 \rceil, M'_f))$ corresponds a sequence σ in $\mathcal{L}(N, \lceil m_0 \rceil, M_f)$ such that $h(\sigma) = \omega$ and $\omega \in \mathcal{L}(A, F)$.

Let σ' be any sequence of N' such that $\lceil m'_0 \rceil \xrightarrow{\sigma'} tr'$ (with $tr' = \langle V', M', E', A' \rangle$) and $h'(\sigma') = \omega$. From the definition 8, it is easy to show that there exists a unique place $q \in Q$ marked in the leaf node of tr'.

We define a mapping z from T' to T depending on there types

- $\forall t.q.q' \in T'_{el}, z(t.q.q') = t$
- $\forall t.q.q'.q'' \in T'_{ab}, z(t.q.q'.q'') = t$
- $z(\tau) = \tau$

Moreover, we define the extended marking $tr = \langle V, M, E, A \rangle$ as follows:

- $V = V'$
- $\forall v \in V, M(v) = M'(v) \setminus (Q \cup \overline{Q})$
- $E = E'$

– $\forall e \in E, A(e) = z(A'(e))$

From the definition 8, it is clear that tr is an extended marking of N. Moreover, it is straightforward that $z(\sigma')$ is a sequence of N from $\lceil m_0 \rceil$ to the extended marking tr. Indeed, m'_0 is a superset of m_0 and the pre and post conditions of the transitions in T' are supersets of the ones in T. Finally, from the definition of the pre and post conditions of the transitions in T', we can deduce a path in A from q_0 to the state q and from the definition of h' and z, we can conclude that this path recognizes the word ω.

We can apply this proof to any word ω of $h'(\mathcal{L}(N', \lceil m'_0 \rceil, M'_f))$ and demonstrate that the extended marking reached by the corresponding sequence in N belongs to M_f.

Now, we demonstrate that to each word ω of $h(\mathcal{L}(N, \lceil m_0 \rceil, M_f)) \cap \mathcal{L}(A, F)$ corresponds a sequence σ' in $\mathcal{L}(N', \lceil m'_0 \rceil, M'_f)$ such that $h'(\sigma') = \omega$.

Let $\omega = a_0.a_1 \ldots a_n$ be a word of $h(\mathcal{L}(N, \lceil m_0 \rceil, M_f)) \cap \mathcal{L}(A, F)$. Then, there exists a sequence $\sigma = \lceil m_0 \rceil \xrightarrow{t_1, v_1} tr_1 \xrightarrow{t_2, v_2} \ldots \xrightarrow{t_m, v_m} \lceil m_f \rceil$ such that $\lceil m_f \rceil \in M_f$ and $h(\sigma) = \omega$. Moreover, there exists a path $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n$ in A such that $q_n \in F$.

We have to demonstrate the existence of a sequence σ' of N' from $\lceil m_0 + q_0 \rceil$ to $\lceil m_f + q_n \rceil$ such that $\lceil m_f + q_n \rceil \in M'_f$ and $h'(\sigma') = \omega$.

First, we define a mapping ind from $[0, m]$ to $[0, n]$.

– $ind(0) = 0$
– $\forall i \in [1, m], h(t_i) \neq \tau \;\Rightarrow ind(i) = ind(i-1) + 1$
– $\forall i \in [1, m], h(t_i) = \tau \;\Rightarrow ind(i) = ind(i-1)$

We can remark that $\forall i \in [1, m], h(t_1 \ldots t_i) = a_1 \ldots a_{ind(i)}$.

Then, we define a mapping z' from $\{t_1, t_2, \ldots, t_m\}$ to T' depending on their types

– $\forall i \in [1..m], t_i \in T_{el} \Rightarrow z'(t_i) = t_i.q_{ind(i-1)}.q_{ind(i)}$
– $\forall i \in [1..m], t_i \in T_{ab} \Rightarrow$ let j be the minimal range such that $j > i$ $\wedge\, depth(tr_{i-1}) = depth(tr_j), z'(t_i) = t_i.q_{ind(i-1)}.q_{ind(i)}.q_{ind(j)}$
– $\forall i \in [1..m], t_i = \tau \Rightarrow z'(t_i) = \tau$

We can notice that for an abstract transition, the range j always exists because the depths of the initial and final markings of the sequence are equal to one and because the firings occur only in the leaf node. More generally, for an extended marking tr_i visited by σ, we denote the range of the cut step which closes the branch opening at the depth d by $return(i, d)$ (i.e. $\forall i \in [1..m], \forall 0 \leq d < depth(tr_i), return(i, d) = Min(\{j > i \mid depth(tr_j) = d\})$).

We only have to demonstrate that $z'(\sigma)$ is a firing sequence of N' from $\lceil m_0 + q_0 \rceil$ to $\lceil m_f + q_n \rceil$. Indeed, from the definition of M'_f, it is clear that $\lceil m_f + q_n \rceil \in M'_f$.

For a given range $i \in [0..m]$, we formulate some hypotheses (Hyp) on tr_i, $\sigma_i = t_1 \ldots t_i$, $\omega_i = a_1 \ldots a_{ind(i)}$ and $q_0, \ldots, q_{ind(i)}$ in relation with tr'_i and $\sigma'_i = z'(t_1) \ldots z'(t_i)$.

$$
(Hyp) \begin{cases}
\text{- } z(\sigma'_i) = \sigma_i \\
\text{- } h'(\sigma'_i) = \omega_i \\
\text{- } depth(tr'_i) = depth(tr_i) \\
\text{- } \forall 1 \le d \le depth(tr'_i), \mathcal{M}_{|P}(tr'_i, d) = \mathcal{M}(tr_i, d) \\
\text{- } \mathcal{M}_{|Q}(tr'_i, depth(tr'_i)) = \{q_{ind(i)}\} \\
\text{- } \forall 1 \le d < depth(tr'_i), \mathcal{M}_{|Q}(tr'_i, d) = \emptyset \\
\text{- } \mathcal{M}_{|\overline{Q}}(tr'_i, 1) = \emptyset \\
\text{- } \forall 1 < d \le depth(tr'_i), \mathcal{M}_{|\overline{Q}}(tr'_i, d) = \{\overline{q}_{ind(return(i,d-1))}\}
\end{cases}
$$

where $\mathcal{M}(tr, d)$ denotes the ordinary marking labeling the node of depth $0 < d \le depth(tr)$ of the extended marking tr (this node is unique because each node of a SRPN extended marking has at most one successor).

From the definitions of h' and z', we can easily deduce that $h'(z'(\sigma)) = \omega$ and then the two first hypotheses are satisfied for any i. For the others, we reason inductively on the size of the prefix of $z'(\sigma)$. If this size is equal to zero, it is clear that (Hyp) holds. Let (Hyp) satisfied for a prefix of length $k-1$, we demonstrate that it is verified for k.

- If $z'(t_k) = t_k.q_{ind(k-1)}.q_{ind(k)} \in T'_{el}$. We know by the hypotheses on the extended marking tr'_{k-1} that the pre condition of t_k is marked in the leaf node as well as the place $q_{ind(k-1)}$. And then the transition t'_k is enabled $(W'^{-}(t_k.q_{ind(k-1)}.q_{ind(k)}) = W^{-}(t_k) + q_{ind(k-1)})$. Moreover, its firing leads to an extended marking satisfying the hypotheses (the place $q_{ind(k)}$ is unmarked and the place $q_{ind(k+1)}$ marked and the firing on the leaf marking projected on P has the same effect of the firing of t_k in σ).
- If $z'(t_k) = t_k.q_{ind(k-1)}.q_{ind(k)}.q_{ind(return(k,depth(tr_{k-1}))} \in T'_{ab}$. Like for elementary transition, we know that the transition t'_k is enabled. Its firing leads to unmark the place $q_{ind(k-1)}$ and to the creation of a new leaf node having $\Omega(t_k) + q_{ind(k)} + \overline{q_{ind(return(k,depth(tr_{k-1}))}}$ as marking. It is clear that this new extended marking satisfies the hypotheses. Moreover, by the definition of z' and the prediction of $ind(return(k, depth(tr_{k-1}))$, we know that the automaton must reach the state $q_{ind(return(k,depth(tr_{k-1}))}$ when the branch will be closed.
- If $z'(t_k) = \tau$. Knowing that the transition t_k is a cut step in σ and by the hypotheses on the extended marking, we know that the marking in the leaf node projected on P belongs to Υ. Moreover, we know that the place $q_{ind(k-1)}$ is marked in this node as well as the place $\overline{q_{ind(return(k-1,depth(tr_{k-1})-1))}}$. But $t_k = \tau$ and then $return(k-1, depth(tr_{k-1}) - 1) = k$. We can deduce that $\overline{q_{ind(k)}}$ is marked in the leaf node. Moreover, if $h(\tau) \neq \lambda$, because ω is a path of the automaton, we have $q_{ind(k-1)} \xrightarrow{a_{ind(k)}} q_{ind(k)}$ and then a cut step is enabled from tr'_{k-1}. If $h(\tau) = \lambda$ then $ind(k-1) = ind(k)$ and a cut step is also enabled . In both cases, from the definition of Υ' and T'_{ab}, we can deduce that the hypotheses are satisfied for the reached extended marking.

□

Proof of Th. 12 (Acceptance of finite sequences)

Proof. Let $\langle\langle N', \lceil m'_0 \rceil\rangle, \Sigma, h'\rangle$ be the product SRPN of A and S. We construct a new SRPN $\langle N'', \lceil m''_0 \rceil\rangle$ in the following way:

- $N'' = N'$ except for $\Upsilon'' = \Upsilon' \cup \{m \mid \exists q \in F, m \geq q\}$
- $m'_0 = m''_0$

Now, we demonstrate that the existence of a finite firing sequence σ of S such that $h(\sigma) \in \mathcal{L}(A, F)$ is equivalent to the reachability of $\perp$ by $\langle N'', \lceil m''_0 \rceil\rangle$.

Let σ be a sequence of N from $\lceil m_0 \rceil$ such that $h(\sigma)$ is recognized by a path of A from q_0 to a state $q \in F$. From σ, we can construct a sequence of N'' from $\lceil m''_0 \rceil$ which reaches an extended marking having the place q marked in its leaf node and such that it has been predicted that all the opened branches are going to be closed in this state q (i.e. excepted for the root, all the nodes have the place $\overline{q}$ marked). From this particular extended marking, the marking of the leaf node belongs to the second part of the set Υ'' and then a cut step can occur. Because this firing marks the place q in the father node, again a cut step can occur and so on until the empty tree $\perp$ is reached.

Now, let σ'' be a sequence of $\langle N'', \lceil m''_0 \rceil\rangle$ such that $\lceil m''_0 \rceil \xrightarrow{\sigma''} tr_n \xrightarrow{\tau, v_0(tr_n)} \perp$. Because $\langle N'', \lceil m''_0 \rceil\rangle$ is a SRPN, tr_n is a tree limited to a single node and, by construction, we have $\forall \overline{q}, M(v_0(tr_n))(\overline{q}) = 0$. Then, only the second condition of Υ'' can be applied for the firing of the last cut step and then $\exists q \in F, M(v_0(tr_n))(q) \geq 1$.

Let σ_f be the minimal prefix of σ'' such that $\lceil m''_0 \rceil \xrightarrow{\sigma_f} tr_f$ with a place $q \in F$ marked in tr_f (necessarily in its leaf). It is clear that by definition of σ_f all the cuts used in σ_f use the first part of the definition of Υ'' and then the sequence σ_f is also a sequence of $\langle N', \lceil m'_0 \rceil\rangle$. From the theorem 9, we can deduce a sequence σ in $\langle N, \lceil m_0 \rceil\rangle$ such that $h(\sigma)$ is recognized by a path of A from q_0 to q. $\square$

Proof of Th. 13 (Acceptance of maximal finite sequences)

Proof. We construct a labeled SRPN $\langle\langle N^b, \lceil m^b_0 \rceil\rangle, \Sigma, h^b\rangle$ similar to the product SRPN.

- $P^b = P \cup Q \cup \overline{Q} \cup \{b, \overline{b}\}$
- $m'_0 = m_0 + q_0$

- $T^b_{el} = \{t.q.q' \mid (t \in T_{el}) \wedge (q, q' \in Q) \wedge (q \xRightarrow{h(t)} q')\}$
- $\forall t.q.q' \in T^b_{el}$,
  - $h^b(t.q.q') = h(t)$
  - $W^{b^-}(t.q.q') = W^-(t) + q$, $W^{b^+}(t.q.q') = W^+(t) + q'$

- $T^b_{ab} = \{t.q.q'.q'' \mid (t \in T_{ab}) \wedge (q, q' \in Q) \wedge (q'' \in (Q \cup \{b\})) \wedge (q \xRightarrow{h(t)} q')\}$
- $\forall t.q.q'.q'' \in T^b_{ab}$,
  - $h^b(t.q.q'.q'') = h(t)$
  - $W^{b^-}(t.q.q'.q'') = W^-(t) + q$,
  - $q'' \in Q \Rightarrow W^{b^+}(t.q.q'.q'') = W^+(t) + q''$

- $q'' = b \Rightarrow W^{b^+}(t.q.q'.q'') = b$
- $\Omega^b(t.q.q'.q'') = \Omega(t) + q' + \overline{q''}$

– $\Upsilon^b = \{m + q + \overline{q'} \mid (m \in \Upsilon) \wedge (q, q' \in Q) \wedge (q \xrightarrow{h(\tau)} q')\} \cup \{m + q + \overline{b} \mid m \in Dead(N) \wedge q \in F\} \cup \{m \mid m \geq b\} \cup \{m + q \mid m \in Dead(N) \wedge q \in F\}$
– $h^b(\tau) = h(\tau)$

where $Dead(N)$ is the effective semilinear set $\{m \in \mathbb{N}^P \mid \forall t \in T \cup \{\tau\}, \neg m \xrightarrow{t}\}$.

We prove that reaching the empty tree in N^b is equivalent to reach a deadlock in the product SRPN with a place $q \in F$ marked in the leaf. As N^b includes the behaviors of the product SRPN, a deadlock sequence can be emulated. However, in order to reach $\perp$ after this sequence, we need to slightly modify this simulation. Places b and $\overline{b}$ are added in order to predict that after the firing of an abstract transition in a node, this node will become again the leaf only when the emulation has led to an adequate deadlock sequence. Place b will be marked if the previous abstract transition closes itself and makes possible to cut this leaf due to the definition of Υ^b (the iteration of this mechanism will necessary lead to $\perp$). Place $\overline{b}$ is marked in the leaf "opened" by the prediction and restricts the closability of this node to two cases: an adequate deadlock is reached in this leaf or the deadlock has been reached before this node becomes again the leaf. The last part of the definition of Υ^b covers the case where one reaches the deadlock in the root. The proof of the correctness of this construction is similar to the one used for Theorem 12. □

Proof of Th. 14 (Acceptance of infinite sequences)

First, we establish the two following lemmas.

Lemma 16 (Recognition). *Let* $A = \langle \Sigma, Q, \delta, q_0 \rangle$ *be an automaton, and* $S = \langle \langle N, \lceil m_0 \rceil \rangle, \Sigma, h \rangle$ *a labeled SRPN. Let* M_f *an effectively semilinear marking set of* N *and* $q_i, q_j \in Q$ *be two automaton states. The existence of a sequence* σ *of* $\langle N, \lceil m_0 \rceil \rangle$ *such that* $\lceil m_0 \rceil \xrightarrow{\sigma} \lceil m_f \rceil$ *where* $m_f \in M_f$ *and* $h(\sigma)$ *is recognized by a path of* A *from* q_i *to* q_j *is decidable.*

Proof. Let $\langle \langle N', \lceil m'_0 \rceil \rangle, \Sigma, h' \rangle$ be the product SRPN of S and $\langle \Sigma, Q, \delta, q_i \rangle$. From this SRPN, we define the SRPN $\langle N^r, \lceil m^r_0 \rceil \rangle$ as follows:

– $P^r = P' \cup \{Init\}, T^r = T'$,
– $W^{r^-} = W'^-, W^{r^+} = W'^+$,
– $\Omega^r = \Omega'$
– $\Upsilon^r = \Upsilon' \cup \{m_f + q_j + Init \mid m_f \in M_f\}$
– $m^r_0 = m'_0 + Init$,

As in the Th. 12, we can show that a sequence required by the lemma exists iff $\perp$ is reachable in N^r from m^r_0. N^r strictly emulates the product SRPN excepted that the place *Init* is added in order to allow the firing of a cut step in the root on reaching an accepting state of the product SRPN. □

We denote by $Rec(A, q_i, q_j, N, m_0, M_f)$ the function which returns true if such a sequence exists.

Lemma 17 (Acceptance). *Let $A = \langle \Sigma, Q, \delta, q_0 \rangle$ be an automaton, $F \subset Q$ a set of accepting states and $S = \langle \langle N, \lceil m_0 \rceil \rangle, \Sigma, h \rangle$ a labeled SRPN. Let M_f an effectively semilinear state set of N and $q_i, q_j \in Q$ be two automaton states. The existence of a sequence σ of $\langle N, \lceil m_0 \rceil \rangle$ such that $\lceil m_0 \rceil \xrightarrow{\sigma} \lceil m_f \rceil$ where $m_f \in M_f$ and $h(\sigma)$ is recognized by a path $q_i = q_1 \xrightarrow{a_1} q_2 \ldots \xrightarrow{a_{n-1}} q_n = q_j$ of A such that $\exists k, 1 < k \leq n \wedge q_k \in F$ is decidable.*

Proof. We construct a particular SRPN product $\langle \langle N^*, \lceil m_0^* \rceil \rangle, \Sigma, h^* \rangle$ of S and A satisfying:

- $P^* = P \cup Q \cup \overline{Q} \cup \widehat{Q} \cup \overline{\widehat{Q}} \cup \{Init\}$
- $m_0^* = m_0 + q_i + Init$

- $T^*_{el} = \{t.q.q', t.\widehat{q}.\widehat{q'} \mid (t \in T_{el}) \wedge (q, q' \in Q) \wedge (q \xRightarrow{h(t)} q')\} \cup \{t.q.\widehat{q'} \mid (t \in T_{el}) \wedge (h(t) \neq \lambda) \wedge (q \xrightarrow{h(t)} q') \wedge (q \in F)\}$
- $\forall t.q.q' \in T^*_{el}$,
  - $h^*(t.q.q') = h(t)$
  - ${W^*}^-(t.q.q') = W^-(t) + q$, ${W^*}^+(t.q.q') = W^+(t) + q'$

- $T^*_{ab} = \{t.q.q'.q'', t.\widehat{q}.\widehat{q'}.\widehat{q''}, t.q.q'.\widehat{q''} \mid (t \in T_{ab}) \wedge (q, q', q'' \in Q) \wedge (q \xRightarrow{h(t)} q'\} \cup \{t.q.\widehat{q'}.\widehat{q''} \mid (t \in T_{ab}) \wedge (q, q', q'' \in Q) \wedge (h(t) \neq \lambda) \wedge (q \xrightarrow{h(t)} q') \wedge (q \in F)\}$
- $\forall t.q.q'.q'' \in T'_{ab}$,
  - $h^*(t.q.q'.q'') = h(t)$
  - ${W^*}^-(t.q.q'.q'') = W^-(t) + q$, ${W^*}^+(t.q.q'.q'') = W^+(t) + q''$
  - $\Omega^*(t.q.q'.q'') = \Omega(t) + q' + \overline{q''}$

- $\Upsilon^* = \{m + q + \overline{q'} \mid (m \in \Upsilon) \wedge (q, q' \in Q) \wedge (q \xRightarrow{h(\tau)} q')\} \cup \{m + \widehat{q} + \overline{\widehat{q'}} \mid (m \in \Upsilon) \wedge (q, q' \in Q) \wedge (q \xRightarrow{h(\tau)} q')\} \cup \{m + q + \overline{\widehat{q'}} \mid (m \in \Upsilon) \wedge (q \in F) \wedge (q' \in Q) \wedge (h(\tau) \neq \lambda) \wedge (q \xrightarrow{h(\tau)} q')\} \cup \{m_f + \widehat{q}_j + Init \mid m_f \in M_f\}$
- $h^*(\tau) = h(\tau)$

A sequence satisfying the requirement of the lemma exists iff $\perp$ is reachable in N^* from m_0^*. The demonstration of this equivalence is similar to the proof of lemma 16. Indeed, the only difference between the SRPN $\langle N^*, \lceil m_0^* \rceil \rangle$ and the one used in this lemma is that the two places related to an automaton state are once more duplicated to indicate that a state of F has been "visited" by the current sequence. The transitions are duplicated in the same way. The reachability of $\perp$ is conditioned by the reachability of a marking of M_f at the root level (the place *Init* must be marked) in such way that the automaton reaches the state q_j having visited a state of F (the place $\widehat{q}_j$ must be marked). □

We denote by $Acc(A, F, q_i, q_j, N, m_0, M_f)$ the function which returns true if such a sequence exists. We are now in position to demonstrate the correctness of the Th. 14.

Proof. The proof is divided in two parts: looking for infinite sequences σ with $dinf(\sigma)$ finite or infinite.

($dinf(\sigma) < \infty$). We have seen that the sequences of this type can be decomposed in $\lceil m_0 \rceil \xrightarrow{\sigma_0} tr_{i_1} \xrightarrow{\sigma_1} \ldots tr_{i_k} \xrightarrow{\sigma_k} tr_{i_{k+1}} \ldots$ (whose characteristics are described in section 4.2).

1^{st} **step** We determine the possible couples of starting markings in the leaf and automaton states reached by σ_0. Indeed, as the depth of successive extended markings will be greater or equal than the current depth, the remainder of the sequence σ is only conditioned by these two informations. So, we compute the set C of couples of the form $(q, \Omega(t))$ such that there exists a sequence σ' of $(N, \lceil m_0 \rceil)$ leading to an extended marking in which the abstract transition t can be fired (necessarily in the leaf) and such that the word $h(\sigma'.t)$ is a word recognized by a path of A from q_0 to q. We have $\sigma_0 = \sigma'.t$. This computation can be done using the function *Rec* iteratively starting with the couple (q_0, m_0) with $M_f = \uparrow Pre(t)$ for each abstract transition t until saturation (i.e. when no new couple is discovered). It necessarily terminates because the number of automaton states as well as the number of abstract transitions are finite.

2^{nd} **step** We construct the ordinary net $\widehat{N}$ in the following way:
- $\widehat{P} = P \cup Q$
- $\widehat{T} = \{t.q.q' \mid (t \in T_{el}) \wedge (q, q' \in Q) \wedge (q \stackrel{h(t)}{\Longrightarrow} q')\} \cup \{t.q.q' \mid (t \in T_{ab}) \wedge (q, q' \in Q) \wedge (\exists q_1, q'_1 \in Q, q \stackrel{h(t)}{\Longrightarrow} q_1 \wedge Rec(A, q_1, q'_1, N, \Omega(t), \Upsilon) \wedge q'_1 \stackrel{h(\tau)}{\Longrightarrow} q')\} \cup \{\overline{t.q.q'} \mid (t \in T_{ab}) \wedge (q, q' \in Q) \wedge (\exists q_1, q'_1 \in Q, q \stackrel{h(t)}{\Longrightarrow} q_1 \wedge (Acc(A, F, q_1, q'_1, N, \Omega(t), \Upsilon) \vee \{q_1, q'\} \cap F \neq \emptyset) \wedge q'_1 \stackrel{h(\tau)}{\Longrightarrow} q')\}$
- $\forall t.q.q' \in \widehat{T}, \widehat{W}^-(t.q.q') = W^-(t) + q, \widehat{W}^+(t.q.q') = W^+(t) + q'$
- $\forall \overline{t.q.q'} \in \widehat{T}, \widehat{W}^-(\overline{t.q.q'}) = W^-(t) + q, \widehat{W}^+(\overline{t.q.q'}) = W^+(t) + q'$

By construction, an infinite sequence in $\langle \widehat{N}, (q, m) \rangle$ with $(q, m) \in C \cup \{(q_0, m_0)\}$ exactly corresponds to a suffix of an infinite sequence in the product SRPN which visits infinitely often a node of the extended marking. This correspondence is obtained since each transition in $\widehat{N}$ corresponds to a finite subsequence in the product between two consecutive visits of the same node. In order to be an accepting sequence, an automaton state $q \in F$ must be infinitely often reached and thus a transition in $\widehat{N}$ which corresponds to a subsequence which encounters q must be infinitely often fired. These transitions are exactly transitions $t.q.q'$ with $q' \in F$ and transitions $\overline{t.q.q'}$.

3^{rd} **step** So for each couple (q, m) in $C \cup \{(q_0, m_0)\}$, we decide whether there exists an infinite sequence in $\langle \widehat{N}, m + q \rangle$ with a transition $t.q.q'$ where $q' \in F$ fired infinitely often or a transition $\overline{t.q.q'}$ fired infinitely often. This last step can be decided using the algorithm of H.C. Yen ([Yen92]).

($dinf(\sigma) = \infty$). The checking of the existence of accepted infinite sequences is reduced to a finite graph analysis. Indeed,we build a graph where the nodes are the computed couples of the first procedure and an edge denotes that one node has been reached from the other one by a sequence increasing by one the depth of the visited extended markings and such that the intermediate subsequences never decrease the depth below its initial value. The edges are partitioned depending on the visit by the sequence of an accepting state of the Büchi automaton. Then the existence of an accepting infinite sequence is equivalent to the existence of some kind of strongly connected component. The different steps of verification are listed below:

1^{st} **step** We build two relations E and $\overline{E}$ on $C \cup \{(q_0, m_0)\}$ such that $\forall (q,m),(q',m') \in C \cup \{(q_0,m_0)\}$
 - $((q,m),(q',m')) \in E \Leftrightarrow \exists t \in T_{ab}, \exists q'_1 \in Q, Rec(A,q,q'_1,N,m,\uparrow Pre(t)) \wedge q'_1 \stackrel{h(t)}{\Longrightarrow} q' \wedge m' = \Omega(t)$
 - $((q,m),(q',m')) \in \overline{E} \Leftrightarrow \exists t \in T_{ab}, \exists q'_1 \in Q, (Acc(A,q,q'_1,N,m,\uparrow Pre(t)) \vee q' \in F) \wedge q'_1 \stackrel{h(t)}{\Longrightarrow} q' \wedge m' = \Omega(t)$

 An accepting infinite sequence σ with $dinf(\sigma) = \infty$ can be decomposed as in section 4.2. An arc of the previous graph exactly corresponds to a finite subsequence of this decomposition. It remains only to check whether an automaton state $q \in F$ is infinitely often visited by the sequence but this exactly corresponds to the infinite occurrence of an arc $e \in \overline{E}$ in an infinite path of the graph.

2^{nd} **step** So we decide whether it exists a strongly connected component of the graph $(R, E \cup \overline{E})$ having an arc of $\overline{E}$. This last step can be decided using the algorithm of Tarjan.

□

Proof of Th. 15 (Acceptance of divergent sequences)

The detection of divergent sequences is based on a lemma concerning sequences which are non observable by the automaton.

Lemma 18 (Non observation). *Let $S = \langle\langle N, \lceil m_0 \rceil\rangle, \Sigma, h\rangle$ be a labeled SRPN. Let M_f be an effectively semilinear state set of N. The existence of a sequence σ of $\langle N, \lceil m_0 \rceil\rangle$ such that $\lceil m_0 \rceil \xrightarrow{\sigma} \lceil m_f \rceil$ where $m_f \in M_f$ and $h(\sigma) = \lambda$ is decidable.*

Proof. Let N^λ be the recursive Petri net N in which the transitions of the set $\{t \in T \mid h(t) \neq \lambda\}$ have been discarded and such that if $h(\tau) \neq \lambda$ then $\Upsilon^\lambda = \emptyset$ else $\Upsilon^\lambda = \Upsilon$. Decide if such a sequence exists is equivalent to decide if $\mathcal{L}(N^\lambda, \lceil m_0 \rceil, M_f) \neq \emptyset$.

□

We denote by $NonObs(N, m_0, M_f)$ the function which returns true if such a sequence exists. We are now in position to demonstrate the correctness of the Th. 15.

Proof. Again, two kinds of infinite sequences have to be detected. The first kind concerns sequences for which the depth of the extended markings visited is bounded. Such sequences are detected by the three following steps:

1^{st} **step** We construct the ordinary net $\widetilde{N}$ in the following way:
 - $\widetilde{P} = P \cup Q$

- $\widetilde{T} = \{t.q.q' \mid (t \in T_{el}) \wedge (q, q' \in Q) \wedge (q \xRightarrow{h(t)} q')\} \cup$
  $\{t.q.q' \mid (t \in T_{ab}) \wedge (q \in Q \setminus F) \wedge (q' \in Q) \wedge Rec(A, q, q', N, \Omega(t), \Upsilon) \wedge \neg NonObs(N, \Omega(t), \Upsilon)\} \cup$
  $\{\overline{t.q.q'} \mid (t \in T_{ab}) \wedge (q \in F) \wedge (q' \in Q) \wedge NonObs(N, \Omega(t), \Upsilon)\}$
- $\forall t.q.q' \in \widetilde{T}, \widetilde{W}^-(t.q.q') = W^-(t) + q, \widetilde{W}^+(t.q.q') = W^+(t) + q'$
- $\forall \overline{t.q.q'} \in \widetilde{T}, \widetilde{W}^-(\overline{t.q.q'}) = W^-(t) + q, \widetilde{W}^+(\overline{t.q.q'}) = W^+(t) + q'$

2^{nd} **step** We compute the set of couples C of the form $(q_i, \Omega(t_j))$ such that there exists a firing sequence σ of (N, m_0) leading to an extended marking in which the abstract transition t_j can be fired (necessarily in the leaf) and such that the word $h(\sigma.t_j)$ is a word recognized by a path of A from q_0 to q_i. This computation can be done using the function *Rec* iteratively.

3^{rd} **step** For each couple (q, m) in $C \cup \{(q_0, m_0)\}$, decide if it exists an infinite sequence σ_{el} in $\langle \widetilde{N}, m + q\rangle$ for which the set of transitions fired infinitely often is a subset of $\{t.q.q', \overline{t.q.q'} \in \widetilde{T} \mid h(t) = \lambda \wedge q \in F\}$. This last step can be decided using the algorithm of H.C. Yen ([Yen92]). If such a sequence exists return *true* else return *false*.

The second kind of infinite sequences are the ones for which such a bound does not exist and they are detected applying the following two steps:

1^{st} **step** We construct a set R and two relations E and $\overline{E}$ such that
- $(m_0, q_0) \in R$
- $\forall (m, q) \in R, \exists t \in T_{ab}, q' \in Q$ such that $Rec(A, q, q', N, m, \uparrow Pre(t)) \Rightarrow (\Omega(t), q') \in R \wedge ((m, q), (\Omega(t), q')) \in E$
- $\forall (m, q) \in R, \exists t \in T_{ab}$ such that $q \in F \wedge NonObs(N, m, \uparrow Pre(t)) \Rightarrow (\Omega(t), q) \in R \wedge ((m, q), (\Omega(t), q)) \in \overline{E}$

2^{nd} **step** Decide if there exists a strongly connected component of the graph $(R, E \cup \overline{E})$ using only some arcs of $\overline{E}$ and having a node (q, m) such that $q \in F$. This last step can be decide using the classical algorithm of Tarjan. If such a component exists return *true* else return *false*.

The demonstration of the correctness of these decision procedures is similar to the one presented for Theorem 14. □