Place bisimulations for the reduction of labeled Petri nets with silent moves*

C. Autant, W. Pfister and Ph. Schnoebelen†

Lab. LIFIA - Institut Imag - CNRS,
Grenoble - FRANCE

Keywords: semantics of concurrency, Petri nets, bisimulation theory.

Abstract

Place bisimulations are equivalences between places of Petri nets. They can be used to fuse bisimilar places in a net: this gives a smaller net with identical behaviour. In this article, we extend the basic theory (from [ABS91, AS92]) to labeled nets with silent (invisible) transitions. We describe a simple algorithm computing τ-place bisimulation.

Introduction

Petri nets are widely used as an operational model of concurrent systems. They offer a much richer structure than labeled transition systems (LTS) used e.g. in the standard semantics of CCS, and they do not suffer from the combinatorial explosion that takes place when one puts two LTS in parallel.

Our interest here is in methods that reduce a given labeled Place/Transition Petri net without modifying its behaviour. Such methods are useful as a preprocessing step in model-checkers for Petri nets, as an optimization step in a compiler generating Petri nets, ... Today, similar methods for LTS are widely used in successful software tools based on the LTS framework [IP91].

We proposed in [ABS91, AS92] a new concept, called *place bisimulations*, that allows to fuse "equivalent" places in a labeled P/T net, preserving in a strong sense (i.e. modulo bisimulation) the behaviour of the net. Strong points of this approach are:

1. it is efficient and runs in polynomial time,
2. it preserves more than just the interleaving aspects of the behaviour [AS92],
3. it is "non-local" and detects in one pass when distant parts of a net can be shared,
4. it is not limited to C/E nets where in general equivalent places cannot be fused.

Apart from place-bisimulation approaches, there already is a rich theory of reductions of Petri nets [Ber85]. However

*A short version of this work has been presented at the ICCI'94, Peterborough, Ontario, May 1994.
†LIFIA-IMAG, 46 Av. Félix Viallet, 38031 GRENOBLE Cedex, FRANCE. E-mail:{cyril,pfister,phs}@lifia.imag.fr

1. these approaches are usually geared towards preserving *specific properties* of nets. One reduction method preserves liveness of transitions, another one preserves boundedness, ...

2. the methods are "local" in essence. They detect a given pattern locally and reduce the net accordingly. Then the net must be examined again for new occurrences of the reduction pattern. This is a strong limitation of the methods.

Regarding this second point, place bisimulation can be seen as a non-local generalization of the notion of equivalent places (TP3) found in [Ber85]. It allows greater reductions and leads to much more efficient algorithms.

In [Pfi92], we investigated practical aspects of the place bisimulation approach to the reduction of nets. A set of tests was obtained by compiling LOTOS specifications into Petri nets, using the LOTOS compiler developed by H. Garavel from VERIMAG. We could simplify (sometimes greatly, sometimes marginally) the resulting nets. The general conclusion of this study was that

- because of its low-complexity, the method can be applied in practical situations,
- it performs non-trivial reductions,
- but it would find much better reductions *if it could take into account the fact that some transitions are internal and not visible from the outside.*

Indeed, in the LTS framework at least, it is well-known that much reduction can be performed when abstracting from the internal moves that naturally result from synchronizations in hierarchically built systems. Several software tools, such as the Concurrency Workbench [CPS93], can simplify (often drastically) a given LTS modulo τ-bisimulation (aka observational equivalence) in time $O(n^3)$ where n is the size of the LTS.

In this article, we define *τ-place bisimulation*, a notion of place bisimulation that is suited to labeled P/T nets with silent moves. By studying effective algorithms for τ-place bisimulation, we show how this new notion can be used in practice. This involves the introduction of several new developments before we are finally able to propose a simple and efficient algorithm. It turns out that the algorithmic problems are much more complicated than in the simpler LTS framework.

In section 1 we recall basic definitions and notations on P/T nets and bisimulations. In section 2 we recall the basic results on place bisimulations with special emphasis on algorithmic aspects. Then in section 3 we define and study τ-place bisimulation and first describe two specific difficulties which forbid a straightforward approach to algorithms for the computation of τ-place bisimulation. The solution involves τp-place bisimulation, defined in section 4 and a method of p-saturation for nets, described in sections 5 to 7. We conclude by studying the relevance of this work and mentioning directions for promising extensions.

Longer proofs have been relegated into an Appendix when they disturb the exposition.

1 Basic notions

(We just recall the basic notations and results we shall need. The reader not familiar with P/T nets or the bisimulation approach to concurrency should consult standard introductory textbooks for motivations and examples.)

1.1 Labeled P/T nets

We consider an alphabet $A = \{a, b, \ldots\}$ with $\tau \notin A$. τ classically denotes internal (invisible) steps, and we write A_τ for $A \cup \{\tau\}$. A *labeled Place/Transition Petri net* (or simply a net) is a structure of the form $N = \langle P_N, T_N, F_N, l_N \rangle$ where

- $P_N = \{p, q, \ldots\}$ is a set of *places*,
- $T_N = \{t, u, \ldots\}$ is any set of *transitions*, disjoint from P_N,
- $F_N : (P_N \times T_N \cup T_N \times P_N) \to \mathbb{N}$ is the *flow relation* (with weights),
- $l_N : T_N \to A_\tau$ is the *labeling* of transitions.

For a transition $t \in T_N$, the *precondition* (resp. *postcondition*) of t, written ${}^\bullet t$ (resp. $t^\bullet$), is the multiset of places given by ${}^\bullet t(p) \stackrel{\text{def}}{=} F_N(p,t)$ (resp. $t^\bullet(p) \stackrel{\text{def}}{=} F_N(t,p)$) for any $p \in P_N$. We use the classical notations $M + M'$, $M \subseteq M'$ and $M - M'$ (when $M' \subseteq M$) for multisets. $M \setminus M'$ denotes $M - (M' \cap M)$.

We use the classical graphical representation of nets, with circles for places and boxes for transitions. The box for t carries the label $l_N(t)$. (A transition labeled with τ is called a *τ-transition.*) The flow relation is depicted with arrows leading from places to transitions (and from transitions to places) with e.g. n arrows from p to t if $F_N(p,t) = n$.

A net is *finite* if $P_N \cup T_N$ is. In this article, we always explicitly state when we restrict to finite nets.

1.2 Behavior of nets

A *marking* of a net N is a finite multiset $M = \{p_1, \ldots, p_n\}$ of places. (We write $\mathcal{M}(P_N)$ for the set of all finite multisets over P_N.) A marking is a global state of N, depicted by putting $M(p)$ tokens in every place $p \in P_N$. A *marked net* is a pair (N, M_{init}) of a net and a distinguished marking (the *initial marking*).

A transition $t \in T_N$ is *enabled* (or *firable*) in state M, written $M \xrightarrow{t}$, iff ${}^\bullet t \subseteq M$. Then firing it yields a new marking $M' \stackrel{\text{def}}{=} M - {}^\bullet t + t^\bullet$. This step is written $M \xrightarrow{t} M'$. We extend these notions to sequences and, given $\sigma = t_1 \ldots t_n \in T_N^*$, we write $M \xrightarrow{\sigma} M'$ when $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \cdots \xrightarrow{t_n} M_n = M'$ for some markings $M_1, \ldots$

We extend the definition of ${}^\bullet t$ to sequences $\sigma \in T_N^*$ through the following

$$\begin{aligned} {}^\bullet \epsilon &\stackrel{\text{def}}{=} \emptyset \\ {}^\bullet(t_1 \ldots t_n) &\stackrel{\text{def}}{=} {}^\bullet t_1 + ({}^\bullet(t_2 \ldots t_n) \setminus t_1{}^\bullet) \end{aligned}$$

and similarly for $\sigma^\bullet$. Clearly $M \xrightarrow{\sigma}$ iff ${}^\bullet\sigma \subseteq M$. $l_N(\sigma) \in A_\tau^*$ is defined in the obvious way.

A multiset $U = \{t_1, \ldots, t_n\} \in \mathcal{M}(T_N)$ of transitions are *concurrently enabled* in M, written $M \xrightarrow{U}$, iff ${}^\bullet U \subseteq M$ (with ${}^\bullet\{t_1, \ldots, t_n\} \stackrel{\text{def}}{=} {}^\bullet t_1 + \cdots + {}^\bullet t_n$), that is, when M has enough tokens to enable all transitions of U at the same time. They can be fired concurrently (in parallel) and we write $M \xrightarrow{U} M'$ with $M' \stackrel{\text{def}}{=} M - {}^\bullet U + U^\bullet$.

In this article, we often compare transitions, sequences of transitions, ... writing e.g. "$\xrightarrow{\sigma'}\subseteq\xrightarrow{\sigma}$". This notation means that (1) $M \xrightarrow{\sigma'} M'$ implies $M \xrightarrow{\sigma} M'$ and (2) $l_N(\sigma) = l_N(\sigma')$. We write $\xrightarrow{\sigma}=\xrightarrow{\sigma'}$ when $\xrightarrow{\sigma}\subseteq\xrightarrow{\sigma'}\subseteq\xrightarrow{\sigma}$. All these extend to multisets of transitions (fired concurrently) with $l_N(U) \in \mathcal{M}(A_\tau)$.

1.3 Quotient nets

We are interested into nets reduced by identifying "equivalent" places. Formally, given an equivalence E over P_N, we write $[p]_E$ for the equivalence class of a place p and $M/_E$ for the multiset $\{[p_1]_E, \ldots, [p_n]_E\}$ if $M = \{p_1, \ldots, p_n\}$.

Definition 1.1 ***(Quotient net)*** *For a net N and an equivalence E on P_N*

$$N/_E \stackrel{def}{=} \langle P_N/_E, T_N, F_N/_E, l_N\rangle$$

where $F_N/_E$ is s.t. if ${}^\bullet t = M$ in N then ${}^\bullet t = M/_E$ in $N/_E$ (and similarly for $t^\bullet$).

Figure 1 below has an example.

Note that if $M \xrightarrow{t} M'$ is a step in N, then $M/_E \xrightarrow{t} M'/_E$ is a step in $N/_E$ but the converse does not hold in general. So that, in general, N and $N/_E$ do not have the same behavior.

1.4 Bisimulations

Given a net N, we say that a symmetric relation $R \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_N)$ between the markings of N has *the transfer property* iff for all $M_1 R M_2$, for all step $M_1 \xrightarrow{t} M_1'$ there is an imitating step $M_2 \xrightarrow{u} M_2'$ with $M_1' R M_2'$ and $l_N(t) = l_N(u)$.

We write $R : N \underline{\leftrightarrow} N$ when R is symmetric and has the transfer property, and then we say R is a *(strong) bisimulation.* Then, if $M_1 R M_2$ for some markings M_1 and M_2, we say that the marked nets (N, M_1) and (N, M_2) are (strongly) bisimilar, written $(N, M_1)\underline{\leftrightarrow}(N, M_2)$. (This notion can be defined between marked nets with different underlying nets: we say that $(N_1, M_1)\underline{\leftrightarrow}(N_2, M_2)$ when M_1 and M_2 are bisimilar markings in the union net $N_1 + N_2$.) It is well known that two bisimilar nets have the same (interleaving) branching-time behavior and e.g. satisfy the same formulas of a branching-time temporal logic like the modal μ-calculus used in the Concurrency Workbench.

1.5 τ-bisimulations

Strong bisimulations do not treat τ-transitions as being internal, i.e. unobservable from outside. Between two strongly bisimilar marked nets, each τ-step can be precisely imitated by one corresponding τ step. τ-bisimulations are a variant of bisimulations designed to treat τ-labels as special internal labels. Formally, we say that a relation $R \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_N)$ has the τ-transfer property iff whenever $M_1 R M_2$ any step $M_1 \xrightarrow{t} M_1'$ can be imitated by a sequence $M_2 \xrightarrow{u_1} \ldots \xrightarrow{u_n} M_2'$ with $M_1' R M_2'$ and $l_N(t) =_\tau l_N(u_1 \ldots u_n)$ ($\stackrel{\text{def}}{=} l_N(u_1) \ldots l_N(u_n)$) where "$=_\tau$" denotes *equality modulo* τ between words of A_τ^*.

Because we will often have to deal with such sequences, we introduce the special notation $T_N^\times$ for the subset of T_N^* that only contains sequences with at most one transition with a visible (non τ) label. Similarly, $\mathcal{M}^\times(T_N)$ is the subset of $\mathcal{M}(T_N)$ where no multiset U has more than one visible transition.

Instead of $\xrightarrow{\sigma}\subseteq\xrightarrow{\sigma'}$ we write $\xrightarrow{\sigma}\subseteq_\tau\xrightarrow{\sigma'}$ when the requirement on labels is that $l_N(\sigma) =_\tau l_N(\sigma')$.

2 Place bisimulations (without τ's)

In this section we recall some results about *strong* place bisimulations (i.e. not dealing with τ's) that explain how place bisimulations can be used to simplify Petri nets.

Place bisimulations are relations between places and not between markings. They can be lifted from places to markings. Given $B \subseteq P_N \times P_N$, we lift B into $\overline{B} \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_N)$ by

$$p_1 B q_1 \wedge \cdots \wedge p_n B q_n \;\Rightarrow\; \{p_1, \ldots, p_n\}\overline{B}\{q_1, \ldots, q_n\}$$

That is, two markings are related by $\overline{B}$ iff their tokens can be paired in a way agreeing with B. If B is an equivalence on P_N then $\overline{B}$ is an equivalence on $\mathcal{M}(P_N)$ [ABS91].

The following definition is fundamental.

Definition 2.1 ***[AS92]*** *A relation $B \subseteq P_N \times P_N$ is a* place bisimulation *over N, written $B : N \sim N$, iff $\overline{B} : N \underline{\leftrightarrow} N$, i.e. if $\overline{B}$ is a bisimulation.*

The idea of place bisimulations originates with [Old89]. The new, modified, definition is from [ABS91, AS92].

Proposition 2.2 ***[ABS91, AS92]*** *Given a net N, there exists a largest reflexive place bisimulation over N, written $B(N)$.*

$B(N)$ is an equivalence on P_N so that we may define $N/_\sim \stackrel{\text{def}}{=} N/_{B(N)}$ (and we extend this notation to markings). $N/_\sim$ is smaller than N but it is equivalent because

Theorem 2.3 *For any net N and marking M, $(N, M)\underline{\leftrightarrow}(N/_\sim, M/_\sim)$.*

Proof Standard because $\overline{B(N)}$ is a bisimulation. □

([AS92] shows in which sense $N/_\sim$ is the smallest quotient net equivalent to N.)

In this article, we investigate the algorithmic aspects of place bisimulation. Indeed, at first sight it is not clear how to compute $B(N)$. It is not even clear, given a relation $B \subseteq P_N \times P_N$, how to see whether B is a place bisimulation between N and itself. In principle, given some symmetric B, we have to check the transfer property in an infinite number of situations, namely for all the pairs $M_1\overline{B}M_2$ and all steps $M_1 \xrightarrow{t} M_1'$. It turns out that this test can be restricted to a finite number of cases.

Definition 2.4 *We say that a relation $B \subseteq P_N \times P_N$ has the* weak transfer property *iff, for all $t \in T_N$, for all $p \in {}^\bullet t$, for all $q \in P_N$ s.t. pBq, there is a $u \in T_N$ s.t. $l_N(t) = l_N(u)$ and, writing M_2 for ${}^\bullet t - \{p\} + \{q\}$, we have $M_2 \xrightarrow{u} M_2'$ with $t^\bullet\overline{B}M_2'$.*

That is, the weak transfer property is the transfer property of place bisimulation

- restricted to steps $M_1 \xrightarrow{t} M_1'$ where $M_1 = {}^\bullet t$,
- and from markings M_2 differing from M_1 by only one place.

This weaker condition only involves a finite number of tests, however it is strong enough because

Proposition 2.5 *[AS92] The largest reflexive and symmetric B having the weak transfer property is $B(N)$.*

So that the following algorithm can be used to compute $B(N)$:

Definition 2.6 ***(An algorithm for** $B(N)$**)***

input a labeled P/T net N.

output the relation $B(N)$.

step 1 Set $B = P_N \times P_N$.

step 2 Check if B has the weak transfer property:

- If it has, then B is $B(N)$.
- Otherwise, there is a $t \in T_N$, a $p \in {}^\bullet t$, a $q \in P_N$ with pBq s.t. ${}^\bullet t \xrightarrow{t} t^\bullet$ cannot be imitated from ${}^\bullet t - \{p\} + \{q\}$. In this case, remove the pairs (p, q) and (q, p) from B and go back to step 2.

When we consider only nets of a fixed degree d (that is such that $|{}^\bullet t| + |t^\bullet| \leq d$ for all $t \in T_N$), the above algorithm can be made to run in time $O(|P_N|^2 \times |T_N|^2)$ (with a constant factor depending on d) [Pfi92].

We implemented this algorithm. Its main limitation is that it respects a version of bisimulation which does not abstract from silent steps. Clearly, much more important reductions could be obtained by a version of place bisimulation suited to τ-bisimulation.

3 Place bisimulations with τ's

A version of place bisimulation suited to τ-bisimulation is

Definition 3.1 *A relation $B \subseteq P_N \times P_N$ is a τ-place bisimulation, written $B : N \sim_\tau N$, iff $\overline{B}$ is a τ-bisimulation.*

Proposition 3.2 *Given a net N, there exists a largest reflexive τ-place bisimulation over N, written $B_\tau(N)$. Furthermore $B_\tau(N)$ is an equivalence over P_N.*

Because $B_\tau(N)$ is an equivalence, we may define $N/_{\sim_\tau} \stackrel{\text{def}}{=} N/_{B_\tau(N)}$. It is equivalent to N in the following sense:

Theorem 3.3 *For all marked nets (N, M), $(N, M) \underline{\leftrightarrow}_\tau (N/_{\sim_\tau}, M/_{\sim_\tau})$.*

Example 3.4 Consider Figure 1. Here $B_\tau(N)$ is the equivalence identifying p_1, p_3, p_4 on one hand, and p_0, p_2 on the other hand. The quotient net $N/_{\sim_\tau}$ has only 2 places (q for $\{p_1, p_3, p_4\}$, q' for $\{p_0, p_2\}$) and only 3

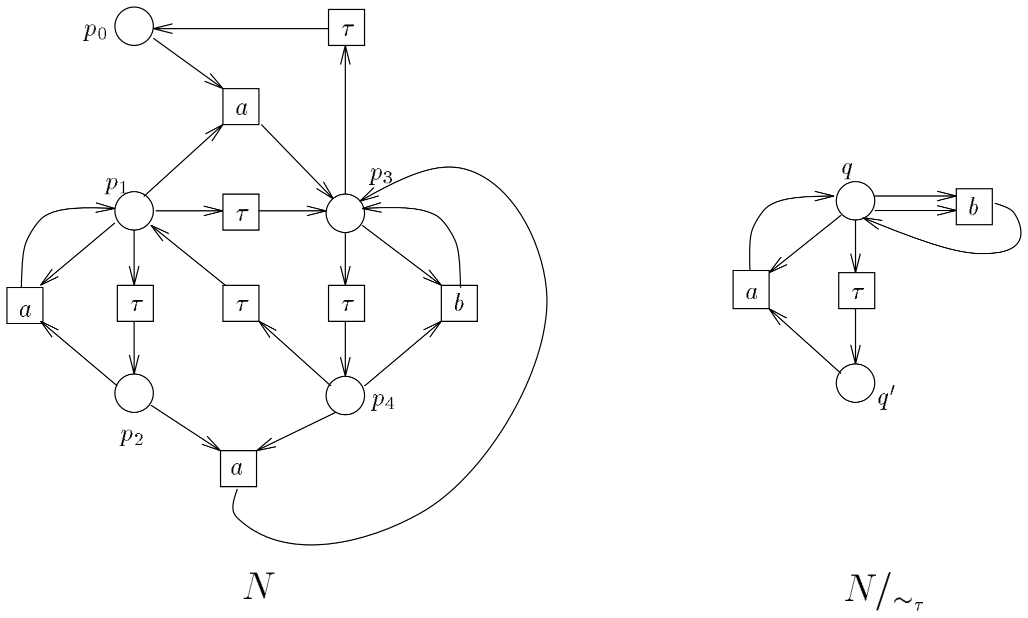


Figure 1: A net N and its quotient by τ-place bisimulation: $q = \{p_1, p_3, p_4\}$ and $q' = \{p_0, p_2\}$

transitions because fusing places introduced redundancies between the original 9 transitions. Assume we now plan to start N with initial marking $M_0 = \{p_1, p_2, p_3\}$. Then a similar behaviour will be obtained by starting $N/_{\sim_\tau}$ with initial marking $M_0/_{\sim_\tau} = \{q, q, q'\}$. In fact, for an external user, the behaviour of $N/_{\sim_\tau}$ cannot be distinguished from the behaviour of N, as is well-known from general bisimulation theory.

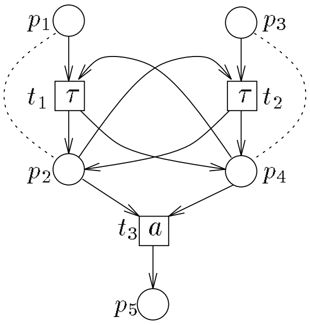
There remains to compute $B_\tau(N)$ for labeled nets. A first result is

Proposition 3.5 *It is possible to compute $B_\tau(N)$ for any finite N.*

Proof Here it suffices to be able to say if a given B is a τ-place bisimulation. Given a candidate $B \subseteq P_N \times P_N$, it is enough to check the transfer property for steps $M_1 \xrightarrow{t} M_1'$ where $M_1 = {}^\bullet t$. For such a step, there only exists a finite number of markings M_2, M_2' s.t. $M_1 \overline{B} M_2$ and $M_1' \overline{B} M_2'$. Then, for any such M_2, M_2', it is decidable through general reachability theory of nets [Kos82] whether there is a sequence $\sigma \in T_N^*$ s.t. $M_2 \xrightarrow{\sigma} M_2'$ with $l_N(\sigma) =_\tau l_N(t)$. $\square$

However, the complexity of this approach is very high. It involves the solution of an exponential number of reachability problems. And it is well-known that these reachability problems have non-elementary complexity. Our goal in this article is to adapt in the best possible way the efficient methods we developed for strong place-bisimulation to the new framework with τ's.

This turns out to be more difficult than one would expect. A first attempt fails because the weak τ-transfer property is not enough to ensure the full τ-transfer property, that is, Proposition 2.5 does not generalize to τ-place bisimulation. We show a simple example where the depicted relation



$$B \stackrel{\text{def}}{=} Id_{P_N} + \{(p_1, p_2), (p_2, p_1), (p_3, p_4), (p_4, p_3)\}$$

has the weak τ-transfer property but is larger than $B_\tau(N)$. Indeed $B_\tau(N) = Id_{P_N}$: from marking $M = \{p_1, p_3\}$ it is not possible to fire any transition while from $M' = \{p_2, p_4\}$ we can fire a visible transition. Note that $M\overline{B}M'$.

A second attempt fails because the classical "τ-saturation" approach used in transition systems to reduce τ-bisimulation to strong bisimulation of a saturated system (see [KS90]), is not applicable here. We spend some time explaining why.

Definition 3.6 *A net N is τ-saturated iff for any sequence $\sigma \in T_N^{\times}$ there is a $t \in T_N$ s.t. $\xrightarrow{\sigma} \subseteq_\tau \xrightarrow{t}$.*

τ-saturated nets are interesting because if N is τ-saturated then $B_\tau(N) = B(N)$. If we could add new transitions to a non-saturated net N in such a way that

- we get a saturated net N',
- such that N and N' have the same τ-place bisimulations.

then we could compute $B_\tau(N)$ easily by reducing this to a computation of $B(N')$. This is the approach classically used for LTS [CPS93].

However this approach cannot succeed here because only degenerate τ-saturated nets exist:

Proposition 3.7 *If N is τ-saturated and has non-trivial τ-transitions then N is infinite.*

where we say a τ-transition t is *trivial* if ${}^\bullet t = t^\bullet$.

Proof Assume N is τ-saturated and contains a τ-transition $t \in T_N$ with ${}^\bullet t \neq t^\bullet$, so that $({}^\bullet t \setminus t^\bullet) + (t^\bullet \setminus {}^\bullet t)$ is not empty. W.l.o.g. assume ${}^\bullet t \setminus t^\bullet$ is not empty and contains some $p \in P_N$. Then, writing t^k for the sequence made of k copies of t, p appears at least k times in ${}^\bullet(t^k) \setminus (t^k)^\bullet$. For every sequence t^k, there is a transition $u_k \in T_N$ with $\xrightarrow{t^k} \subseteq_\tau \xrightarrow{u_k}$, and the u_i's must be all distinct because the ${}^\bullet(t^k) \setminus (t^k)^\bullet$ are all different. Therefore T_N is infinite. $\square$

4 τp-place bisimulations

In this section, we define τp-bisimulations, a new notion of bisimulation that will allow us to solve satisfactorily our problem of computing $B_\tau(N)$. τp-bisimulation is useful because (1) it yields simple algorithms, and (2) τ-bisimulation can often be reduced to τp-bisimulation.

Definition 4.1 *A relation $R \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_N)$ is a τp-*bisimulation *over N, written $B : N \underline{\leftrightarrow}_{\tau p} N$, iff for all $M_1 R M_2$ for all $M_1 \xrightarrow{t} M_1'$, there exists a multiset $U \in \mathcal{M}(T_N)$ with $l_N(t) =_\tau l_N(U)$ and s.t. $M_2 \xrightarrow{U} M_2'$ with $M_1' R M_2'$.*

The τp-transfer property is intermediary between strong bisimulation and τ-bisimulation. A single step needs not be imitated by a single corresponding step. However an imitating sequence is not allowed in general. Only a *concurrent* step can be used. Therefore we clearly have

Proposition 4.2 *1. $R : N \underline{\leftrightarrow} N$ implies $R : N \underline{\leftrightarrow}_{\tau p} N$, and*

2. $R : N \underline{\leftrightarrow}_{\tau p} N$ implies $R : N \underline{\leftrightarrow}_{\tau} N$.

Definition 4.3 *A relation $B \subseteq P_N \times P_N$ is a τp-*place bisimulation *iff $\overline{B} : N \underline{\leftrightarrow}_{\tau p} N$.*

Proposition 4.4 *There is a largest reflexive τp-place bisimulation over N, written $B_{\tau p}(N)$. It is an equivalence over P_N.*

With τp-place bisimulations, the weak transfer property does not exhibit the problems we had with τ-place bisimulations. We can generalize Proposition 2.5:

Definition 4.5 *We say that a relation $B \subseteq P_N \times P_N$ has the weak τp-transfer property if it has the τp-transfer property for all situations $M_1 \overline{B} M_2$ and $M_1 \xrightarrow{t} M_1'$ s.t. $M_1 = {}^\bullet t$ and $M_2 = M_1 - \{p\} + \{q\}$ for some pBq.*

Proposition 4.6 *If B is reflexive and has the weak τp-transfer property, then B^*, the transitive closure of B, is a τp-place bisimulation.*

A corollary is that we now have a simple algorithm computing $B_{\tau p}(N)$:

Definition 4.7 ***(An algorithm for** $B_{\tau p}(N)$**)***

input a labeled P/T net N.

output the relation $B_{\tau p}(N)$.

step 1 Set $B = P_N \times P_N$.

step 2 Check if B has the weak τp-transfer property:

- If it has, then B is $B_{\tau p}(N)$.
- Otherwise, there is a $t \in T_N$, a $p \in {}^\bullet t$, a $q \in P_N$ with pBq s.t. ${}^\bullet t \xrightarrow{t} t^\bullet$ cannot be imitated from $M_2 = {}^\bullet t - \{p\} + \{q\}$ by a concurrent step $M_2 \xrightarrow{U} M_2'$. In this case, remove the pairs (p, q) and (q, p) from B and go back to step 2.

Because we have to consider concurrent steps as potential imitations of t, the above algorithm runs in time $O(|P_N|^2 \times |T_N|^{d+1})$ when restricted to nets of degree no larger than d.

5 p-saturated Petri nets

We describe a class of nets for which τ-place bisimulation reduces to τp-place bisimulation: a p-saturated net is a net where every sequence of transitions with at most one visible label can be replaced by a *concurrent step* with same visible label. Formally

Definition 5.1 *A sequence $\sigma \in T_N^\times$ is* covered *in some net N iff there exists a $U \in \mathcal{M}^\times(T_N)$ s.t. $\xrightarrow{\sigma} \subseteq_\tau \xrightarrow{U}$.*
*We say that N is p-*saturated *iff every $\sigma \in T_N^\times$ is covered in N.*

In p-saturated nets, τ-place bisimulation reduces to τp-place bisimulation. Obviously

Proposition 5.2 *If N is p-saturated then $R : N \underline{\leftrightarrow}_\tau N$ iff $R : N \underline{\leftrightarrow}_{\tau p} N$.*

Corollary 5.3 *If N is p-saturated then $B_\tau(N) = B_{\tau p}(N)$.*

Now a first problem is to see whether a given net is p-saturated. We reduce this problem to a check for $2p$-saturation through the following

Definition 5.4 *A net is $2p$-saturated iff all sequences $\sigma \in T_N^\times$ of length 2 are covered.*

Proposition 5.5 *A net is p-saturated iff it is $2p$-saturated.*

is fundamental. The proof is not straightforward. See Appendix A. It involves a close examination of the process structure of firing sequences $M \xrightarrow{\sigma} M'$.

Reducing p-saturation to $2p$-saturation entails

Corollary 5.6 *There exists a simple polynomial-time algorithm telling if a finite net is p-saturated.*

6 p-saturation of Petri nets

In this section we investigate methods reducing the problem of computing $B_\tau(N)$ for arbitrary nets to the same problem on p-saturated nets that we solved in section 5. The basic idea involves a notion of *saturation* of a net. Intuitively, a saturation of a net N is a net obtained from N by adding (without modifying τ-bisimilarity) some new transitions to cover all sequences $\sigma \in T_N^\times$. Formally

Definition 6.1 *A net N^* is a (finite) p-*saturation *of N iff*

1. *N^* is N plus a (finite) set of transitions,*
2. *N^* is p-saturated,*
3. *$Id_{P_N} : N \sim_\tau N^*$.*

*We say that N is p-*saturable *if it admits a finite p-saturation.*

Proposition 6.2 *If N^* is a p-saturation of N, then $B_\tau(N) = B_{\tau p}(N^*)$.*

This fundamental result can be used to compute $B_\tau(N)$:

Definition 6.3 ***(An algorithm for $B_\tau(N)$)***

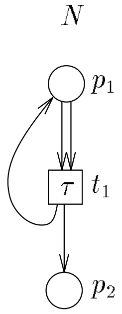
input a (p-saturable) labeled P/T net N.

output the relation $B_\tau(N)$.

step 1 Compute a p-saturation N^* of N,

step 2 Compute $B_{\tau p}(N^*)$. This is $B_\tau(N)$.

Step 2 just uses Definition 4.7, but step 1 is more problematic. We describe some of the difficulties before we give a simple algorithm in the next section.



First of all

Remark 6.4 *Not all finite nets are p-saturable.*

Example 6.5 It can be proved directly that the net N given here has no finite p-saturation but this is quite complex and it is easier to rely on Theorem 7.4. Observe that a p-saturation N^* must have transitions covering $t_1.t_1$, $t_1.t_1.t_1$, ... which have larger and larger preconditions.

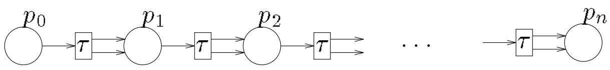
However large classes of nets are p-saturable. See Theorem 6.8. Furthermore, we have for example

Proposition 6.6 *All finite τ-graphs nets N (over a fixed finite alphabet) have a finite p-saturation N^* with $|T_{N^*}|$ bounded by $O(|T_N|^2)$.*

(where a τ-graph net is a P/T net where all τ-transitions have $|{}^\bullet t|$ and $|t^\bullet|$ at most 1. Our running example, in Figure 1, is a τ-graph.)

However, observe that p-saturations can be large and in general $|T_{N^*}|$ cannot be bounded by an exponential of $|T_N|$.

Example 6.7 The following net



is p-saturable but its smallest p-saturation has $S(0) + \cdots + S(n-1)$ transitions, where $S(k)$ is the number of solutions (in non-negative integers) of the equation

$$X_0 + 2X_1 + \cdots + 2^k X_k = 2^{k+1}$$

$S(k)$ is the number of partitions of 2^{k+1} into powers of 2. This problem was studied by K. Mahler in the forties. Roughly, $S(k)$ grows like 2^{k^2}.

We conclude this section with the following

Theorem 6.8 ***(Pfister)*** *It is decidable whether a net is p-saturable.*

the proof of which is outside the scope of this article. In essence, a net is p-saturable unless it contains the pattern exhibited in Example 6.5 appearing in a context where it is not subsumed by other transitions.

7 A covering algorithm for p-saturation

A p-saturation of N is obtained by adding new transitions to N so that every sequence in $T_N^\times$ is covered. Therefore the following definition is natural:

Definition 7.1 *We write $N \leadsto_{t_1,t_2} N'$ if $t_1.t_2$ is not covered in N and N' is N to which has been added a new transition t with*

$$^\bullet t \stackrel{def}{=} {}^\bullet(t_1.t_2),\ t^\bullet \stackrel{def}{=} (t_1.t_2)^\bullet \text{ and } l_{N'}(t) =_\tau l_N(t_1.t_2)$$

We write $N \leadsto N'$ if $N \leadsto_{t_1,t_2} N'$ for some (t_1, t_2). Clearly, N is $2p$-saturated iff $N \not\leadsto$ (i.e. there is no N' s.t. $N \leadsto N'$).

A *covering sequence* for N is a (possibly infinite) sequence of the form $N \leadsto N_1 \leadsto \cdots N_k \cdots$. A maximal covering sequence is infinite or ends with some $2p$-saturated net N_k. Thus a finite maximal covering sequence is called a *saturating sequence* for N. Note that in general there are several (maximal) covering sequences starting from a given N.

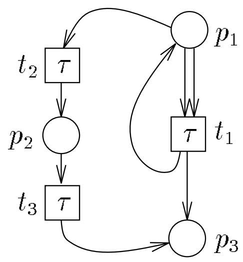
Proposition 7.2 *If $N \leadsto N'$, then $Id_{P_N} : N \sim_\tau N'$.*

Corollary 7.3 *If $N \leadsto \cdots N'$ is a saturating sequence, then N' is a p-saturation of N.*

Therefore, building covering sequences is a simple way of computing p-saturations of nets. Indeed, this method is complete:

Theorem 7.4 *If N is p-saturable, then there exists a saturating sequence for N.*

See Appendix C for a proof.



Now, even if there exists a saturating sequence for a given N, not all maximal saturation sequences are finite. Here is an example: $N \leadsto_{t_2.t_3} N'$ is a saturating sequence. But it is possible to build an infinite covering sequence by adding some transition u_1 covering $t_1.t_1$, then some u_2 covering $u_1.u_1$ and so on *ad infinitum*, without ever covering $t_2.t_3$.

We now turn to the problem of finding a saturating sequence when one exists, i.e. when N is p-saturable. Define a *covering strategy* as any (not necessarily deterministic) criterion choosing a way to extend a covering sequence $N_0 \leadsto N_1 \leadsto \cdots N_k$ by some $N_k \leadsto N_{k+1}$ when N_k is not p-saturated. Say a strategy is *complete* if it never allows an infinite covering sequence starting from a p-saturable net. We saw that not all strategies are complete. E.g. in the previous example, strategy "*never cover $t_2.t_3$ if something else is possible*" is not complete. But it is not "fair".

Definition 7.5 *An infinite covering sequence $N_0 \leadsto \cdots N_k \cdots$ is* fair *if no pair $t.t'$ remains not covered in all N_i, $i \geq k$ (for some k).*
A fair strategy *is a strategy that only produces fair (or finite) covering sequences.*

Theorem 7.6 *All fair saturation strategies are complete.*

(see Appendix B for a proof.)

8 Simplifying Petri nets

The previous section gives us a simple algorithm finding p-saturations of p-saturable nets. We just have to use a fair covering strategy to be sure that a p-saturation N^* will be eventually reached. Many such strategies exist and are easily implemented.

For example, net N in Figure 1 admits a p-saturation N^* with 102 transitions that we quickly built with our prototype implementation of a τ-place bisimulation simplifier.

When this step has been performed, it is easy to compute $B_{\tau p}(N^*)$ using Definition 4.7. Our prototype quickly computed

$$B_{\tau p}(N^*) = \{p_0 \equiv p_2; p_1 \equiv p_3 \equiv p_4\}.$$

Then $B_\tau(N)$ has been computed and it only remains to fuse equivalent places in N.

A practical approach

The general approach we just exemplified exactly follows Definition 6.3. It can fail if we consider a net N which is not p-saturable. However it is still possible to simplify such a net using τ-place bisimulation. Today, we can offer two possible approaches:

1. **$B_\tau(N)$ is computable for any N.**

   This is Proposition 3.5. We do not consider this further because it dismisses practical considerations of complexity, ...

2. **It is possible to use an approximation of $B_\tau(N)$.**

   In fact, Theorem 3.3 can be generalized and it is correct to fuse places in N according to any equivalence $B \subseteq B_\tau(N)$. It turns out that correct approximations of $B_\tau(N)$ are easily computed. For example, for any covering sequence $N \rightsquigarrow N_1 \rightsquigarrow \cdots N_k$, $B_{\tau p}(N_k)$ is a correct approximation of $B_\tau(N)$.

   This suggests a simple practical way of dealing with an arbitrary finite net N without checking for p-saturability: Just compute a covering sequence $N \rightsquigarrow \cdots N_k$ without allowing k to reach a given arbitrary bound. If that bound is reached before p-saturation, interrupt the construction of the covering sequence and use $B_{\tau p}(N_k)$ as an approximation.

   This approach is not perfect, but it is practical. We must observe that there exists nets for which the sequence $B_{\tau p}(N_0), B_{\tau p}(N_1), \ldots, B_{\tau p}(N_k), \ldots$ does not converge towards $B_\tau(N_0)$ even if $N_0 \rightsquigarrow N_1 \rightsquigarrow \cdots N_k \cdots$ is fair. Furthermore, it needs not be an increasing sequence. Even the (increasing) sequence

   $$B_{\tau p}(N_0),\ \ B_{\tau p}(N_0) \cup B_{\tau p}(N_1),\ \ \ldots,\ \ \bigcup_{i \leq k} B_{\tau p}(N_i),\ \ \ldots$$

   needs not converge towards $B_\tau(N_0)$. Finally, even though in practice we found it to be a good candidate, from a theoretical point of view, $B_{\tau p}(N_k)$ is not better than many other possible approximations.

Conclusions

We proposed an extension of place bisimulations suited to the simplification of P/T nets labeled with both visible and invisible actions. We gave a simple algorithm computing the largest τ-place bisimulation over a given net N. The simplification it allows are non-trivial and compatible with τ-bisimulation. Therefore they can be used e.g. as a "code reduction" phase in a compiler generating P/T nets from a parallel language, or as a simplification phase in a model-checker for Petri nets, provided that the properties one wants to check are preserved by τ-bisimilarity. On a natural set of examples, τ-place bisimulation typically divides by 2 the number of places of a given net. Note that the complexity of model-checking Petri nets is very high, so that even a slight (correct) reduction of the net under scrutiny is much welcome.

Place bisimulation appears to be a very promising subject. Several questions still need to be investigated:

Complexity issues: The covering algorithm may be costly, as is usually the case with P/T nets. We want to prove that covering sequences of polynomial-length exists for a large class of P/T nets (not just τ-graphs as in Proposition 6.6).

Branching bisimulation: It would be useful to have an algorithm for branching place bisimulation [GW89]. This would yield slightly weaker reductions but typical branching-time temporal logics are compatible with branching bisimulation and not τ-bisimulation.

Marked nets: By essence, place bisimulation considers all possible markings of a net. With marked nets, it would be interesting to restrict one's attention to reachable markings. In principle this could allow better reductions. However, this problem is very tricky as our first experiences show (see section 8.1 in [ABS91]).

Combination of methods: Other reduction methods exist in the literature. From a practical point of view, it would be interesting to know how they combine with place bisimulation. Note that this can be a successful approach to the reduction of marked nets.

Of course, it is necessary to investigate "real-life" situations to see the feasibility and the benefits of our approach in practice. We are conducting such a study for both strong place bisimulation and τ-place bisimulation.

Acknowledgments

Ph. Flajolet introduced us to Mahler's partition problem. H. Garavel generously provided interesting examples. We would also like to thank M. Nielsen, Ph. Darondeau, E.-R. Olderog, R. Devillers, G. Berthelot, S. Haddad, C. Girault and his team and the whole concurrency group here at LIFIA for many helpful remarks, hints, advices, and even questions about place bisimulations.

References

[ABS91] C. Autant, Z. Belmesk, and Ph. Schnoebelen. Strong bisimilarity on nets revisited (extended abstract). In *Proc. PARLE'91, vol. II: Parallel Languages, Eindhoven, LNCS 506*, pages 295–312. Springer-Verlag, June 1991.

[AS92] C. Autant and Ph. Schnoebelen. Place bisimulations in Petri nets. In *Proc. 13th Int. Conf. Application and Theory of Petri Nets, Sheffield, UK, LNCS 616*, pages 45–61. Springer-Verlag, June 1992.

[Ber85] G. Berthelot. Checking properties of nets using transformation. In *Advances in Petri Nets 1985, LNCS 222*, pages 19–40. Springer-Verlag, 1985.

[CPS93] R. Cleaveland, J. Parrow, and B. Steffen. The concurrency workbench: A semantics-based tool for the verification of concurrent systems. *ACM Transactions on Programming Languages and Systems*, 15(1):36–72, January 1993.

[GW89] R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in process algebra. In *Information Processing 89, Proc. IFIP 11th World Computer Congress, San Francisco*, pages 613–618. North-Holland, August 1989.

[IP91] P. Inverardi and C. Priami. Evaluation of tools for the analysis of communicating systems. *EATCS Bull.*, 45:158–187, October 1991.

[Kos82] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proc. 14th ACM Symp. Theory of Computing, San Francisco*, pages 267–281, May 1982.

[KS90] P. C. Kanellakis and S. A. Smolka. CCS expressions, finite state processes and three problems of equivalence. *Information and Computation*, 86:43–68, 1990.

[Old89] E.-R. Olderog. Strong bisimilarity on nets: a new concept for comparing net semantics. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, Noordwijkerhout, LNCS 354*, pages 549–573. Springer-Verlag, 1989.

[Pfi92] W. Pfister. *Simplification sémantique des réseaux de Petri par la bisimulation de places.* Rapport de DEA, Univ. Grenoble, June 1992.

Appendix

A Proof of Proposition 5.5

The proof requires some knowledge of causal nets and processes of P/T nets. See e.g. [Old89, AS92] for definitions.

Proof Consider a process $\pi = (C, f)$ corresponding to a given way of firing the sequence $M = {}^\bullet\sigma \xrightarrow{\sigma} \sigma^\bullet = M'$ (and write $C = (B_C, E_C, F_C)$ for causal nets). We show that there exists another process $\pi' = (C', f')$ in N with

(1) $f'({}^\circ C') = M$,

(2) $f'(C'^\circ) = M'$,

(3) $l_N(f'(E_{C'})) =_\tau l_N(\sigma)$, and

(4) $B_{C'} = {}^\circ C' \cup C'^\circ$.

(4) ensures that π' denotes a concurrent step U in N, (1) and (2) say that $M \xrightarrow{U} M'$ and (3) says that $l_N(U) =_\tau l_N(\sigma)$.

π' is built by modifying π. If π satisfies (4) then it is a concurrent step. If it does not, then there exist $e, e' \in E_C$ and $b \in B_C$ s.t. $b \in (e^\bullet \cap {}^\bullet e')$. Consider the causal net C_1 extracted from C by restricting to conditions $B_{C_1} = {}^\bullet e^\bullet \cup {}^\bullet e'^\bullet$ and events $E_{C_1} = \{e, e'\}$. Write f_1 for f restricted to C_1. The process (C_1, f_1) corresponds to some $M_1 \xrightarrow{t.t'} M_2$ in N. Because N is $2p$-saturated, this can be replaced by some $M_1 \xrightarrow{V} M_2$ with a *concurrent step* process (C_2, f_2). We can choose ${}^\circ C_2 = {}^\circ C_1$ and $C_2{}^\circ = C_1{}^\circ$ so that we can replace (C_1, f_1) in π by (C_2, f_2), yielding another process π' satisfying (1-3) and having (at least) one less condition not in ${}^\circ C \cup C^\circ$. This can be repeated until the resulting process is a concurrent step. $\square$

B Proof of Theorem 7.6

Before proving Theorem 7.6, we first need to establish a few intermediary results.

Proposition B.1 *If a covering sequence $N_0 \rightsquigarrow N_1 \rightsquigarrow \cdots$ is fair, then no $\sigma \in T^\times_{N_k}$ remains not covered in all N_m, $m \geq k$.*

Proof Because every N_i is just N_{i-1} with one more transition, it is possible to consider N_∞ the limit of the increasing sequence. N_∞ may have an infinite number of transitions. Because the covering sequence is fair, N_∞ is $2p$-saturated, and then p-saturated (Proposition 5.5). Then any $\sigma \in T^\times_{N_k}$ is covered in N_∞, implying that it is covered in some N_m. $\square$

Definition B.2 *Given two nets N_1 and N_2 with same set of places, we say that N_1 is* less p-saturated *than N_2, written $N_1 \preceq N_2$, iff*

(1) for all $U_1 \in \mathcal{M}^\times(T_{N_1})$ there is a $U_2 \in \mathcal{M}^\times(T_{N_2})$ s.t. $\xrightarrow{U_1} \subseteq_\tau \xrightarrow{U_2}$, and

(2) for all $\sigma_2 \in T^\times_{N_2}$ there is a $\sigma_1 \in T^\times_{N_1}$ s.t. $\xrightarrow{\sigma_2} \subseteq_\tau \xrightarrow{\sigma_1}$.

The intuition behind this definition is that N_1 can τ-simulate the behaviour of N_2 while N_2 can τp-simulate N_1.

Proposition B.3 *$\preceq$ is a quasi-ordering.*

is obvious from the definition. The name "less p-saturated" is meaningful because of the following two propositions:

Proposition B.4 *If $N \rightsquigarrow N'$ then $N \preceq N'$.*

Proof Assume $N \rightsquigarrow_{t_1,t_2} N'$ with $T_{N'} = T_N + \{t\}$. Because $T_N \subseteq T_{N'}$, **(1)** is satisfied. **(2)** is satisfied because $\xrightarrow{t} =_\tau \xrightarrow{t_1.t_2}$. $\square$

Proposition B.5 *If $N \preceq N'$ and N is p-saturated, then N' is p-saturated.*

Proof Assume σ is a sequence in $T^\times_{N'}$, then there is a σ_1 in $T^\times_N$ with $\xrightarrow{\sigma} \subseteq_\tau \xrightarrow{\sigma_1}$, so that there is U_1 in $\mathcal{M}^\times(T_N)$ with $\xrightarrow{\sigma_1} \subseteq_\tau \xrightarrow{U_1}$ (because N is p-saturated), so that there is a U_2 in $\mathcal{M}^\times(T_{N'})$ with $\xrightarrow{U_1} \subseteq_\tau \xrightarrow{U_2}$. Finally $\xrightarrow{\sigma} \subseteq_\tau \xrightarrow{U_2}$ and then σ is covered. $\square$

Proposition B.6 *If $N \preceq N' \preceq N$ then $Id_{P_N} : N \sim_{\tau p} N'$.*

is obvious.

Now we can prove Theorem 7.6.

Proof Assume N is p-saturable. Then there exists a saturating sequence $N = N_0 \rightsquigarrow N_1 \rightsquigarrow N_2 \cdots \rightsquigarrow N_k = N^*$. Assume $N = N'_0 \rightsquigarrow N'_1 \cdots$ is another covering sequence. We show that if this sequence is fair, then it eventually reaches a p-saturated net.

We prove by induction that there is a sub-sequence $N'_{i_0}, N'_{i_1}, \ldots, N'_{i_k}$ s.t. $N_j \preceq N'_{i_j}$ for $j = 0, \ldots, k$. This is shown by induction. The base case is done by picking $i_0 = 0$. Then assume $N_j \preceq N'_{i_j}$ and $N_j \rightsquigarrow_{t.t'} N_{j+1}$, so that $T_{N_{j+1}} = T_{N_j} + \{u\}$ with $\xrightarrow{t.t'} =_\tau \xrightarrow{u}$. Because $N_j \preceq N'_{i_j}$, there is a U and a U' in $\mathcal{M}^\times(T_{N'_{i_j}})$ s.t. $\xrightarrow{t} \subseteq_\tau \xrightarrow{U}$ and $\xrightarrow{t'} \subseteq_\tau \xrightarrow{U'}$. Let σ be a given sequentialization of $U.U'$, so that $\xrightarrow{U.U'} \subseteq_\tau \xrightarrow{\sigma}$. Proposition B.1 implies that σ is covered (say by V) in some N'_m (pick $m \geq i_j$). We verify that $N_{j+1} \preceq N'_m$ so that letting $i_{j+1} \stackrel{\text{def}}{=} m$ will complete the induction step.

To do this, consider any $U \in \mathcal{M}^\times(T_{N_{j+1}})$. U has the form $U_0 + a.\{u\}$ with $U_0 \in \mathcal{M}^\times(T_{N_j})$ and $a \in I\!N$. Because $N_j \preceq N'_{i_j}$, there is a $V_0 \in \mathcal{M}^\times(T_{N'_{i_j}})$ s.t. $\xrightarrow{U_0} \subseteq_\tau \xrightarrow{V_0}$. We have seen that

$$\xrightarrow{u} =_\tau \xrightarrow{t.t'} \subseteq_\tau \xrightarrow{U.U'} \subseteq_\tau \xrightarrow{\sigma} \subseteq_\tau \xrightarrow{V}$$

so that $\xrightarrow{U} \subseteq_\tau \xrightarrow{V_0 + a.V}$ with $V_0 + a.V \in \mathcal{M}^\times(T_{N'_{i_{j+1}}})$. Finally, $N_{j+1} \preceq N'_{i_{j+1}}$.

There only remains to observe that $N_k \preceq N'_{i_k}$ implies that N'_{i_k} is p-saturated (Proposition B.5) to conclude the proof. $\square$

C Proof of Theorem 7.4

The proof uses a reasoning similar to what we did for Theorem 7.6.

Proof Assume N^* is a finite p-saturation of N. Then N^* is N plus a finite set $\{u_1, \ldots, u_k\}$ of transitions. Write N_i for $N + \{u_1, \ldots, u_i\}$. Clearly $N_i \preceq N_{i+1}$ for $i < k$. Assume N admits no saturating sequence. Then there exists a fair infinite covering sequence $N = N'_0 \rightsquigarrow N'_1 \rightsquigarrow \cdots$. We shall extract a subsequence $N'_{i_0}, \ldots, N'_{i_k}$ with $N_j \preceq N'_{i_j}$ for $j = 0, \ldots, k$, so that $N^* = N_k \preceq N'_{i_k}$ implies that N'_{i_k} is p-saturated, a contradiction.

The subsequence is built inductively. Pick $i_0 = 0$. Now assume we have $N_{j-1} \preceq N'_{i_{j-1}}$ for some $0 < j \leq k$. N_j is N_{j-1} plus transition u_j. Because $N \preceq N_j$, $\xrightarrow{u_j} \subseteq_T \xrightarrow{\sigma_j}$ for some $\sigma_j \in T_N^\times$. Because we chose a fair saturation sequence, σ_j is covered at some stage m in the sequence. Pick $i_j \geq m$ (and $i_j \geq i_{j-1}$). This ensures $N_j \preceq N'_{i_j}$. $\square$