Parallel Processing Letters
© World Scientific Publishing Company

New Efficient Petri Nets Reductions for Parallel Programs Verification

Serge HADDAD
LAMSADE-CNRS UMR 7024 Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, FRANCE

and

Jean-François PRADAT-PEYRE
CEDRIC EA 1395, Conservatoire National des Arts et Métiers
292 rue Saint Martin, 75141 Paris Cedex 03, FRANCE

Received April 2005
Revised January 2006
Communicated by J. Beauquier

ABSTRACT

Structural model abstraction is a powerful technique for reducing the complexity of a state based enumeration analysis. We present in this paper new efficient Petri nets reductions. First, we define "behavioural" reductions (i.e. based on conditions related to the language of the net) which preserve a fundamental property of a net (i.e. liveness) and any formula of the (action-based) linear time logic that does not observe reduced transitions of the net. We show how to replace these conditions by structural or algebraical ones leading to reductions that can be efficiently checked and applied whereas enlarging the application spectrum of the previous reductions. At last, we illustrate our method on a significant and typical example of a synchronisation pattern of parallel programs.

Keywords: Reduction theory, structural abstraction, concurrent software verification, Petri nets.

1. Introduction

It is currently admitted that the use of formal methods is essential to obtain less error-prone parallel programs. Given a parallel program and a property (or a model of the program and of the property) the verification process proceeds either by a state enumeration or by applying structural algorithms. In case of finite state systems, the former ones lead to a complete verification but the analysis is restricted by the inherent combinatorial explosion factor. Moreover it does not give insight on how to identify and correct the faulty parts of the program. Structural methods do not generally ensure the complete correctness of the modelled system. However they are efficient and they produce results that allow practitioners to make pertinent modifications.

Thus, an attractive trade-off would be to first perform structural abstractions in order to obtain a simplified model on which an enumeration based method can more easily be applied. Usually, the model may be abstracted in two ways [CGL94]. Data abstraction maps the range of a variable to a smaller domain and propagates this transformation on the control flow. Operation abstraction merges consecutive instructions into a virtual atomic one whose effect is the composition of the effects of these instructions. In this paper, we will focus on the latter abstraction. The main advantage of such a transformation is the drastic reduction of the combinatorial explosion due to the elimination of the intermediate states.

The first works in this area were performed by Lipton in [Lip75] which developed a reduction theory aiming at preserving the deadlock property. These works have been extended by Doeppner, Schneider, Cohen and Lamport for different formalisms or different properties [Doe77,LS89,Gri96,CL98]. More recently, Freund, Qadeer and Flanagan [FQ03b,FQ03c,FQ03a] leveraged the Lipton's theory to detect transactions in multithreaded programs (and consider these transactions as atomic in the verification process). The main drawbacks of these different approaches are the difficulty to detect conditions allowing to apply the reduction and, when directly defined at a programming language level, an incomplete theoretical justification. Thus, we have chosen to develop abstractions for a low-level model with a formal semantics: the Petri nets. The advantages of this approach are twofold:

- Due to the formal semantics, the set of properties to be preserved can be easily expressed with some temporal logic and the preservation of this set by an abstraction is fully proved.
- Dealing with a low level model leads to abstractions useful for a wide range of applications and can be straightforwardly adapted or specialised for a target high-level model.

Our work on these abstractions is an important generalisation of two reductions proposed by Berthelot in [Ber83,Ber85,Ber86], the pre- and the post-agglomeration, which merge sequential transitions into an atomic one. Indeed, original application conditions only rely on local structural patterns. Thus, the time complexity application is linear w.r.t. the size of the Petri net. Nevertheless, since the conditions are purely local, they are quite restrictive and lead to a limited range of possible applications and are unable to reduce real parallel program synchronisation patterns. Esparza and Schröter [ES01] tried to enhance the application area of one of these reductions by simplifying one point in the original pre-agglomeration conditions. However, they consider only 1-safe Petri nets (the marking of each place is bounded by 1), the application conditions remain purely structural, and as the authors focus only on infinite sequence preservation, their reductions do not even preserve the deadlock property!

We proceed here in a different way: first, we characterise a set of behavioural conditions that cover a large class of parallel program patterns and that ensure the preservation of the considered properties. Secondly, we give structural sufficient conditions for the behavioural ones. In order to obtain the less restrictive possible conditions while keeping the possibility to check them easily and efficiently, we include algebraic constraints in our structural conditions. These ones are based on the description of linear programs including the linear invariants of the net for which efficient computations are known [CS91]. Such constraints express restrictions on the global behaviour of the net which were not taken into account by the previous reductions.

Our new transition agglomerations are presented in Section 2. We state that they preserve a large class of properties and we prove one of the two main theorems (complete proofs are provided in the report [HPP04]). In Section 3, we show how to obtain structural and algebraical conditions and we demonstrate on a typical example of a parallel program synchronisation pattern that our results considerably enhance previous ones.

2. Petri nets behavioural agglomerations

A Petri net reduction is characterised by some application conditions, by a net transformation and by a set of preserved properties (i.e. which properties are simultaneously true or false in the original net and in the reduced one). Before developing these three points for our reductions we recall some definitions.

2.1. Petri net definitions

Definition 1 (Petri net model) *A marked net (N, m_0) is defined by a tuple (P, T, W^-, W^+, m_0) where: P is the finite set of places, T is the finite set of transitions disjoint from P, W^- (resp. W^+) an integer matrix indexed by $P \times T$ is the backward (resp. forward) incidence matrix, m_0 an integer vector indexed by P is the initial marking. The transitions linked to a place p are defined by $^\bullet p = \{t | W^+(p, t) > 0\}$ and $p^\bullet = \{t | W^-(p, t) > 0\}$.*

Definition 2 (Firing rule) *Let (N, m_0) be a marked net then a transition $t \in T$ is firable from a marking m (denoted by $m[t\rangle$) iff $\forall p \in P\ m(p) \geq W^-(p, t)$. The firing of $t \in T$ firable from m leads to the marking m' (denoted by $m[t\rangle m'$) defined by $\forall p \in P\ m'(p) = m(p) + W(p, t)$ where W the incidence matrix is defined by $W = W^+ - W^-$.*

We use the following notations.

- T^* is the set of finite sequences of transitions and T^ω is the set of infinite sequences of transitions; λ defines the empty sequence of transitions;
- If s is a finite sequence of transitions, $|s|$ denotes the length of s (that is recursively defined by $|\lambda| = 0$ and $|s.t| = |s| + 1$);
- $\Pi_{T'}(s)$ denotes the projection of the sequence s on a subset of transitions T' and is recursively defined by $\Pi_{T'}(\lambda) = \lambda$, $\forall t \in T'$, $\Pi_{T'}(s.t) = \Pi_{T'}(s).t$ and $\forall t \notin T'$, $\Pi_{T'}(s.t) = \Pi_{T'}(s)$;
- $|s|_{T'} = |\Pi_{T'}(s)|$ denotes the number of occurrences of transitions of T' in s;
- $Pref(s) = \{s' \mid \exists s''$ s.t. $s = s'.s''\}$ denotes the set of prefixes of s.

Definition 3 (Firing rule extension) *Let (N, m_0) be a marked net. A finite sequence $s \in T^*$ is firable from a marking m and leads to m' (also denoted by $m[s\rangle$ and $m[s\rangle m'$) iff either $s = \lambda$ and $m' = m$ or $s = s_1.t$ with $t \in T$ and $\exists m_1\ m[s_1\rangle m_1$ and $m_1[t\rangle m'$ We note $Reach(N, m_0) = \{m | \exists s \in T^*\ m_0[s\rangle m\}$ the set of reachable markings. An infinite sequence $s \in T^\omega$ is firable from a marking m (also denoted $m[s\rangle$) iff for every finite prefix s_1 of s, $m[s_1\rangle$.*

Proposition 1 *The incidence matrices W, W^- and W^+ can be extended to matrices indexed by $P \times T^*$ by the following recursive definition:*

- $W(p,\lambda) = W^-(p,\lambda) = W^+(p,\lambda) = 0$
- $W(p, s_1.t) = W(p, s_1) + W(p, t)$
- $W^-(p, s_1.t) = Max(W^-(p, s_1), W^-(p, t) - W(p, s_1))$
- $W^+(p, s_1.t) = W(p, s_1.t) + W^-(p, s_1.t)$

such that this extension is equivalent with the firing rule of a sequence, i.e. $\forall s \in T^*,\ m[s\rangle m' \iff \forall p \in P,\ m(p) \geq W^-(p,s)$ *and* $m'(p) = m(p) + W(p,s)$

Definition 4 (Basic Petri net properties) *A marked Petri net (N, m_0) is* **live** *iff $\forall m \in Reach(N, m_0)\, \forall t \in T\, \exists s \in T^*\ m[s.t\rangle$. A marking m is a* **dead marking** *if $\forall t \in T\ NOT(m[t\rangle)$.*

Definition 5 (Generated language) *Let (N, m_0) be a marked net then*

- $L(N, m_0) = \{s \in T^* | m_0[s\rangle\}$ *is the language of finite sequences,*
- $L^{Max}(N, m_0) = \{s \in T^* | m_0[s\rangle m, m\, a\, dead\, marking\}$ *is the language of finite maximal sequences,*
- $L^\omega(N, m_0) = \{s \in T^\omega | m_0[s\rangle\}$ *is the language of infinite sequences.*

2.2. Agglomeration scheme

We suppose in the sequel that the set of transitions of the net is partitioned as: $T = T_0 \biguplus_{i \in I} H_i \biguplus_{i \in I} F_i$ where I denotes a non empty set of indices, and $\biguplus$ the disjoint union. The underlying idea of this decomposition is that a pair (H_i, F_i) defines transitions sets that are causally dependent: an occurrence of $f \in F_i$ in a firing sequence may always be related to a previous occurrence of some $h \in H_i$ in this sequence. Starting from this property, we develop conditions on the behaviour of the net which ensure that we can restrict the dynamics of the net to sequences where each occurrence $h \in H_i$ is immediately followed by an occurrence of some $f \in F_i$ without changing its behaviour w.r.t. a set of properties.

Definition 6 (Reduced net) *The reduced Petri net (N_r, m_0) is defined by*

- $P_r = P$, $T_r = T_0 \cup_{i \in I} (H_i \times F_i)$. *We note hf the transition $(h, f) \in H_i \times F_i$*
- $\forall t_r \in T_0, \forall p \in P_r$, $W_r^-(p,t) = W^-(p,t)$ *and* $W_r^+(p,t) = W^+(p,t)$
- $\forall i \in I, \forall hf \in H_i \times F_i, \forall p \in P_r$ $W_r^-(p, hf) = W^-(p, h.f)$ *and* $W_r^+(p, hf) = W^+(p, h.f)$

From now, we note $H = \cup_{i \in I} H_i$ and $F = \cup_{i \in I} F_i$. The firing rule in the reduced net is noted $\rangle_r$ (i.e. $m[s\rangle_r m'$ denotes a firing sequence in the reduced net).

As we want to compare the behaviour of the reduced and the original nets and as the sets of transitions are not identical we introduce the following one to one homomorphism which allows such a comparison.

Definition 7 *We note ϕ the homomorphism from the monoid T_r^* to the monoid T^* defined by: $\forall t \in T_0, \phi(t) = t$ and $\forall i \in I, \forall h \in H_i, \forall f \in F_i, \phi(hf) = h.f$. This homomorphism is extended to an homomorphism from $\mathcal{P}(T_r^*)$ to $\mathcal{P}(T^*)$ and from $\mathcal{P}(T_r^\omega)$ to $\mathcal{P}(T^\omega)$.*

The next basic proposition (which proof is straightforward from proposition 1) states in a formal way that the behaviour of the reduced net is a subset of the original behaviour.

Proposition 2 *Let* (N, m_0) *be a marked Petri net. Then:*

1 $\forall s_r \in T_r^*, m[s_r\rangle_r m' \iff m[\phi(s_r)\rangle m'$
2 $\forall s_r \in T_r^\omega, m[s_r\rangle_r \iff m[\phi(s_r)\rangle$

At this point, we have proved that if a maximal or infinite sequence violates a sequence property in the reduced net then the property is also violated in the original one. However, some original sequences that highlight problems may disappear in the reduced net and we have no result regarding the Petri net liveness property (the reduced net may be live while the original is not and vice versa). So we need to formalise the dependence of F_i on H_i. As we consider an abstraction that merges transitions of H_i with transitions of F_i it seems reasonable to impose that, in all sequences of the original net, a transition of F_i must always be preceded by a transition of H_i. We introduce this constraint with the help of a set of counting functions, denoted Γ_i.

Definition 8 (Potential agglomerability) (N, m_0) *is potentially agglomerable (p-agglomerable for short) iff* $\forall s \in L(N, m_0)$*,* $\forall i \in I$*,* $|s|_{H_i} - |s|_{F_i} \geq 0$*. We will denote in the following* Γ_i *the mapping from* T^* *to* $I\!N$ *defined by* $\Gamma_i(s) = |s|_{H_i} - |s|_{F_i}$

This behavioural hypothesis can easily be ensured by the following structural sufficient condition: $\forall i \in I$, $\exists p_i$ such that $m_0(p_i) = 0$, ${}^\bullet p_i = H_i$, $p_i{}^\bullet = F_i$, and $\forall h \in H_i, \forall f \in F_i$, $W^+(p_i, h) = W^-(p_i, f) = 1$.

In the following, we study p-agglomerable nets. The remainder of the section is devoted to the presentation of two sets of conditions that ensure the equivalence between the behaviours of the original and the reduced net. Informally stated, the **pre-agglomeration** scheme expresses the fact that firing the transitions of H_i is only useful for firing the transitions of F_i whereas the **post-agglomeration** scheme expresses the fact that the firing of transitions of F_i are mainly conditioned by the firing of the transitions of H_i.

2.3. Behavioural post-agglomeration

In this section, we restrict I to a singleton (i.e. $I = \{1\}$ and we set $\Gamma = \Gamma_1$, $H = H_1$ and $F = F_1$). The main property that the conditions of the post-agglomeration implies is the following one: in every firing sequence with an occurrence of a transition h of H followed later by an occurrence of a transition f of F, one can immediately fire f after h. From a modelling point of view, the set F represents local actions while the set H corresponds to global actions possibly involving synchronisation.

Definition 9 *Let* (N, m_0) *be a p-agglomerable marked net.* (N, m_0) *is:*

1. HF***-interchangeable*** *iff one of these two conditions is fulfilled:*

   1 $\forall m \in Reach(N, m_0)$*,* $\forall h, h' \in H$*,* $\forall f \in F$*,* $m[h.f\rangle \iff m[h'.f\rangle$

2 $\forall m \in Reach(N, m_0)$, $\forall h \in H$, $\forall f, f' \in F$, $m[h.f\rangle \iff m[h.f'\rangle$

2. *F-**independent** iff* $\forall h \in H$, $\forall f \in F$, $\forall s \in (T_0 \cup H)^*$, $\forall m \in Reach(N, m_0)$, $m[h.s.f\rangle \implies m[h.f.s\rangle$
*and **strongly** F-**independent** iff* $\forall h \in H$, $\forall f \in F$, $\forall s \in T^*$ *s.t.* $\forall s' \in Pref(s), \Gamma(s') \geq 0$ $\forall m \in Reach(N, m_0)$, $m[h.s.f\rangle \implies m[h.f.s\rangle$

3. *F-**continuable** iff* $\forall h \in H$, $\forall s \in T^*$, *s.t.* $\forall s' \in Pref(s), \Gamma(s') \geq 0$ $\forall m \in Reach(N, m_0)$ $m[h.s\rangle \implies \exists f \in F$ *s.t.* $m[h.s.f\rangle$

We express the dependence of the set F on the set H with three hypotheses. We first notice that, in the original net, the transitions $h \in H_i$ and $f \in F_i$ may be live whilst the sequence $h.f$ is not live. Thus the HF-interchangeability condition forbids this behaviour. The F-independence means that any firing of $f \in F$ may be anticipated just after the occurrence of a transition $h \in H$ which enables this firing. The F-continuation means that an excess of occurrences of $h \in H$ can always be reduced by subsequent firings of transitions of F.

The following theorem expresses which properties are preserved by the post-agglomeration giving in each case the required conditions. The first point is related with maximal sequences and allows for instance the modeller to look for deadlocks. The second point is related to infinite sequences which characterise, for instance, fairness properties. More generally these two points allow to check any (action-based) linear time logic that does not observe transitions of F. The third point is related to the Petri net liveness as liveness cannot be specified with linear time logic. Note that we have proven in [HPP04] that this reduction is a strict extension of the post-agglomeration of Berthelot [Ber83].

Theorem 1 *Let (N, m_0) be a p-agglomerable Petri net. If furthermore*

1 (N, m_0) *is F-continuable then*

$$\Pi_{T_0 \cup H}(L^{max}(N, m_0)) = \Pi_{T_0 \cup H}(\phi(L^{max}(N_r, m_0)))$$

2 (N, m_0) *is F-continuable and strongly F-independent then*

$$\Pi_{T_0 \cup H}(L^{\omega}(N, m_0)) = \Pi_{T_0 \cup H}(\phi(L^{\omega}(N_r, m_0)))$$

3 (N, m_0) *is F-continuable, F-independent and HF-interchangeable then*

$$(N, m_0) \text{ is live} \iff (N_r, m_0) \text{ is live}$$

The remainder of the section is devoted to the proof of this theorem. First we remark that among sequences of the original net, some of them look like sequences of the reduced net; we call them simulateable (they can be directly simulated in the reduced net).

Definition 10 (Simulateable sequence) *A sequence $s \in T^*$ (resp. T^{ω}) is said to be* **simulateable** *if there exists a decomposition of s:*
$s = \phi(s_1).s'_1.\phi(s_2).s'_2 \ldots \phi(s_n).s'_n$ *(resp.* $s = \phi(s_1).s'_1.\phi(s_2).s'_2 \ldots \phi(s_n).s'_n \ldots$ *) with* $\forall m$, $s_m \in H \times F$ *and* $s'_m \in T_0^*$.

Remark 1 *A sequence s is simulateable iff there exists s_r such that $s = \phi(s_r)$. Since s_r is unique, s_r is denoted by $\phi^{-1}(s)$.*

We now prove that finite and infinite sequences can be re-ordered into simulateable sequences while preserving the projection of these sequences on $T_0 \cup H$.

Proposition 3 (F^*-independence) *Let (N, m_0) be a p-agglomerable net which is F-independent. Then $\forall s \in T^*$ such that $m_0[s_0\rangle m[s\rangle m'$ with $\forall s'$ prefix of s $\Gamma(s') \geq 0$, there exists a permutation of s, $\widehat{s} = s_{\bowtie}.s_{\lhd}$, such that:*

1 $m[\widehat{s}\rangle m'$ and $\Pi_{T_0 \cup H}(\widehat{s}) = \Pi_{T_0 \cup H}(s)$
2 $\Pi_F(s_{\lhd}) = \lambda$ and $s_{\bowtie}$ is simulateable.

Furthermore if $\Gamma(s) = 0$ then $s_{\lhd} = \lambda$. We will denote by $\widehat{s} = s_{\bowtie}.s_{\lhd}$ any sequence fulfilling the above requirements with respect to s.

Proof. We prove by induction on the length of $|s|_F$ that there exists at least one sequence $\widehat{s}$.

- $|s|_F = 0$: The decomposition of s, $\widehat{s} = \lambda.s$, fulfils the conditions.
- $|s|_F > 0$: As by hypothesis $\forall s_p \in Pref(s)$, $\Gamma(s_p) \geq 0$ the sequence s can be written $s = s'.h.s_1.f.s_2$ with $s' \in (T_0)^*$, $h \in H$, $s_1 \in (T_0 \cup H)^*$ and $f \in F$. Since the net is F-independent, $m[s'.h.f.s_1.s_2\rangle m'$.
  By construction $s'.h.f$ is a balanced sequence (i.e. $\Gamma(s'.h.f) = 0$). Furthermore a straightforward checking shows that the prefixes of the sequence $s_1.s_2$ fulfil the hypothesis of the proposition.
  Thus by induction $\widehat{s_1.s_2}$ exists and $(s'.h.f.(s_1.s_2)_{\bowtie}).(s_1.s_2)_{\lhd}$ fulfils the conditions of the proposition w.r.t. s.

When $\Gamma(s) = 0$, $\Gamma(s_{\lhd}) = 0$ and since $|s_{\lhd}|_F = 0$, one has also $|s_{\lhd}|_H = 0$ which means that $s_{\lhd} \in (T_0)^*$ and can be concatenated to $s_{\bowtie}$ to obtain a simulateable sequence.

Proposition 4 (F^ω-independence) *Let (N, m_0) be a p-agglomerable net which is F-independent. Then for any infinite sequence $s \in L^\omega(N, m_0)$ there exists a permutation of s, $\widehat{s}$ such that*

1 $\forall s' \in Pref(\widehat{s})$, $m_0[s'\rangle$ and $\Pi_{T_0 \cup H}(\widehat{s}) = \Pi_{T_0 \cup H}(s)$
2 $\exists (s^i_{\bowtie})_{i \geq 0}$ an infinite sequence of simulateable sequences such that: $\widehat{s} = s^1_{\bowtie}.s^1_{\lhd}.s^2_{\bowtie}.s^2_{\lhd} \ldots s^k_{\bowtie}.s^k_{\lhd} \ldots$ with $s^n_{\lhd} \in H^*$

Proof. In order to prove this proposition we need to decompose infinite sequences according to their ultimate behaviour w.r.t. the Γ function. So, we introduce a new notation to characterise this behaviour. The *degree of a sequence* $s \in T^\omega$ $(s = t_1 \ldots t_n \ldots)$ of a p-agglomerable net (denoted by $d^\circ(s)$) is defined by:

$$d^\circ(s) = \liminf_{k \to \infty}(\Gamma(t_1 \ldots t_k)) \overset{def}{=} \lim_{k \to \infty} min\{\Gamma(t_1 \ldots t_{k'}) \mid k' \geq k\}$$

Now we distinguish two cases:

1 $d^\circ(s) = d$. Thus there is the following decomposition of s.
$s = s_1.h_1.s_2.h_2.s_3.\ldots.h_d.s_{d+1}.s_{d+2}.s_{d+3}\ldots$ with $\forall k \geq 0$, $\Gamma(s_k) = 0$ and $\forall s'$ prefix of s_k, $\Gamma(s') \geq 0$ and $\{h_1,\ldots,h_d\} \subset H$. We apply the previous proposition on every s_k obtaining a permutation $\widehat{s_k} = s^k_{\bowtie}$ (remember that $\Gamma(s_k) = 0$). This leads to the following infinite firing sequence:
$s = s^1_{\bowtie}.h_1.s^2_{\bowtie}.h_2.s^3_{\bowtie}.\ldots.h_d.s^{d+1}_{\bowtie}.s^{d+2}_{\bowtie}.s^{d+3}_{\bowtie}\ldots$ This is the kind of sequence we search for.

2 $d^\circ(s) = \infty$. Thus there is the following decomposition of s.
$s = s_1.h_1.s_2.h_2.s_3.\ldots.h_k.s_{k+1}.\ldots$ with $\forall k \geq 0$, $\Gamma(s_k) = 0$ and $\forall s'$ prefix of s_k, $\Gamma(s') \geq 0$ and $\{h_1,\ldots,h_k,\ldots\} \subset H$. So the proof of this case is similar to the one developed for the first case.

We establish now the results claimed in theorem 1.

Lemma 1 *Let (N, m_0) be a p-agglomerable net which is F-independent and HF-interchangeable. Then (N, m_0) live $\Longrightarrow (N_r, m_0)$ live*

Proof. Let $m_0[s_r\rangle_r m$ and $t_r \in T_r$. We have $m_0[\phi(s_r)\rangle m$ and we distinguish three cases.

1 Let $t_r \in T_0$. Since (N, m_0) is live, there exists s_1 such that $m[s_1.t_r\rangle m'$. Since $\Gamma(\phi(s_r)) = 0$ and the net is p-agglomerable, for all prefixes s' of s_1, $\Gamma(s') \geq 0$ holds. Let us pick some s_1 minimising $\Gamma(s_1)$ and suppose that $\Gamma(s_1) > 0$. Then $\widehat{s_1} = s^1_{\bowtie}.h.s'$ with $h \in H$ and $s' \in (T_0 \cup H)^*$. Since the net is live, there is a (shortest) sequence ended by a transition of F, $m'[s''.f\rangle$ with $f \in F$ and $s'' \in (T_0 \cup H)^*$. Thus $s'.t_r.s'' \in (T_0 \cup H)^*$. We apply the F-independence transformation leading to the firing sequence $m[s^1_{\bowtie}.h.f.s'.t_r.s''\rangle$ and $\Gamma(s_1) > \Gamma(s^1_{\bowtie}.h.f.s')$. So necessarily $\Gamma(s_1) = 0$.
We now substitute s_1 by its permutation $s^1_{\bowtie}$ leading finally in the reduced net to the firing sequence $m[\phi^{-1}(s^1_{\bowtie}).t_r\rangle$.

2 Let $t_r = hf$ with $h \in H$ and $f \in F$ and suppose that the HF-interchangeability is fulfilled due to the assertion 1 of this hypothesis. Since (N, m_0) is live there exists a sequence s such that $m[s.f\rangle$. Since $\Gamma(\phi(s_r)) = 0$ and the net is p-agglomerable, $\Gamma(s') \geq 0$ holds for all prefixes s' of s. Thus there is a permutation of s, $\widehat{s} = s_{\bowtie}.s_{\lhd}$ with f occurring in $s_{\bowtie}$, i.e. $s_{\bowtie} = s_1.h'.f.s_2$, $h' \in H$, s_1 and s_2 being simulateable. Due to the assertion 1 of the HF-interchangeability one substitutes h to h' leading to $m[s_1.h.f\rangle$. Thus in the reduced net, $m[\phi^{-1}(s_1).t_r\rangle$.

3 Let $t_r = hf$ with $h \in H$ and $f \in F$ and suppose that the HF-interchangeability is fulfilled due to the assertion 2 of this hypothesis. Since (N, m_0) is live, there exists s_1 such that $m[s_1.h\rangle m'$. Since $\Gamma(\phi(s_r)) = 0$ and the net is agglomerable one has for all prefixes s' of s_1, $\Gamma(s') \geq 0$. Let us pick some s_1 minimising $\Gamma(s_1)$. Similarly to the first point of this proof, $\Gamma(s_1) = 0$. We now substitute s_1 by its permutation $s^1_{\bowtie}$ i.e. leading finally in the reduced net to the firing sequence $m[s^1_{\bowtie}.h\rangle m'$. Using again the liveness, there is a (shortest) sequence ended by a transition of F, $m'[s''.f'\rangle$ with $f' \in F$ and $s'' \in (T_0 \cup H)^*$. We apply the F-independence transformation leading to the firing sequence $m[s^1_{\bowtie}.h.f'\rangle$. Due to the assertion 2 of the HF-interchangeability one substitutes f to f' leading to $m[s^1_{\bowtie}.h.f\rangle$. Thus in the reduced net, $m[\phi^{-1}(s^1_{\bowtie}).hf\rangle$.

Lemma 2 *Let (N, m_0) be a p-agglomerable net which is F-independent and F-continuable. Then (N_r, m_0) live $\Longrightarrow$$(N, m_0)$ live*

Proof. Let $m_0[s\rangle m$ and let $t \in T$. We prove that t is necessarily fireable from m by induction on $\Gamma(s)$.

- $\Gamma(s) = 0$: Let us define $t_r \in T_r$ by $t_r = t$ if $t \in T_0$, $t_r = hf$ if $t = h$ with some $f \in F$ and $t_r = hf$ if $t = f$ with some $h \in H$. Since $\Gamma(s) = 0$, there is a permutation of s, $s_{\bowtie}$ s.t. $m_0[s_{\bowtie}\rangle m$. Then $m_0[\phi^{-1}(s_{\bowtie})\rangle_r m$. Since (N_r, m_0) is live, there exists s_r such that $m[s_r.t_r\rangle_r$. Thus $m[\phi(s_r).\phi(t_r)\rangle$ and by construction t occurs in $\phi(t_r)$.
- $\Gamma(s) > 0$: then the permutation of s can written as $\widehat{s} = s_{\bowtie}.s_1.h.s_2$ with $h \in H$, and $|s_2|_F = 0$. Since the original net is F-continuable, there exists $f \in F$ such that $m_0[s_{\bowtie}.s1.h.s2\rangle m[f\rangle m'$. Since $\Gamma(s_{\bowtie}.s1.h.s2.f) = \Gamma(s) - 1$, t is necessarily fireable from m' (and thus from m).

Lemma 3 *Let (N, m_0) be a p-agglomerable net which is F-continuable. Then*

$$\Pi_{T_0 \cup H}(L^{max}(N, m_0)) = \Pi_{T_0 \cup H}(\phi(L^{max}(N_r, m_0)))$$

Proof.

1 $\Pi_{T_0 \cup H}(L^{max}(N, m_0)) \subseteq \Pi_{T_0 \cup H}(\phi(L^{max}(N_r, m_0)))$
Let s be a sequence such that $m_0[s\rangle m_d$ with m_d a dead marking. The continuation hypothesis implies that s is a balanced sequence. So $m_0[\phi^{-1}(\widehat{s})\rangle_r m_d$. Since any sequence $m_d[s_r\rangle_r$ leads to a sequence $m_d[\phi(s_r)\rangle$, m_d is dead in the reduced net.

2 $\Pi_{T_0 \cup H}(L^{max}(N, m_0)) \supseteq \Pi_{T_0 \cup H}(\phi(L^{max}(N_r, m_0)))$
Let s_r be a sequence such that $m_0[s_r\rangle_r m_d$ and m_d a dead marking (of the reduced net). We know that $m_0[\phi(s_r)\rangle m_d$. It remains to prove that m_d is a dead marking of the original net. Let us suppose that t is a fireable transition from m_d ($m_d[t\rangle$). As $\phi(s_r)$ is a balanced sequence, $t \notin F$. Furthermore, $t \notin T_0$; otherwise $m_d[t\rangle_r$ which contradicts the fact that m_d is a dead marking in the reduced net. So $t = h \in H$. The continuation hypothesis implies that $\exists f \in F$ such that $m_d[h.f\rangle$. Hence $m_d[\phi^{-1}(h.f)\rangle_r$ with the same contradiction.

Lemma 4 *Let (N, m_0) be a p-agglomerable net which is strongly F-independent and F-continuable. Then*

$$\Pi_{T_0 \cup H}(L^{\omega}(N, m_0)) = \Pi_{T_0 \cup H}(\phi(L^{\omega}(N_r, m_0)))$$

Proof. We prove that $\Pi_{T_0 \cup H}(L^{\omega}(N, m_0)) \subseteq \Pi_{T_0 \cup H}(\phi(L^{\omega}(N_r, m_0)))$ (the other inclusion is a direct consequence of proposition 2).

Let $s \in L^{\omega}(N, m_0)$. We only consider here the case where $d(s) = n$ is finite (a same reasoning can be performed in the infinite case). Due to proposition 4, there exists a sequence $\widehat{s}$ such that $\forall s' \in Pref(\widehat{s})$, $m_0[s'\rangle$, $\Pi_{T_0 \cup H}(\widehat{s}) = \Pi_{T_0 \cup H}(s)$, $\exists(\sigma^i_{\bowtie})_{i \geq 0}$ an infinite sequence of simulateable sequences such that: $\widehat{s} = \sigma^1_{\bowtie}.\sigma^1_{\triangleleft}.\sigma^2_{\bowtie}.\sigma^2_{\triangleleft} \ldots \sigma^k_{\bowtie}.\sigma^k_{\triangleleft} \ldots$ with $\sigma^n_{\triangleleft} \in H^*$. So, by (possible) insertions of empty sequences, $\widehat{s}$ may be rewritten as: $\widehat{s} = s^1_{\bowtie}.h_1.s^2_{\bowtie}.h_2 \ldots s^n_{\bowtie}.h_n.s^{n+1}_{\bowtie}$ with $\forall i, h_i \in H$ and $s^i_{\bowtie}$ a simulateable sequence.

We build then by induction a simulateable infinite sequence s' satisfying $\Pi_{T_0 \cup H}(s') = \Pi_{T_0 \cup H}(s)$. The induction hypothesis is the following one: there exists an infinite fireable sequence s_k for $k \leq n$ obtained from $\widehat{s}$ by inserting immediately after each of the $h_i, (i = 1..k)$ a transition of F. The basic case is handled by considering $s_0 = \widehat{s}$.

Now let us look at h_{k+1} : $s_k{=}s'_k.h_{k+1}.s''_k$ with s'_k a balanced sequence and s''_k an infinite suffix of $\widehat{s}$. Thus there exists a transition f_{s_p} such that $s'_k.h_{k+1}.s_p.f_{s_p}$ is a firing sequence (due to the F-continuation hypothesis). Let us pick a transition f which occurs infinitely often in $\{f_{s_p}\}$. Then (due to the strong F-independence hypothesis) $s'_k.h_{k+1}.f.s''_k$ is an infinite firing sequence and the induction step is verified. So s_n is the balanced sequence we look for.

2.4. Behavioural pre-agglomeration

We state now four conditions which "roughly speaking" ensure that delaying the firing of a transition $h \in H_i$ until some $f \in F_i$ fires does not modify the behaviour of the net w.r.t. the set of properties we want to preserve.

Definition 11 *Let (N, m_0) be a p-agglomerable net. (N, m_0) is*

*1 H-**independent** iff $\forall i \in I$, $\forall h \in H_i$, $\forall m \in Reach(N, m_0)$, $\forall s$ such that $\forall s' \in Pref(s)$, $\Gamma_i(s') \geq 0$, $m[h.s\rangle \Longrightarrow m[s.h\rangle$*

*2 **divergent-free** iff $\forall s \in L^{\omega}(N, m_0)$, $|s|_{T_0 \cup F} = \infty$*

*3 **quasi-persistent** iff $\forall i \in I, \forall m \in Reach(N, m_0)$, $\forall h \in H_i$,
$\forall s \in (T_0 \cup F)^*$, such that $m[h\rangle$ and $m[s\rangle$
$\exists s' \in (T_0 \cup F)^*$ fulfilling: $m[h.s'\rangle$, $\Pi_F(s') = \Pi_F(s)$ and $W(s') \geq W(s)$.
Furthermore, if $s \neq \lambda \Longrightarrow s' \neq \lambda$ then the net is* **strongly** *quasi-persistent.*

*4 H-**similar** iff $\forall i, j \in I, \forall m \in Reach(N, m_0)$, $\forall s \in T_0^*$,
$\forall h_i \in H_i$, $\forall h_j \in H_j, \forall f_j \in F_j$
$m[h_i\rangle$ and $m[s.h_j.f_j\rangle \Longrightarrow \exists s' \in (T_0)^*$, $\exists f_i \in F_i$ such that $m[s'.h_i.f_i\rangle$ and such that $s = \lambda \Longrightarrow s' = \lambda$.*

The H-independence means that once a transition $h \in H_i$ is fireable it can be delayed as long as one does not need it to occur in order to fire a transition of F_i. When a net is divergent-free it does not generate infinite sequences with some suffix included in H. In the pre-agglomeration scheme, we transform original sequences by permutation and deletion of transitions into simulateable sequences. Such an infinite sequence cannot be transformed by this way into an infinite simulateable sequence. Therefore this condition is introduced in order to avoid this situation. The quasi-persistence ensures that in the original net a "quick" firing of a transition of H does not lead to some deadlock which could have been avoided by delaying this firing. At last, the H-similarity forbids situations where the firing of transitions of F is prevented due to a "bad" choice of a subset H_i.

Under the previous conditions (or a subset of), fundamental properties of a net are preserved by the pre-agglomeration reduction. This result is stated in the following theorem whose demonstration is similar to the one of previous theorem.

Its demonstration and the proof that this reduction is a strict extension of the pre-agglomeration of Berthelot [Ber83] are provided in [HPP04].

Theorem 2 *Let (N, m_0) be a p-agglomerable Petri net which is H-independent. If furthermore*

1 (N, m_0) *is divergent-free, strongly quasi-persistent and H-similar then*

$$\Pi_{T_0 \cup F}(L^{max}(N, m_0)) = \Pi_{T_0 \cup F}(\phi(L^{max}(N_r, m_{0r})))$$

2 (N, m_0) *is divergent-free then*

$$\Pi_{T_0 \cup F}(\phi(L^{\omega}(N_r, m_0))) = \Pi_{T_0 \cup F}(L^{\omega}(N, m_0))$$

3 (N, m_0) *is HF-interchangeable, quasi-persistent and H-similar then*

$$(N, m_0) \textit{ is live} \iff (N_r, m_0) \textit{ is live}$$

3. Illustration

3.1. How to define structural conditions

Behavioural hypotheses defined in the previous section cannot be used directly in practice since they refer to the behaviour of the model. In the worst case, verifying these hypotheses leads to building the reachability graph before applying the reductions!

We have designed a set of structural and algebraical conditions that are sufficient to ensure the behavioural hypotheses. These conditions can be automatically and efficiently checked [HPP04]. Unlike the older works about ordinary Petri net reductions [Ber85,Ber86,PPP00,ES01], we intensively define algebraical conditions based on flows and linear invariants of the net. A flow is a vector $\vec{v}$ over the set of places, such that $\vec{v} \cdot W = 0$. A flow $\vec{v}$ induces the following linear invariant: $\forall m \in \mathit{Reach}(N, m_0), \vec{v} \cdot m = \vec{v} \cdot m_0$. These (flows and) invariants can be obtained by two means: the first one is to apply algorithms like the Gaussian elimination or the Farkas algorithm [CS91] when positive constraints on coefficients are required. The second way is to derive already known information when nets are produced by an automatic generation from a high level specification.

We only show the methodology we used to define structural and algebraical conditions corresponding to the behavioural hypothesis defined in previous definitions. We illustrate these principles on the Petri net depicted in Fig 1 for which a simple computation leads to the following invariants:

- $\forall m \in \mathit{Reach}(N, m_0), m(p) + m(q) + m(u) = 1$ meaning that the sum of tokens contained in places p, q and u is always equal to 1.
- $\forall m \in \mathit{Reach}(N, m_0), m(r1) + m(r2) = 1$ meaning that there is always exactly one token in either $r1$ or $r2$.

Let us suppose that we want to establish the following properties:

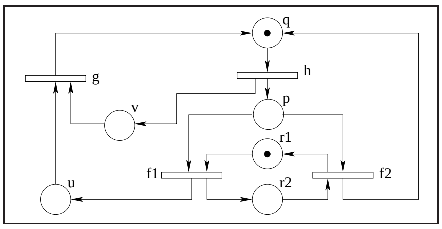

Figure 1: A simple Petri net

1 when the process is in the state p (i.e. p is marked) it is never suspended (either $f1$ or $f2$ is fireable). This illustrates the F-continuation.
2 when the process is in the state p some activity is forbidden (e.g. g is not fireable). This illustrates the H-independence.

For the first property we build a linear programming problem (LP problem) in which we associate with each place p a variable x_p that denotes the number of tokens contained in this place. Thus an assignment of the variables is equivalent to a potential marking since we introduce the linear invariants of the net to characterise a superset of the reachable markings. The constraints of this LP problem are defined by the invariants of the net, by the hypothesis that p is marked and by the negation of the conclusion (i.e. neither $f1$ nor $f2$ are fireable). We conclude that the property is satisfied if the LP is not satisfiable (but the converse is not true).

$$\left[\begin{array}{ll}
\forall i \in P, x_i \geq 0 & \text{the markings are positive} \\
\left.\begin{array}{l} x_q + x_p + x_u = 1 \\ x_{r1} + x_{r2} = 1 \end{array}\right\} & \text{the constraints defined by the invariants are satisfied} \\
x_p \geq 1 & \text{the place } p \text{ is marked} \\
\left.\begin{array}{l} x_{r1} = 0 \\ x_{r2} = 0 \end{array}\right\} & \text{neither the transition } f1 \text{ nor the transition } f2 \text{ is fireable}
\end{array}\right.$$

The second property is expressed similarly. Let us observe that here the negation of the conclusion leads to lower bounds for markings of places.

$$\left[\begin{array}{ll}
\forall i \in P, x_i \geq 0 & \text{the markings are positive} \\
\left.\begin{array}{l} x_q + x_p + x_u = 1 \\ x_{r1} + x_{r2} = 1 \end{array}\right\} & \text{the constraints defined by the invariants are satisfied} \\
x_p \geq 1 & \text{the place } p \text{ is marked} \\
\left.\begin{array}{l} x_v \geq 1 \\ x_u \geq 1 \end{array}\right\} & \text{the transition } g \text{ is fireable}
\end{array}\right.$$

Since we want to prove the non existence of a marking satisfying the linear problem, we should solve an integer linear problem (ILP). It is well-known that solving an ILP may be highly time consuming. Thus a less accurate sufficient condition is to interpret this problem as a rational linear problem. This satisfiability checking is now processed in polynomial time. Moreover, practical experiments have shown that, for the kind of problems we solve, it seldom happens that the ILP is unsatisfiable when the LP is satisfiable.

This shows that the structural conditions are quite accurate w.r.t. the behavioural ones.

3.2. A typical example

Consider the following fragment of a Petri net (modelled in the left-hand side of the figure 2) modelling the access by two threads to data protected by locks (modelled by places Lock1 and Lock2). These locks could be the ones associated with each Java object when Java is used in a multithreaded context. Let us note that the two processes take these locks in a different order but that these actions are performed under the protection of a mutual exclusion mechanism. Note that this construction follows some well-known guidelines used to prevent deadlocks [Hab69].

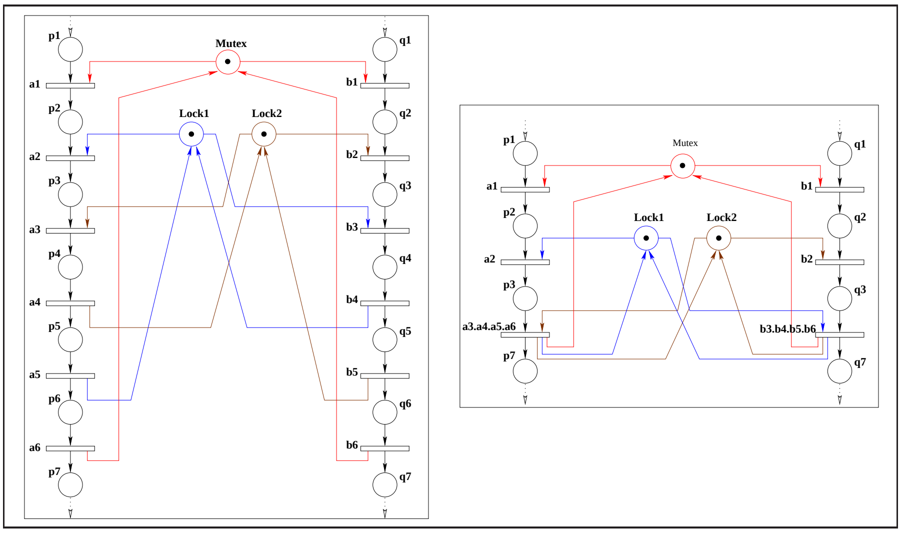

Figure 2: Taking two locks under the protection of a mutex

There exist in this net different binary places invariants (i.e. the corresponding vectors $\vec{v} \in \{0,1\}^P$) ensuring that when place $p2$ is marked then all transitions that have the place $Lock1$ as a pre-condition cannot be fired and symmetrically that, when place $q2$ is marked then all transitions that have the place $Lock2$ as a pre-condition cannot be fired.

We now describe the reduction process. First of all, we post-agglomerate transitions $a3$ with $a4$. Then this new transition $a3.a4$ can be post-agglomerated with $a5$ and then with $a6$. We also apply a similar sequence of post-agglomerations on transitions $b3$ to $b6$ and we obtain then the model on the right-hand side of figure Fig.2. Note that these reductions can be applied without using the algebraical part of the conditions we have proposed in this paper (original Berthelot's conditions are sufficient for performing these reductions). However, after these first reductions, Berthelot's reductions are useless : their conditions forbid further reductions.

The conditions we defined in this paper allow us to perform a (structural) pre-agglomeration. Indeed, if we consider $H = \{a2\}$ and $F = \{a3.a4.a5.a6\}$ we immediately remark that the net is p-agglomerable around place $p3$. Let us prove that the five hypotheses of the pre-agglomerations are fulfilled:

- HF-interchangeability : since $|H| = |F| = 1$ this point is obviously satisfied.
- H-independence : since $a2^{\bullet} \setminus \{p3\} = \emptyset$ the set of transitions that the firing of $a2$ may enable is $a3.a4.a5.a6$. Thus, a sequence s, fireable after the firing of $a2$ from a marking m ($m[a2.s\rangle$) and such that its prefixes s' verify $\Gamma(s') > 0$, is also fireable before the firing of $a2$ ($m[s\rangle$). Let $q \neq p3 \in {}^{\bullet}a2$. As $a2^{\bullet} = \{p3\}$, $W^{+}(q, a2) = 0$. By hypothesis, $m(q) \geq W^{-}(q, a2.s) = Max(W^{-}(q, a2), W^{-}(q, s) - W(q, a2))$. So, $m(q) \geq W^{-}(q, a2) + W^{-}(q, s)$ and $m(q) \geq W^{-}(q, a2) + W^{-}(q, s) - W^{+}(q, s)$. It comes $m(q) + W(q, s) \geq W^{-}(q, a2)$. As, $m[s\rangle$, $m[s.a2\rangle$.
- Divergence freeness : as the place $Lock1$ belongs to ${}^{\bullet}a2 \setminus a2^{\bullet}$, $a2$ cannot be infinitely fired and this point is also fulfilled.
- Quasi-persistence : Let $S = Lock1^{\bullet} \setminus \{a2\}$. By construction, this net satisfies the invariant $\forall m \in Reach(N, m_0), m(Mutex) + m(p2) + m(p3) + m(q2) + m(q3) = 1$. So, when $p2$ is marked, transition $b3.b4.b5.b6 \in S$ is not fireable. So this point is also fulfilled.
- H-similarity : As here H is a singleton this point is obviously fulfilled.

We obtain the net depicted in the left-hand side of figure 3. Now, symmetrically, we perform a pre-agglomeration around place $q3$. This leads to the model at the top right of figure 3.

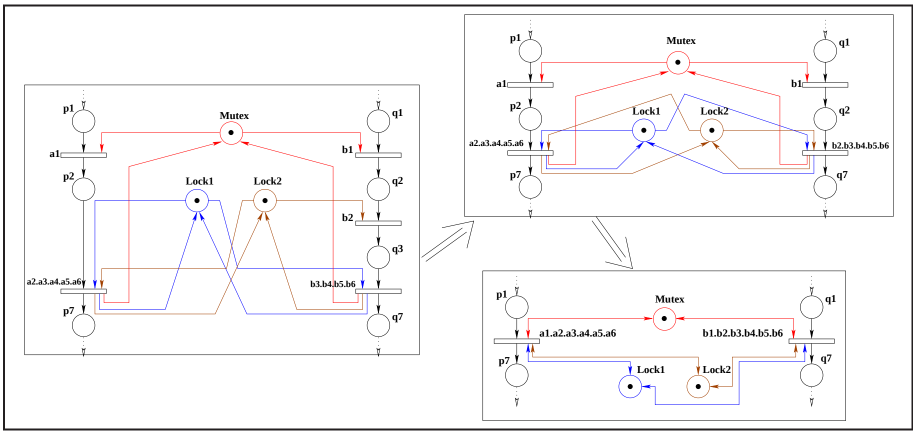

Figure 3: The model at different stages of the reduction process

At last, we can apply on this model a "parallel" pre-agglomeration of $a1$ with $a2.a3.a4.a5.a6$ and of $b1$ with $b2.b3.b4.b5.b6$ ($H_1 = \{a1\}$ and $F_1 = \{a2.a3.a4.a5.a6\}$,

$H_2 = \{b1\}$ and $F_2 = \{b2.b3.b4.b5.b6\}$). Note that it is also possible to first suppress places $Lock1$ and $Lock2$ (that are now implicite places) and then apply two post-agglomerations. In the final model, the two threads operate atomically on the locks.

4. Conclusion

We have presented a method which automatically reduces a Petri net model whereas preserving its behaviour w.r.t. the liveness property and the linear time formulae. Our method is based on a set of rules which merge transitions which are causality dependent whenever some structural conditions are satisfied.

We have significantly enlarged the application field of the reductions previously defined since we have weakened strong local structural conditions and introduced global behavioural conditions specified by (un)satisfiability of linear programming problems. For instance, the structural reductions defined in [Ber85] may be viewed as specialisations of our reductions (as shown in [HPP04]). With such an approach we cover frequently used synchronisation patterns like the monitors, the access control to shared variables and the management of locks.

These algorithms have been implemented in the Quasar tool for analysing concurrent Ada programs [EKPPR03]. With the help of these reductions, large programs have been successfully certified. These experiments show that reductions defined for a low level model (as a semantic for an high level model) cover more patterns in more contexts than reductions directly defined for the high-level model.

References

[Ber83] G. Berthelot. *Transformation et analyse de réseaux de Petri, applications aux protocoles.* Thèse d'état, Université Pierre et Marie Curie, Paris, 1983.

[Ber85] G. Berthelot. Checking properties of nets using transformations. In G. Rozenberg, editor, *Advances in Petri nets*, volume No. 222 of *LNCS*. Springer-Verlag, 1985.

[Ber86] G. Berthelot. Transformations and decompositions of nets. In *Advances in Petri Nets*, number 254 in LNCS, pages 359–376. Springer-Verlag, 1986.

[CGL94] Edmund M. Clarke, Orna Grumberg, and David E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, September 1994.

[CL98] Ernie Cohen and Leslie Lamport. Reduction in TLA. In *International Conference on Concurrency Theory*, pages 317–331, 1998.

[CS91] J. M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal P-semiflows. *LNCS; Advances in Petri Nets 1990*, 483:79–112, 1991. NewsletterInfo: 33,39.

[Doe77] Thomas W. Doeppner, Jr. Parallel program correctness through refinement. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 155–169. ACM Press, 1977.

[EKPPR03] S. Evangelista, C. Kaiser, J. F. Pradat-Peyre, and P. Rousseau. Quasar: a new tool for analysing concurrent programs. In *Reliable Software Technologies - Ada-Europe 2003*, volume 2655 of *LNCS*. Springer-Verlag, 2003.

[ES01] J. Esparza and C. Schröter. Net Reductions for LTL Model-Checking. In T. Margaria and T. Melham, editors, *Correct Hardware Design and Verification Methods*, volume 2144 of *LNCS*, pages 310–324. Springer-Verlag, 2001.

[FQ03a] Cormac Flanagan and Shaz Qadeer. Transactions for software model checking. In Byron Cook, Scott Stoller, and Willem Visser, editors, *Electronic Notes in Theoretical Computer Science*, volume 89. Elsevier, 2003.

[FQ03b] Cormac Flanagan and Shaz Qadeer. A type and effect system for atomicity. In *Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, pages 338–349. ACM Press, 2003.

[FQ03c] Stephen N. Freund and Shaz Qadeer. Checking concise specifications for multithreaded software. In *Formal Techniques for Java-like Programs*, 2003.

[Gri96] E. Pascal Gribomon. Atomicity refinement and trace reduction theorems. In Rajeev Alur and Thomas A. Henzinger, editors, *CAV*, volume 1102, pages 311–322, New Brunswick, NJ, USA, / 1996. Springer Verlag.

[Hab69] A. N. Habermann. Prevention of system deadlocks. *ACM*, 12(7):373–ff., 1969.

[HPP04] S. Haddad and J.F. Pradat-Peyre. Efficient reductions for LTL formulae verification. Technical Report 634, CEDRIC, CNAM, Paris, http://cedric.cnam.fr, 2004.

[Lip75] Richard J. Lipton. Reduction: a method of proving properties of parallel programs. *Commun. ACM*, 18(12):717–721, 1975.

[LS89] Leslie Lamport and Fred B. Schneider. Pretending atomicity. Technical Report TR89-1005, 1989.

[PPP00] D. Poitrenaud and J.F. Pradat-Peyre. Pre and post-agglomerations for *LTL* model checking. In M. Nielsen and D Simpson, editors, *High-level Petri Nets, Theory and Application*, number 1825 in LNCS, pages 387–408. Springer-Verlag, 2000.