Efficient Reductions for *LTL* Formulae Verification

Serge Haddad [a] and Jean-François Pradat-Peyre [b]

[a] *LAMSADE-CNRS UMR 7024 Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, FRANCE*

[b] *CEDRIC-CNRS EA 1395, Conservatoire National des Arts et Métiers, 292 rue Saint Martin, 75141 Paris Cedex 03, FRANCE*

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 ordinary Petri nets reductions. At 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 *LTL* formula that does not observe reduced transitions of the net. We substitute 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 significant and typical examples.

1 Introduction

It is currently admitted that the use of formal methods is essential to obtain less error-prone complex software. Such a process is decomposed in two steps: a modelling stage which must lead to a model as close as possible to the analysed software and a verification stage involving properties expression and model checking via appropriate algorithms.

In this context, there are two kinds of verification techniques: state enumeration based and structural algorithms. In case of finite state systems, the former ones lead to a complete verification but the analysis is restricted by the inherent combinatory explosion factor. Moreover it does not give insight on how to identify and correct the faulty parts of the software. 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 do 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. 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 work, we will focus on the latter abstraction. The main advantage of such a transformation is the drastic reduction of the combinatory explosion due to the elimination of the intermediate states. In the context of software engineering, different solutions for this kind of transformations have been proposed; e.g. [Cor98], [Mis03], [FQ03a, QRR04] or [Hol03].

However they suffer three drawbacks: they are language dependent, they are only partially automated and at last, due to the lack of formal semantics of the analysed languages, they cannot be fully theoretically justified.

Thus, we have chosen to develop abstractions for a low-level model with a formal semantic: the Petri nets. The advantages of this approach are threefold:

- Due to the formal semantic, 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.
- The developed abstractions can be straightforwardly adapted or specialised for a target high-level model.

Our work on these abstractions is an important generalisation of the reduction method proposed by Berthelot in [Ber83, Ber85]. We focus here on the most important reductions proposed by this author: the pre-agglomeration and the post-agglomeration. These reductions merge sequential transitions into an atomic one reducing considerably the number of reachable states. Original application conditions of Berthelot's reductions rely only on structural conditions. 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.

We proceed here in a different way: first, we characterise a set of behavioural conditions that ensure the preservation of the considered properties. Indeed, defining behavioural conditions simplifies the search for alternative sets of sufficient structural conditions. 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 algorithms are known. Such constraints express restrictions on the global behaviour of the net which were not taken into account by the previous reductions. At last, for each family of properties, we detail the required subset of conditions in order to enlarge their application.

The paper is organised as follows. The second section emphasises the interest of Petri nets and temporal logic in the context of concurrent software engineering and presents alternative approaches. In the third section, we define two behavioural agglomerations including their conditions of applications, the transformation rule and the preserved properties. The forth section develops structural and algebraic sufficient conditions for the behavioural ones. Then, we illustrate our reductions on significant and typical examples.

The appendix A contains the syntax and the semantics of Petri nets whereas the appendix B deals with the proofs of the different propositions and theorems given in the paper.

2 Concurrent software verification

In many contexts, the introduction of concurrent activities enables to mimic the structure of the application domain in which natural parallelism and cooperation often occur. The implementation is simpler and it gains in scalability.

However, while providing many advantages, concurrency introduces also specific difficulties due to the non determinism and to the numerous interactions between activities (e.g. deadlock occurrence or fairness violation).

Standard test methods are not sufficient to detect these kinds of problems. For instance, it is well-known that reproducing an error is a difficult task. Thus verification methods must be applied to enforce confidence in concurrent software. The main difficulty is then to cope with the combinatory explosion. Among several strategies, abstracting a sequence of actions as an atomic one is very efficient since it reduces the interlacing of processes. Here, the main problem is to exhibit conditions which, on the one hand, are not too restrictive and on the other hand preserve the significant part of the application behaviour.

2.1 Petri nets and the atomicity problem

2.1.1 Informal presentation of Petri nets

The Petri nets model [Rei83] is a suitable formalism for the representation of concurrency. It combines a simple syntax with a precise semantic and it supports numerous analysis tools which either use a state-space exploration technique or exploit the structure of the model. In particular, some structural techniques, like invariants computation, give "high-level" information about the structure of processes, variable bounds, repetitive sequences, and so on, without needing any execution of the model [Mur89].

Furthermore, there exist abbreviations of Petri nets, like coloured nets [Jen91], that allow a more concise description of systems while preserving analysis abilities by maintaining an equivalence with ordinary Petri nets. In a coloured net, a place contains typed (or coloured) tokens instead of anonymous tokens, and a transition may be fired in multiple ways (i.e. instantiated). More precisely, to each place and each transition is attached a type (or a colour) domain. An arc from a transition to a place (resp. from a place to a transition) is labelled by a linear function called a colour function. This function determines the number and the type (or the colour) of tokens that have to be added to (or removed from) the place upon firing the transition with respect to a colour instantiation. By definition, a coloured net is always an abbreviation of an ordinary Petri net, called the *underlying* Petri net.

We illustrate our reductions with coloured nets models because they provide "high level" description capabilities. However, since they remain equivalent to ordinary Petri nets [1], we look here for an abstraction method applicable to Petri nets. In a subsequent work, we will show how to directly work at the coloured Petri net level.

2.1.2 Illustration of the atomicity problem

Let us model a variable incrementation, `VarX := VarX + 1`, performed by a process.

Depending on the data type, the programming language, the compiler and the operating system, different semantics are possible. We give below two modellings of such a statement. In the left model of the figure Fig.1 the statement is considered to be atomic. In the right model, this statement is considered to be performed in three steps; first a local copy of the variable is made, then the local copy is incremented (atomically) and, third, the local copy is written

[1] This equivalence is based on the finite nature of the colour domains.

into the global variable.

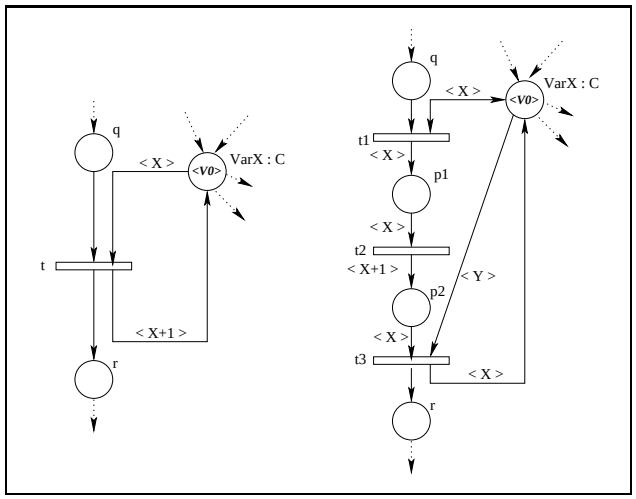

Fig. 1. Are these two assignments equivalent and for which properties ?

In both nets, the place q models the process state before the assignment and the place r the state after the statement. These two places contain ordinary tokens. [2] Place `VarX` models the variable to be incremented, and is coloured upon a colour domain named `C` that represents the domain of the variable. This place always contains a single token whose colour is the value of the variable.

In the first model, the transition `t` may be fired as soon as a process is in state `q` (i.e. the place `q` is marked). Then the variable `X` is bound to the colour of the token contained in the place `VarX`. This token is consumed and replaced by a token of the next value (i.e. the value of the expression `X+1`). The process changes its state (i.e. the neutral token moves from q to r). Observe that `X` is a variable of the coloured Petri net local to the transition t which should not be confused with the program variable `VarX` modelled by the eponymous place.

In the second model, the transition `t1` produces a token in place `p1` with the same colour as the one in `VarX` letting this token unchanged. The places `p1` and `p2` both model the state of the process and the value of the local copy of the variable. Then the firing of the transition `t2` performs the local incrementation of the value by producing an appropriate token in place `p2`. At last, the firing of `t3` replaces the value of the variable `VarX` by the computed value. Note that transition `t3` may be potentially fired for any instance of the domain $C \times C$ corresponding to the variables `X` and `Y`. Here, `Y` is bound to the current value of `VarX`, whereas `X` is bound to the current value of the local copy of the process.

[2] Note that these places may be coloured upon a domain reflecting more precisely a state of a process, including, for instance, the process identifier, the value of local variables, or other useful informations.

Now the fundamental problem for the application designer is to have some guarantee that the two models are equivalent and more precisely to know for which kinds of properties the equivalence is ensured. It should be clear that such an equivalence depends on the global behaviour of the program. The topic of this work is exactly to characterise such behaviours by mean of sets of conditions.

2.2 Expressing and verifying properties

Verification of a concurrent software system aims to check that the dynamic behaviour of the software satisfies some properties. There are two kinds of properties the designer is interested with: generic properties like termination, deadlock occurrence, etc. or specific ones related to the functionalities of the applications. Such properties are often defined with the help of a temporal logic which considers either the execution as a computation tree (i.e. branching time logics) or as a set of states and/or actions sequences (i.e. linear time logics). We focus here on the latter logics whose a typical representative is the linear time temporal logic (*LTL* for short) defined by Pnueli [Pnu81]. In *LTL*, properties are described by mean of logic formulae using atomic propositions (involving states for "state-based logic" or actions for "action-based logic"), boolean operators (and, not, or) and specific temporal operators (Until, Next, etc.). Our work does not depend on a particular logic and is action based whereas usual transformations from state-based logics to action-based ones [DV90] may be used to extend our results. Since we merge transitions firings we solely require that the truth of a formula depends only on the projection of the sequences on action occurrences involved in the formula.

We take into account two kinds of sequences: maximal finite sequences (e.g. relevant for deadlock detection) and infinite sequences (e.g. for fairness properties).

2.3 Related works

The first theoretical work concerning reduction of sequences into atomic actions for simplification purpose was performed by Lipton in [Lip75] which focused on deadlock property preservation. This work has been extended by Doeppner, Schneider, Cohen and Lamport in different papers [Doe77, LS89, Gri96, CL98] aiming at preserving safety or liveness properties. More recently, Freund, Qadeer and Flanagan [FQ03b, FQ03b, FQ03c, FQ03a] leveraged the Lipton's theory of reduction to detect transactions in multithreaded programs that can be considered as atomic. The main drawback of these approaches is

the difficulty to detect conditions allowing to apply the reduction while staying at a very abstract level.

In Petri nets formalism, the first works concerning reductions have been performed by Berthelot [BRV80,Ber83,Ber85,Ber86]. The author focused only on specific Petri nets properties preservation such like liveness or boundedness. The link between transition agglomeration and general properties, expressed in LTL formalism, is done in [PPP00]. All these reductions lied on "pure" structural application conditions and then have a quite thin application area. Weakening of application conditions of the pre and post agglomeration have been studied in [Had87]. In this work, only specific Petri nets properties are considered. More recently, Esparza and Schröter, simplify one point in the original pre agglomeration conditions in [ES01]. However, they consider only 1-safe Petri nets (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!

At a software level, several works address the possibility of defining directly in the source program syntactic constructions enforcing atomicity of sequential actions. One can cite [Lom77] or [Hol03]. These constructions reduce the benefit induced by the parallelism of the executions and furthermore they are not implementable on a architecture that does not support such mechanisms. Alternatively, such annotations can be used to verify that each block of code annotated as being atomic does not interfere and is not affected by other threads [FQ03b,HRD04,FF04]. These verifications use either static type system or model checking techniques.

3 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).

We propose in this section two net reductions, the pre and the post agglomeration, that preserve a large class of properties (liveness and any *LTL* formula that not observing some specific transitions) under simple behavioural hypotheses.

The proposed hypotheses rely on the behaviour of the model; so single definitions handle a large set of specific cases. Furthermore, as each hypothesis characterises a specific behavioural constraint, we develop for each of them sufficient conditions that are based on the structure of the model (i.e. checked by a direct examination or by the satisfaction of linear problems). These struc-

tural conditions are given in section 4.

3.1 Notations

We assume that the reader is familiar with the classical Petri nets definitions. We only recall in this section needed notations; complete definitions are provided in Appendix A.

- We note $\langle P, T, W^+, W^-, m_0 \rangle$ a marked Petri net;
- The transitions linked to a place, or the places linked to a transitions are defined by:
  - $\forall p \in P$, ${}^\bullet p = \{t | W^+(p,t) > 0\}$ and $p^\bullet = \{t | W^-(p,t) > 0\}$;
  - $\forall t \in T$, ${}^\bullet t = \{p | W^-(p,t) > 0\}$ and $t^\bullet = \{p | W^+(p,t) > 0\}$;
  - We also extend in a natural way this notation to subsets of places and transitions.
- λ defines the empty sequence of transitions;
- If s is a 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.

3.2 Agglomeration scheme

We suppose on the sequel that the set of transitions of the net is partitioned as: $T = T_0 \uplus_{i \in I} H_i \uplus_{i \in I} F_i$ where I denotes a non empty set of indices. The underlying idea of this decomposition is that a couple (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 will develop conditions on the behaviour (or alternatively on the structure) 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. to a set of properties. This restricted behaviour is the behaviour of a reduced net as shown in the next definitions and propositions.

DEFINITION **3.1 (Reduced net)** *Let (N, M_0) be a Petri net and suppose that $T = T_0 \uplus_{i \in I} H_i \uplus_{i \in I} F_i$ The reduced Petri net (N_r, m_{0r}) w.r.t. this partition is defined by:*

- $P_r = P$ *and* $T_r = T_0 \cup_{i \in I} (H_i \times F_i)$.
  One denotes by hf *the transition* (h, f) *of* $H_i \times F_i$*;*
- $\forall t_r \in T_0, \forall p \in P_r$, $W_r^-(p,t) = W^-(p,t)$ *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* $W_r^+(p,hf) = W^+(p,h.f)$ *(see Appendix A)*
- $m_{0_r} = m_0$

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).

We want to compare the behaviour of the reduced and the original nets. However the sets of transitions are not identical. Thus, the following one to one homomorphism allows such a comparison.

Definition **3.2** *We note* ϕ *the homomorphism from the monoid* T_r^* *to the monoid* T^* *defined by:*

$$\forall t \in T_0, \phi(t) = t \text{ and } \forall i \in I, \forall h \in H_i, \forall f \in F_i, \phi(hf) = h.f$$

This homomorphism is extended to an homomorphism from $\mathcal{P}(T_r^*)$ *to* $\mathcal{P}(T^*)$ *and from* $\mathcal{P}(T_r^\infty)$ *to* $\mathcal{P}(T^\infty)$.

Among sequences of the original net, some of them look like more or less as sequences of the reduced net. It depends on the way the transitions of H_i are immediately followed by a transition of F_i.

Definition **3.3 (Simulateable sequence)** *Let* $J \subseteq I$*, A sequence* $s \in T^*$ *(resp.* T^∞*) is said to be* J**-simulateable** *or simulateable for short when* $J = I$ *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 $\forall m,\ s_m \in \uplus_{i \in J} H_i \times F_i$ *and* $s'_m \in (T_0 \uplus_{i \notin J} (H_i \cup F_i))^*$

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

As mentioned previously, three kinds of properties are of interest when performing model checking :

- The liveness of the net which is a central property.
- Properties of a linear temporal logic evaluated on finite maximal sequences (including for instance the presence of deadlock markings).
- Properties of a linear temporal logic evaluated on infinite sequences (including for instance fairness properties).

The next basic theorem states in a formal way that the behaviour of the reduced net is a subset of the original behaviour.

THEOREM 1 *Let (N, m_0) be a net. Then:*

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

PROOF. Straightforward from proposition 11 (in the Appendix A, page 32).

At this point, we know that if a maximal or infinite sequence violates a 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 dependency 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. Using these functions we characterise potentially agglomerable Petri nets, for which each occurrence of F_i is preceeded by an occurrence of H_i. We give then a simple structural scheme which ensures that a net is potentially agglomerable.

DEFINITION **3.4 (Counting functions)** *Let $s \in T^*$ be a finite sequence. We note $\Gamma_i(s) = |s|_{H_i} - |s|_{F_i}$.*

DEFINITION **3.5 (Potentially agglomerability)** *A marked net (N, m_0) is*

(1) potentially agglomerable (p-agglomerable for short) iff $\forall s \in L(N, m_0)$, $\forall i \in I$, $\Gamma_i(s) \geq 0$.
(2) structurally p-agglomerable (sp-agglomerable for short) iff $\forall i \in I$, $\exists p_i$ such that
  (a) $m_0(p_i) = 0$, ${}^\bullet p_i = H_i$, $p_i{}^\bullet = F_i$;
  (b) $\forall h \in H_i, \forall f \in F_i$, $W^+(p_i, h) = W^-(p_i, f) = 1$.

Let us note that for a firing sequence $s \in L(N, m_0)$ leading to m, one has $\Gamma_i(s) = m(p_i) \geq 0$. Thus the structural condition ensures the behavioural one: a net that is sp-agglomerable is necessarily p-agglomerable.

In the following, we study p-agglomerable nets. The remainder of the section is devoted to the presentation of two sets of conditions thet 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.

3.3 Behavioural pre-agglomeration

We state in the following definition five 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 **3.6** *Let (N, m_0) be a p-agglomerable net. (N, m_0) is*

(1) HF***-interchangeable*** *iff* $\forall i \in I$*, one of these two conditions is fulfilled:*
(a) $\forall m \in Reach(N, m_0)$, $\forall h, h' \in H_i$, $\forall f \in F_i$, $m[h.f\rangle \iff m[h'.f\rangle$
(b) $\forall m \in Reach(N, m_0)$, $\forall h \in H_i$, $\forall f, f' \in F_i$, $m[h.f\rangle \iff m[h.f'\rangle$

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

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

(4) ***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.*

(5) 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* $s = \lambda \Longrightarrow s' = \lambda$.

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 H-independence roughly means that once a transition $h \in H_i$ is fireable it can be delayed as long as one does not need its occurrence 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 to simulateable sequences. Such an infinite sequence cannot be transformed by this way into an infinite simulateable sequence. Therefore this condition is mandatory. 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 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 provided in Appendix B.

THEOREM 2 *Let (N, m_0) be a Petri net.*

(1) If (N, m_0) is p-agglomerable, H-independent, HF-interchangeable, quasi-persistent and H-similar then

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

(2) If (N, m_0) is p-agglomerable, H-independent, divergent-free, strongly quasi-persistent and H-similar then

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

(3) If (N, m_0) is p-agglomerable, H-independent and divergent-free then

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

3.4 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 **3.7** *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:*
(a) $\forall m \in Reach(N, m_0)$, $\forall h, h' \in H$, $\forall f \in F$, $m[h.f\rangle \iff m[h'.f\rangle$
(b) $\forall m \in Reach(N, m_0)$, $\forall h \in H$, $\forall f, f' \in F$, $m[h.f\rangle \iff m[h.f'\rangle$

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

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

We express the strong dependence of the set F on the set H with three hypotheses. We have already discussed the HF-interchangeability hypothesis. 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 "makes possible" 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.

THEOREM 3 *Let (N, m_0) be a Petri net.*

(1) If (N, m_0) is p-agglomerable, F-continuable, F-independent and HF-interchangeable then

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

(2) If (N, m_0) is p-agglomerable and F-continuable then

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

(3) If (N, m_0) is p-agglomerable, F-continuable and F-independent then

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

As for the pre-agglomeration, the proof of this theorem is obtained progressively by different lemmas in Appendix B, section B.2, pages 41-43.

4 Petri nets structural agglomerations

4.1 Methodology

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 the building of the reachability graph before the reductions!

So we propose in this section some structural and algebraical conditions that are sufficient to ensure the behavioural hypotheses. Unlike the older works about ordinary Petri net reductions [Ber85,Ber86,PPP00,ES01], we intensively use algebraical conditions based on linear invariants of the net. This allows us to considerably enlarge the application spectrum of these reductions. These invariants can be obtained from two ways: 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.

Before specifying these structural and algebraical conditions, we illustrate such a methodology on the Petri net depicted Fig 2 for which a simple computation

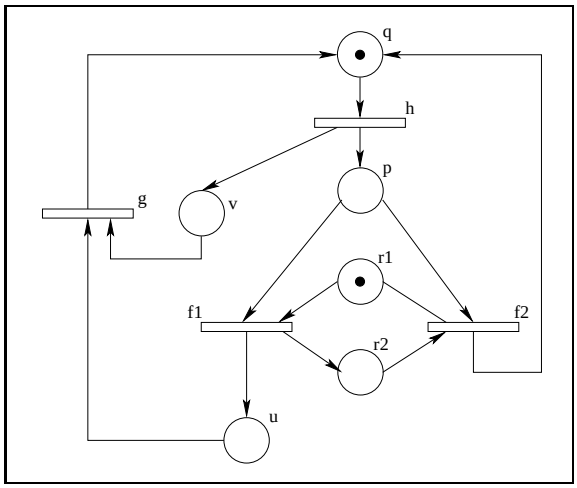

Fig. 2. A simple Petri net

leads to the following invariants:

- $\forall m \in Reach(N, m_0), m(p) + m(q) + m(u) = 1$ meaning that whatever the marking reached, the sum of tokens contained in place p, q and u is equal to 1. This invariant characterises a process with p, q and u as potential states.
- $\forall m \in Reach(N, m_0), m(r1) + m(r2) = 1$ meaning that there is always exactly one token in either $r1$ or $r2$.

Let us suppose that we want to establish the following properties (we will encounter these two kinds of properties in a more general context during the elaboration of the structural conditions):

(1) when the process is in the state p (i.e. p is marked) then it is never suspended (i.e. necessarily either $f1$ or $f2$ is fireable);
(2) when the process is in the state p some activity is forbidden (e.g. g is not fireable).

For the first property we build a linear programming problem (LP problem) in which we associate to 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 and we can use the linear invariants of the net for characterising 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 not only if). Let us observe that this translation into a single LP problem is possible due to the particular form of the preconditions of $f1$ and $f2$. Starting

from other standard situations, a single LP problem is also produced (see later the F-continuation). But generally, the translation may lead to multiple LP problems.

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

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

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

More generally this kind of property corresponds to a scheme often encountered in our conditions: the marking of a place p disables the fireability of a subset of transitions. So, we introduce the notion of transition freezing based on LP problems.

PROPOSITION 1 (TRANSITION FREEZING) *Let (N, m_0) be a Petri net, p be a place and t be a transition. Suppose that the LP problem where:*

- *the variables are $\{x_q\}_{q \in P}$*
- *the constraints are given by the positivity of the variables, the invariants of the net and by the inequations $x_p \geq 1$ and $\forall q \in {}^\bullet t$, $x_q \geq W^-[q, t]$*

does not admit a solution. Then $\forall m \in Reach(N, m_0)$, $m(p) > 0 \Longrightarrow$ NOT $m[t\rangle$. We say that p **freezes** *t. By extension, p freezes a set of transitions T' if $\forall t \in T'$, p freezes t.*

The proof of this proposition is straightforward.

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.

4.2 Structural Pre-agglomeration

We propose in this section five structural conditions that imply the respect of the behavioural hypotheses used in the previous section. Some of them are only based on structural constraints while others use both structural and algebraical conditions.

The HF-interchangeable hypothesis ensures that any transitions of H_i can be replaced by any other transition of H_i or similarly that that any transition of F_i can be replaced by any other transition of F_i. We propose four simple structural conditions that guarantee such a behaviour. The conditions 2 and 3 are related to the F-interchangeability and the conditions 1 and 4 are related to the H-interchangeability. The technical aspect of the point 4 is due to the fact that if H_i and F_i are not reduced to a singleton, then the possibility to replace a transition h by a transition h' implies an equivalence in term of pre-conditions but also an equivalence in term of tokens produced by these transitions and needed for the firing of a transition of F_i.

Proposition 2 (Structural HF-interchangeability) *A sp-agglomerable net (N, m_0) is HF-interchangeable if $\forall i \in I$, one of these conditions is fulfilled:*

(1) $|H_i| = 1$
(2) $|F_i| = 1$
(3) $\forall f, f' \in F_i$, $\forall p \in P$, $W^-(p, f) = W^-(p, f')$
(4) $\forall h, h' \in H_i$
- $\forall p \in P, W^-(p, h) = W^-(p, h')$ *and*
- $\forall f \in F_i, \forall p \in h^\bullet \cap {}^\bullet f$, $W^+(p, h) = W^+(p, h')$

The proof of this proposition is straightforward.

In the following figure (Fig. 3), the model on the left does not verify the structural conditions: neither H_i nor F_i are reduced to a singleton, and place $q2$ prevents from verifying point 3 or 4. In the model of the right, since $Pre(q2, f) = Pre(q2, f')$ the point 3 is fulfilled. Thus the HF-interchangeability is verified.

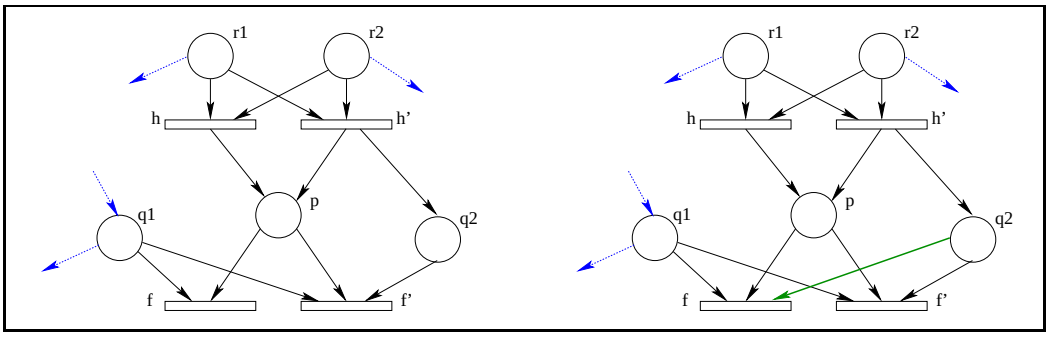

Fig. 3. Example of nets that verify or not the structural HF-interchangeability

In order to obtain a structural characterisation of the H-independence, we require first that the tokens produced by a transition $h \in H_i$ (other than the one produced in p_i) cannot be consumed by a transition which does not belong to F_i while the place p_i is marked. Furthermore, in the case where such a token can be consumed by a transition of F_i, the transitions H_i are frozen by the place p_i.

PROPOSITION 3 (STRUCTURAL H-INDEPENDENCE) *A sp-agglomerable net* (N, m_0) *is* H*-independent if:*
$\forall i \in I$, *denoting* $HP_i = (H_i^\bullet \setminus \{p_i\})$,

a) p_i *freezes* $HP_i^\bullet \setminus F_i$
b) if $HP_i^\bullet \cap F_i \neq \emptyset$ *then* p_i *freezes* H_i

The proof of this proposition is given in Appendix B, section B.3, page 44.

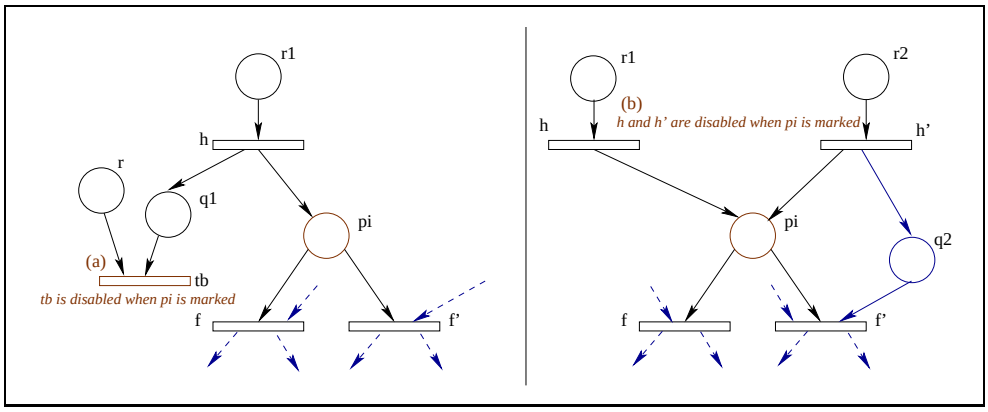

Fig. 4. Illustration of the structural conditions of the H-independence

The divergence freeness hypothesis focuses on the possibility to enter in an infinite loop composed only by transitions of H. In order to structurally forbid this behaviour we impose that either the places p_i are structurally bounded

(point 1) or that the firing of transitions of H needs tokens that are not produced by these transitions (point 2). In both cases, the undesirable behaviour is disabled.

PROPOSITION 4 (STRUCTURAL DIVERGENCE FREENESS) *A sp-agglomerable net (N, m_0) is divergent-free if $\forall i \in I$,*

(1) either p_i is covered by a positive flow
(2) or $\forall h \in H_i$, $\exists q \in {}^\bullet h$ such that ${}^\bullet q \subset T_0 \cup F$.

The proof of this proposition is given in Appendix B, section B.3, page 45.

The main idea on which is based the structural condition of the quasi-persistence is that any transition that can be in conflict with a transition h of H either has no impact on the marking (it is a neutral transition) or that such a conflict is not effective. One more time, this last point is obtained by the expression of a linear programming problem using positive flows of the net.

PROPOSITION 5 (STRUCTURAL QUASI-PERSISTENCE) *A sp-agglomerable net is quasi-persistent if one of the following structural conditions are verified: $\forall h \in H$, $\forall t \in ({}^\bullet h)^\bullet \setminus H$, then*

(1) either $t \in T_0$ is a neutral transition
(2) or the linear programming problem where
- *the variables are $\{x_q\}_{q \in P}$,*
- *the constraints are defined by the positivity of the variables, the invariants of the net and by the inequations $\forall q \in {}^\bullet h, x_q \geq W^-[q, h]$ and $\forall q \in {}^\bullet t, x_q \geq W^-[q, t]$*

does not admit a solution.

If all transitions of $({}^\bullet h)^\bullet \setminus H$ verify the point 2 then the net is strongly quasi-persistent.

The proof of this proposition is given in Appendix B, section B.3, page 45.

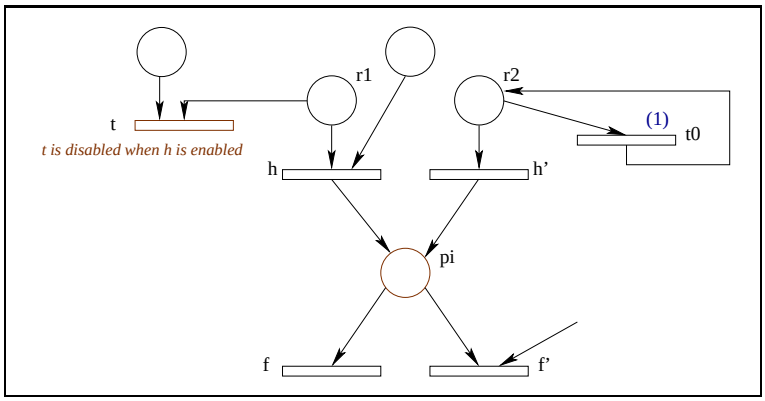

Fig. 5. Illustration of the structural conditions of the quasi-persistence

The H-similarity hypothesis states that when $|I| > 1$ a "bad" choice cannot be made between a transition of H_i and a transition of H_j with $i \neq j$.

PROPOSITION 6 (STRUCTURAL H-SIMILARITY) *A sp-agglomerable net which is H-independent and quasi-persistent is H-similar if:*
$\forall i, j \in I$, $\forall h_j \in H_j, f_j \in F_j$, $\forall h_i \in H_i$,
$\exists f_i \in F_i$ *such that* $\forall p \in {}^\bullet f_i \setminus \{p_i\}$, $W^-(p, h_j.f_j) \geq W^-(p, h_i.f_i)$.

The proof of this proposition is given in Appendix B, section B.3, page 45.

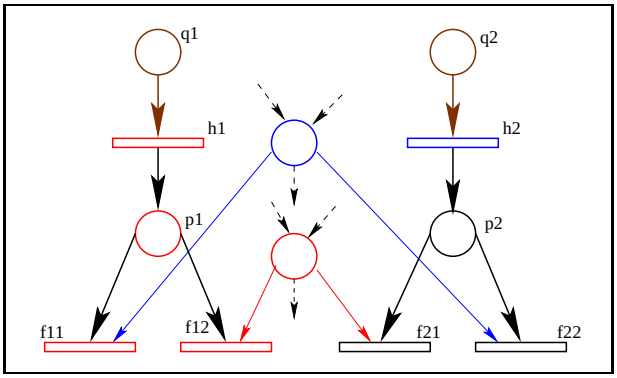

Fig. 6. Illustration of the structural conditions of the H-similarity

PROPOSITION 7 (GENERALISATION OF BERTHELOT'S REDUCTION) *The previous structural and algebraical conditions generalise the conditions of the pre-agglomeration proposed by Berthelot in [Ber83, Ber85].*

The proof of this proposition is not present here since it would require a detailed presentation of Berthelot's reductions. The interested reader will found it in [HPP04].

4.3 Structural Post-agglomeration

In this section, we propose structural conditions for the two post-agglomeration specific behavioural properties: the F-independence and the F-continuation. We recall that $I = \{1\}$, so we abbreviate p_1 by p.

The F-independence hypothesis supposes that any transition f of F can commute with sequences of $(T_0 \cup H)^*$ (or with sequences s in T^* s.t. $\forall s' \in Pref(s), \Gamma(s') \geq 0$ for the strong version). The first way for obtaining this behaviour is to suppose that no transition which produces useful tokens for the firing of F can be fired when p is marked (including H for the strong version). A second way is to require that the structure of the net around concerned transitions is such that transition f commutes with other transitions.

PROPOSITION 8 (F-INDEPENDENCE) *A sp-agglomerable net (N, m_0) is F-independent if $\forall f \in F$, $\forall q \in ({}^\bullet f \setminus \{p\})$, $\forall t \in ({}^\bullet q \setminus F)$*

(1) either p freezes t
(2) or t and f fulfill conditions a) and b)
  (a) $W^-(q,t) \geq min(W^+(q,t), W^-(q,f))$
  (b) $W^+(q,f) \geq min(W^-(q,f), W^+(q,t))$

A F-independent net is strongly F-independent if p freezes H.

The proof of this proposition is given in Appendix B, section B.4, page 46.

At last, the F-continuation condition ensures that there exists always a transition of F that is fireable as soon as p is marked.

PROPOSITION 9 (F-CONTINUATION) *A sp-agglomerable net (N, m_0) is F-continuable if one of the three conditions is fulfilled:*

(1) $\exists f \in F$ such that ${}^\bullet f = \{p\}$
(2) or $\exists$ $F_s \subset F$ such that:
  (a) all transitions f of F_s have only one input place p_f different from p,
  (b) the linear programming problem where the variables are $\{x_q\}_{q \in P}$, the constraints are given by the positivity of the variables, the invariants of the net and by the inequations $\forall p_f \in {}^\bullet F_s \setminus \{p\}, x_{p_f} \leq W^-[p_f, f] - 1$ and $x_p \geq 1$ does not admit a solution.
(3) or $\exists$ $F_s \subset F$ such that:
  (a) $\forall q \in {}^\bullet F_s$, q is a structural safe place (e.g. is covered by a binary positive flow)
  (b) $\forall f \in F_s$, $\forall q \in {}^\bullet f$, $W^-(q,f) = 1$
  (c) the linear programming problem where the variables are $\{x_q\}_{q \in P}$, the constraints are given by the positivity of the variables, the invariants of the net and by the inequations $\forall f \in F_s$ $\sum_{q \in {}^\bullet f \setminus \{p\}} x_q \leq |{}^\bullet f| - 2$ and $x_p \geq 1$ does not admit a solution.

The proof of this proposition is straightforward.

In the left model of figure Fig.7 the invariant $\forall m, m(q1) + m(q2) = 2$ ensures that there can not exist a marking m such that $m(q1) = 0$ and $m(q2) \leq 1$. So, as soon as p is marked one of the transition $f1$ or $f2$ is fireable. In the right model, let us suppose that there exists two invariants $m(q1) + m(q2) = 1$ and $m(r1) + m(r2) = 1$ (which may correspond to the fact that places $q1$ and $q2$ model the two possible values of a variable q when places $r1$ and $r2$ model the two possible values of a variable r). These invariants ensures that one of the four transition $f11$ to $f22$ is fireable as soon as p is marked. Obviously, the place p may have other output transitions in both cases.

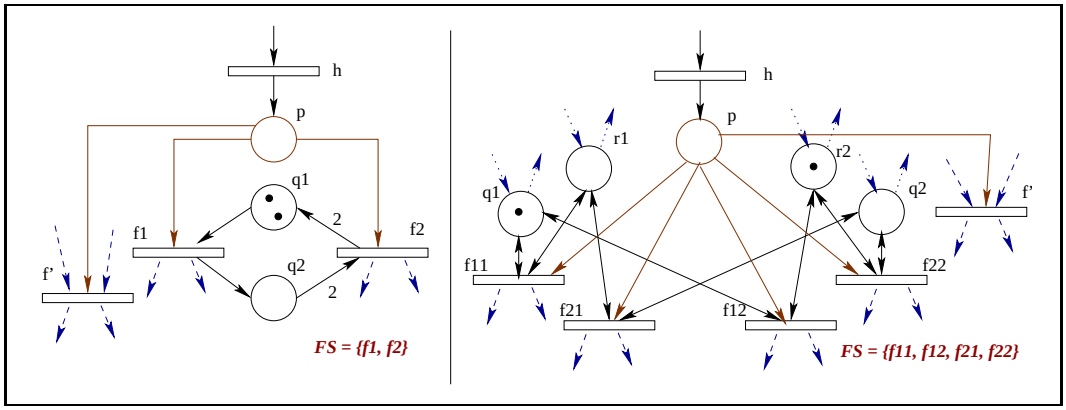

Fig. 7. Illustration of the structural conditions of the F-continuation

PROPOSITION 10 (GENERALISATION OF BERTHELOT'S REDUCTION) *The previous structural and algebraical conditions generalise the conditions of the post-agglomeration proposed by Berthelot in [Ber83, Ber85].*

The proof of this proposition is not present here since it would require a detailed presentation of Berthelot's reductions. The interested reader will found it in [HPP04].

5 Examples

5.1 Sharing multiple locks

Consider the following fragment of Petri net (figure Fig.8) modelling the access by two threads to data protected by locks (modelled by places Lock1 and Lock2). These locks could be the ones associated to 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.

Suppose now that there exist some binary places invariants ensuring that when $p2$ is marked then all transitions that have $Lock1$ as a pre-condition cannot be fired and symmetrically that, when $q2$ is marked then all transitions that have $Lock2$ as a pre-condition cannot be fired. The use of mutexes (or an equivalent synchronisation mechanism) could lead to such invariants as done in the model depicted in the figure Fig.8. Remark that this construction follows some well-known guidelines used to prevent deadlock [Hab69].

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

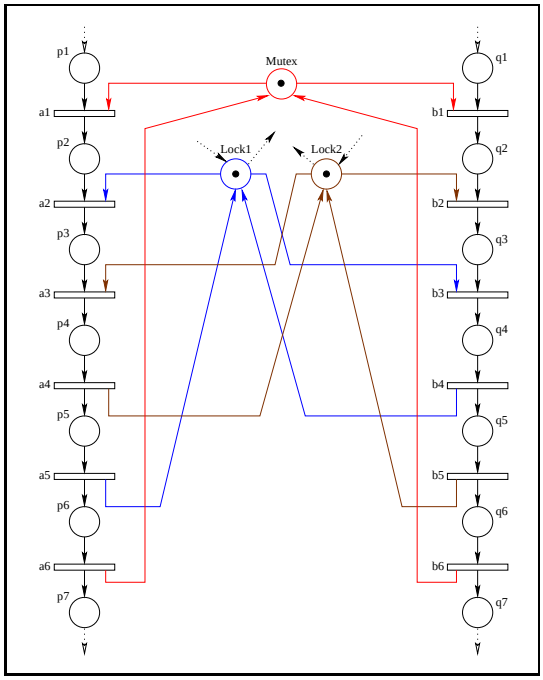

Fig. 8. Taking two locks under the protection of a mutex

a similar sequence of post-agglomerations on transitions $b4$ to $b7$ and we obtain then the model depicted in figure Fig.9. 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, the Berthelot's reductions are useless.

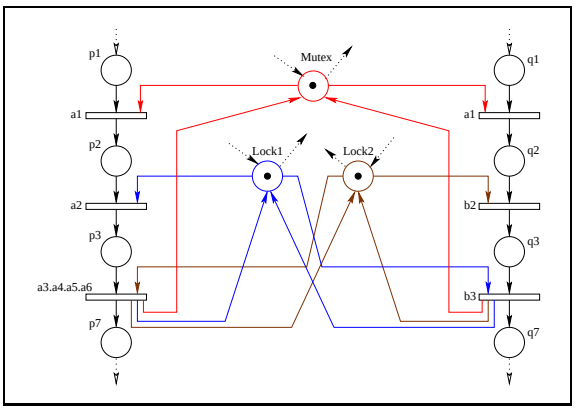

Fig. 9. A first reduced model

The conditions 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 sp-agglomerable around place $p3$. Let us prove that the five structural hypotheses are fulfilled:

- Structural HF-interchangeability : since $|H| = 1$ this point is satisfied.

- Structural H-independence: since $a2^\bullet \setminus \{p3\} = \emptyset$ this point is fulfilled.
- Structural divergence freeness: as the place $Lock1$ belongs to ${}^\bullet a2 \setminus a2^\bullet$, this point is also fulfilled.
- Structural quasi-persistence: Let $S = Lock1^\bullet \setminus \{a2\}$. By construction, this net satisfies the invariant $\forall m \in Acc(N, m_0), m(Mutex) + m(p2) + m(p3) + m(q2) + m(q3) = 1$. So, when $p2$ is marked, transition $b3 \in S$ is not fireable. Furthermore, by hypothesis, we have supposed that some other invariants (obtained, for instance, by the use of other mutexes) ensure that as soon as $p2$ is marked, no transition of S is fireable. So this point is fulfilled.
- Structural H-similarity: As there is a single set H this point is obviously fulfilled.

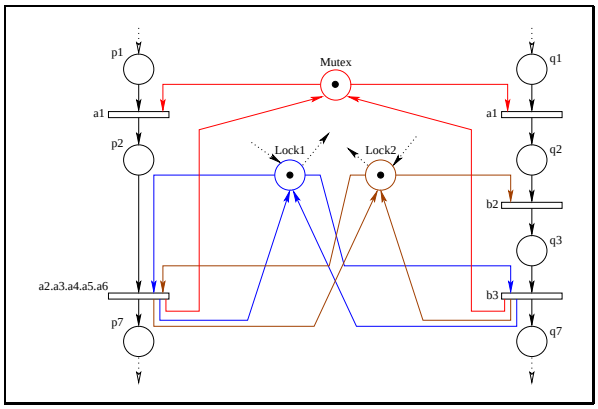

Fig. 10. The previous model after a pre-agglomeration

We obtain the net depicted in figure Fig.10. Remark that, at this point, the mutex modelled by the place `Mutex` is no more necessary to prevent deadlock.

Now, symmetrically, we perform a pre-agglomeration around the place $q3$. This leads to the model of the figure Fig.11.

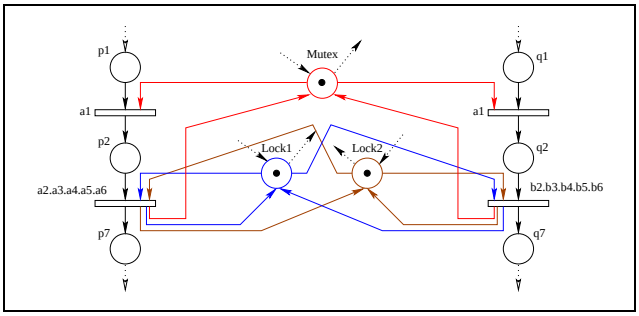

Fig. 11. The previous model after another pre-agglomeration

At last, we apply on this model (figure Fig.11) 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 $F_1 = \{a2.a3.a4.a5.a6\}$, $H_2 = \{a2\}$ and $F_2 = \{b2.b3.b4.b5.b6\}$). This reduction is interesting since no reduction with I reduced to a singleton is possible.

In the final model, the two threads operate atomically on the locks.

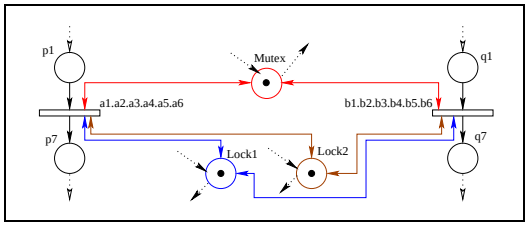

Fig. 12. The final reduced model

5.2 Updating a shared variable

Let us consider the statement `VarX := VarX + 1` which may be executed by several concurrent processes that share the common variable `VarX`. We suppose that this operation is not atomic. The next figure (Fig. 13) depicts on the left the coloured Petri net corresponding to this statement and in the right, the underlying Petri nets when the variable `VarX` takes two values (1 or 2).

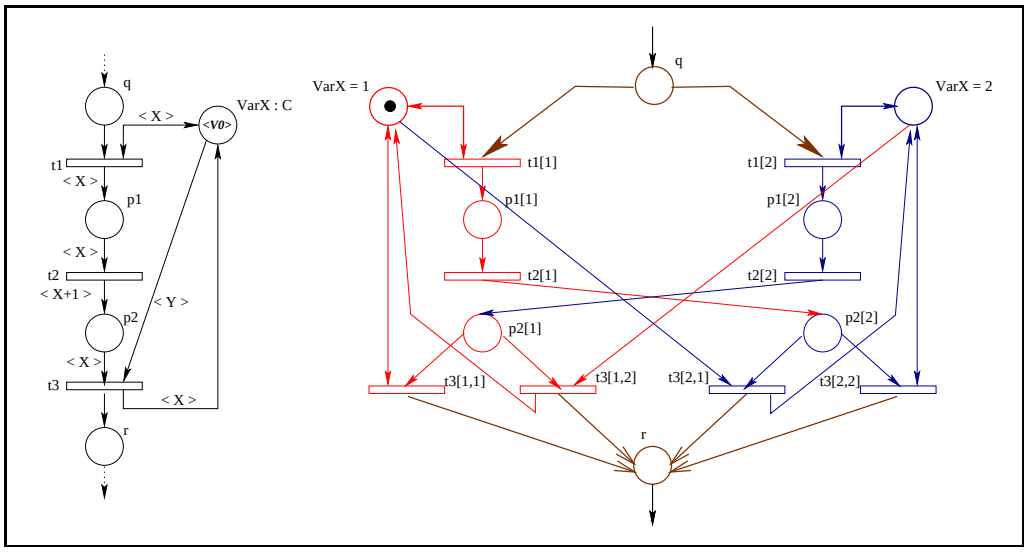

Fig. 13. Incrementation of a shared variable

Suppose that there exists a place invariants (in the underlying Petri net) $\forall m \in Reach(N, m_0), m(q)+m(p1[1])+m(p2[1])+m(p1[2])+m(p2[2])+\cdots = 1$ ensuring that there is a mutual exclusion between the places `q`, `p1[1]`, `p2[1]`, `p1[1]` and `p2[2)]`. We describe the process reduction of this net.

At first we apply two post-agglomerations "à la Berthelot" around the places `p1[1]` and `p2[2]` (also covered by our definition). We then obtain the net depicted in the figure Fig.14.

Now a post-agglomeration around the place `p2[2]` is possible. Indeed, let us note $H =$ { `t1[1]t2[1]` } and $F =$ { `t3[2,1]`, `t3[2,2]` } then

(1) the net is clearly sp-agglomerable;
(2) the HF-interchangeability is verified since $|H| = 1$;

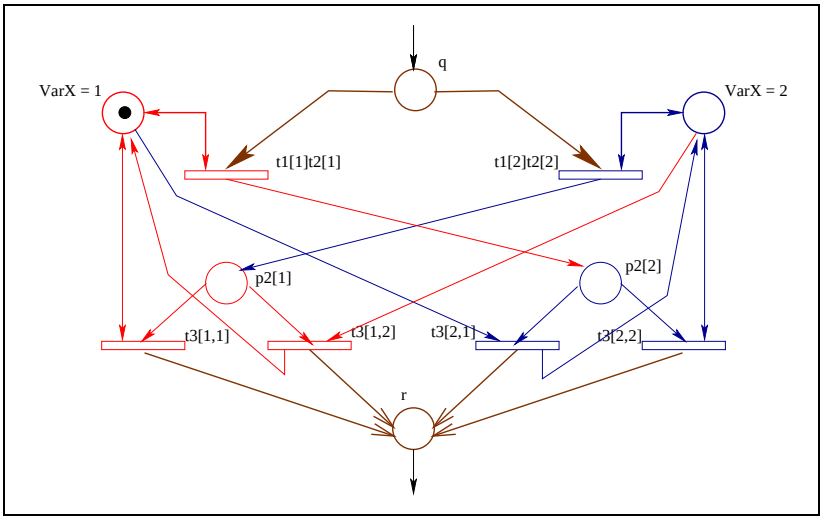

Fig. 14. Incrementation of a shared variable after two post-agglomerations

(3) the F-continuation is verified since one of the places `VarX=1` or `VarX=2` is always marked and then one of the two transitions `t3[2,1]` or `t3[2,2]` is always fireable as soon as place `p[2]` is marked.
(4) the strong F-independence is verified since the input places of `t3[2,1]` or `t3[2,2]` different from `p[2]` are `VarX=1` and `VarX=2`. The input transitions different from F of such places are `t3[1,1]`, `t3[1,2]`, `t1[1]t2[1]` and `t1[2]t2[2]`. Due to the assumed invariant described above, all these transitions are frozen by `p2[2]`.

We obtain the reduced net depicted in Fig.15. In this net a post-agglomeration

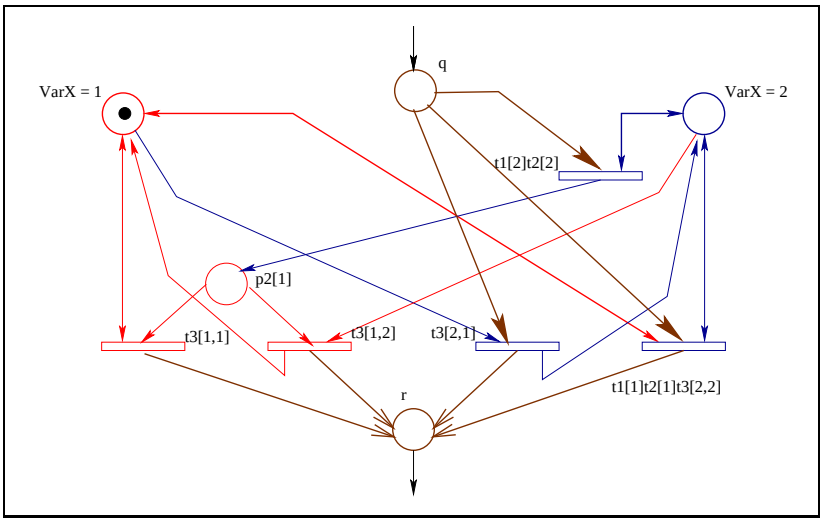

Fig. 15. The previous model after a post-agglomeration

can be performed around the place `p2[1]` for similar reasons. We obtain the model depicted in Fig16. In this model, due to the linear invariant $m(\texttt{VarX=1})+m(\texttt{VarX=2}) = 1$, the transitions `t1[1]t2[1]t3[2,2]` and `t1[2]t2[2]t3[1,1]` are dead (i.e. never fireable). It is interesting to note that these transitions cor-

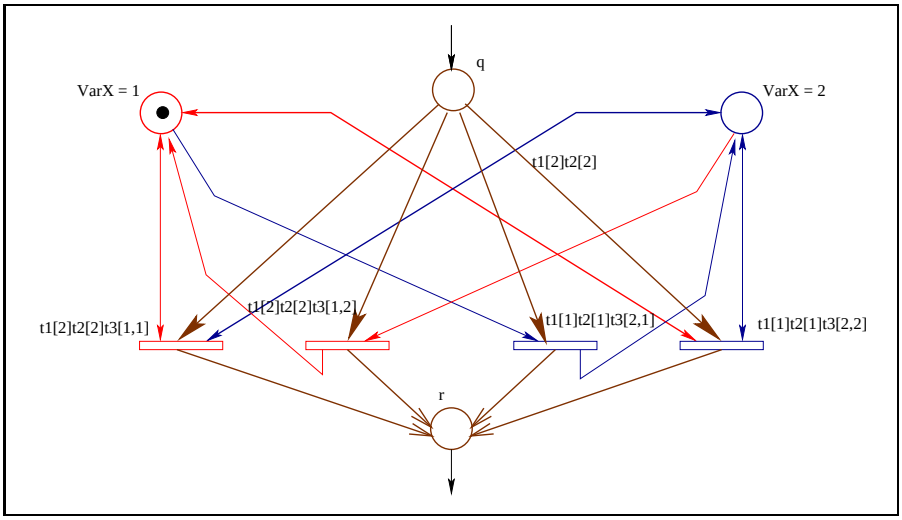

Fig. 16. The previous model after a post-agglomeration

respond to pathological races between the processes and then our reductions have shown that this dangerous behaviour cannot occur. So, we delete them and we obtain the net of of Fig.17 (with its coloured version on the right of the figure). In this last model, the incrementation is performed atomically.

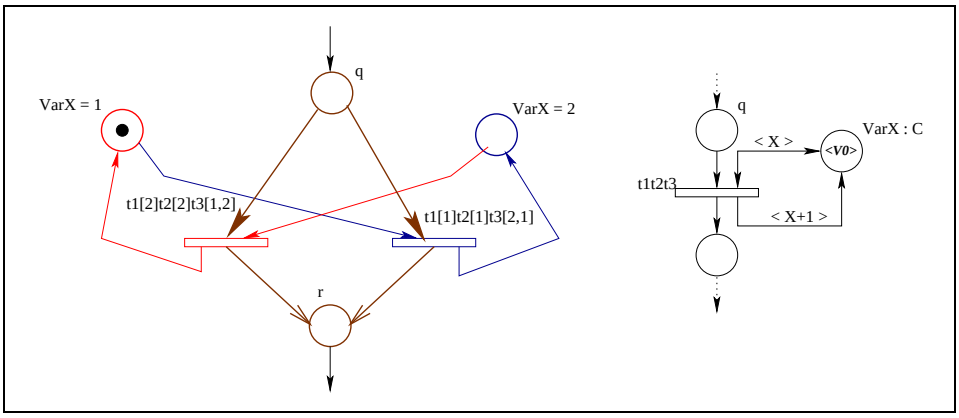

Fig. 17. The final model

Let us generalise the pattern of the figure 13. Suppose now that `VarX` is shared with other processes (i.e. other transitions than `t1` and `t3` are connected to the places `VarX`). An additional hypothesis is necessary (needed for the F-independence hypothesis): a process that modifies the value of this variable must be blocked as soon as one of the place `p1` or `p2` is marked (an underway modification of the variable must be achieved before the access to this variable is released). We emphasise that this additional condition corresponds to standard guidelines of concurrent programming (e.g. using synchronised method in Java or protected object in Ada). This condition will then induce additional invariants in the Petri net model which will be used in our algebraical conditions. Once again this shows the accurateness of our reductions.

6 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 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 linear programming problems. For instance, the structural reductions defined in [Ber83, Ber85] may be viewed as specialisations of our reductions.

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 (being a semantic for an high level model) cover more patterns in more contexts than reductions directly defined for the high-level model.

The perspectives of this work are threefold:

- Complex modelling requires coloured Petri nets rather than ordinary ones. Since the former one is an abbreviation of the latter one, we could apply our method on coloured nets. However in order to apply our method, we need to fix the parameters of the model which is unsatisfactory w.r.t. the verification point of view. Thus we plan to directly define reductions for *parametrised* coloured nets which would generalise those defined in [Had91].
- We observe that the specification of the behavioural reductions is not specific to the Petri nets model but require a weak diamond behavioural property i.e. $m[t.t'\rangle m' \wedge m[t'.t\rangle m" \Rightarrow m' = m"$. Therefore we are comparing our reductions to reductions proposed for other formalisms (like those in TLA [CL98]).
- In practice the verification of concurrent programs is often related to detection of particular bad behaviours associated to pathologic race conditions between the threads [BR01, FF01, BLR02, SC03] or to atomicity violation [FF04]. In such cases, our application conditions could be considerably relaxed. We are currently investigating such a direction.

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.

[BLR02] Chandrasekhar Boyapati, Robert Lee, and Martin Rinard. Ownership types for safe programming: preventing data races and deadlocks. In *Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 211–230. ACM Press, 2002.

[BR01] Chandrasekhar Boyapati and Martin Rinard. A parameterized type system for race-free java programs. *SIGPLAN Not.*, 36(11):56–69, 2001.

[BRV80] G. Berthelot, G. Roucairol, and R. Valk. Reduction of nets and parallel programs. In Brauer, W., editor, *Lecture Notes in Computer Science: Net Theory and Applications, Proc. of the Advanced Course on General Net Theory of Processes and Systems, Hamburg, 1979*, volume 84, pages 277–290, Berlin, Heidelberg, New York, 1980. Springer-Verlag.

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

[Cor98] James C. Corbett. Constructing compact models of concurrent java programs. In *Proceedings of ACM SIGSOFT international symposium on Software testing and analysis*, pages 1–10. ACM Press, 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. *Lecture Notes in Computer Science; 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.

[DV90] Rocco De Nicola and Frits Vaandrager. Action versus state based logics for transition systems. In *Proceedings of the LITP spring school on theoretical computer science on Semantics of systems of concurrent processes*, pages 407–419. Springer-Verlag New York, Inc., 1990.

[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 (CHARME'01)*, volume 2144 of *Lecture Notes in Computer Science*, pages 310–324. Springer-Verlag, 2001.

[FF01] Cormac Flanagan and Stephen N. Freund. Detecting race conditions in large programs. In *Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering*, pages 90–96. ACM Press, 2001.

[FF04] Cormac Flanagan and Stephen N Freund. Atomizer: a dynamic atomicity checker for multithreaded programs. In *Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 256–267. ACM Press, 2004.

[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 *FTfJP 03: 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, *Proceedings of the Eighth International Conference on Computer Aided Verification CAV*, volume 1102, pages 311–322, New Brunswick, NJ, USA, / 1996. Springer Verlag.

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

[Had87] S. Haddad. *Une catégorie régulière de réseau de Petri de haut niveau : définition, propriétés et réductions. Application à la validation de systèmes distribués.* PhD thesis, Université Pierre et Marie Curie, Paris, 1987.

[Had91] S. Haddad. A reduction theory for colored nets. In Jensen and Rozenberg, editors, *High-level Petri Nets, Theory and Application*, LNCS, pages 399–425. Springer-Verlag, 1991.

[Hol03] G.J. Holzmann. *The Spin Model Checker, Primer and Reference Manual*. Addison-Wesley, Reading, Massachusetts, 2003.

[HPP04] S. Haddad and J.F. Pradat-Peyre. Efficient reductions for LTL formulae verification. Technical report, CEDRIC, CNAM, Paris, http://cedric/AfficheArticle.php?id=634, 2004.

[HRD04] John Hatcliff, Robby, and Matthew B. Dwyer. Verifying atomicity specifications for concurrent object-oriented software using model-checking. In *Proceedings of the International Conference on Verification, Model Checking and Abstract Interpretation*, 2004.

[Jen91] K. Jensen. Coloured Petri nets : A high level language for system design and analysis. In Jensen and Rozenberg, editors, *High-level Petri Nets, Theory and Application*, pages 44–119. Springer-Verlag, 1991.

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

[Lom77] D. B. Lomet. Process structuring, synchronization, and recovery using atomic actions. In *Proceedings of an ACM conference on Language design for reliable software*, pages 128–137, 1977.

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

[Mis03] Jayadev Misra. A reduction theorem for concurrent object-oriented programs. pages 69–92, 2003.

[Mur89] T. Murata. Petri nets : properties, analysis and applications. In *proceedings of the IEEE Vol 77*, number 4, pages 39–50, January 1989.

[Pnu81] A. Pnueli. The temporal semantics of concurrent programs. In *Theoretical Computer Science*, number 13, pages 45–60, 1981.

[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.

[QRR04] Shaz Qadeer, Sriram K. Rajamani, and Jakob Rehof. Summarizing procedures in concurrent programs. In *Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 245–255. ACM Press, 2004.

[Rei83] W. Reisig. *EATCS-An Introduction to Petri Nets*. Springer-Verlag, 1983.

[SC03] Scott D. Stoller and Ernie Cohen. Optimistic synchronization-based state-space reduction. In H. Garavel and J. Hatcliff, editors, *Proceedings of the 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 2619 of *Lecture Notes in Computer Science*, pages 489–504. Springer-Verlag, April 2003.

A Formal definition of Petri nets

DEFINITION **A.1** *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 a integer vector indexed by P is the initial marking.*

DEFINITION **A.2** *Let (N, m_0) be a marked net then:*

- *$t \in T$ is firable from m a marking (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 $W = W^+ - W^-$,*

DEFINITION **A.3** *Let (N, m_0) be a marked net then:*

- *$s \in T^*$ is firable from m a marking and leads to m' (also denoted by $m[s\rangle$ and $m[s\rangle m'$) iff*

(1) either $s = \lambda$ and $m' = m$
(2) or $s = s_1.t$ with $t \in T$ and $\exists m_1\ m[s_1\rangle m_1$ and $m_1[t\rangle m'$

- *$s \in T^\infty$ is firable from m a marking (also denoted $m[s\rangle$) iff for every finite prefix s_1 of s, $m[s_1\rangle$.*

DEFINITION **A.4** *Let (N, m_0) be a marked net then:*

- *$Reach(N, m_0) = \{m | \exists s \in T^*\ m_0[s\rangle m\}$ is the set of reachable markings,*
- *m is a dead marking if $\forall t \in T\ NOT(m[t\rangle$,*
- *(N, m_0) is live iff $\forall m \in Reach(N, m_0)\, \forall t \in T\, \exists s \in T^*\ m[s.t\rangle$,*
- *$L(N, m_0) = \{s \in T^* | m_0[s\rangle\}$ is the language of finite sequences,*
- *$L^{Max}(N, m_0) = \{s \in T^* | \exists m\, dead\, marking\, m_0[s\rangle m\}$ is the language of finite maximal sequences,*
- *$L^\infty(N, m_0) = \{s \in T^\infty | m_0[s\rangle\}$ is the language of infinite sequences,*

DEFINITION **A.5** *The incidence matrices W, W^- and W^+ are extended to matrices indexed by $P \times T^*$ by the following recursive definition:*

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

- $W^+(p,s) = W(p,s) + W^-(p,s)$

PROPOSITION 11 *Let* (N, m_0) *be a marked net then:*
$\forall s \in T^*,\ m[s\rangle m' \iff \forall p \in P,\ m(p) \geq W^-(p,s)$ *and* $m'(p) = m(p) + W(p,s)$

B Proofs of the different theorems and propositions

B.1 Proofs related to the behavioural Pre-agglomeration

In order to deal with infinite sequences we need to decompose them according to their ultimate behavior w.r.t. the Γ_i functions. We introduce a notation to characterize this behavior.

DEFINITION **B.1 (i^{th} degree of a sequence)** *The degree w.r.t. $i \in I$ of a sequence $s = t_1 \dots t_n \dots \in T^\infty$ of a p-agglomerable net (denoted by $d_i^\circ(s)$) is defined by:*

$$d_i^\circ(s) = \liminf_{k\to\infty}(\Gamma_i(t_1 \dots t_k)) \stackrel{def}{=} \lim_{k\to\infty} Inf(\Gamma_i(t_1 \dots t_{k'}) \mid k' \geq k)$$

Since Γ_i is lower bounded by 0, this degree is well defined but may be finite or infinite and one has the following decompositions:

- Let $d_i^\circ(s) = d$, then there exists a decomposition $s = s_1.h_1 \dots s_d.h_d.s_{d+1}.s_{d+2} \dots s_{d+k} \dots$ with $\forall n \leq d, h_n \in H_i$, $\forall n$, $\Gamma_i(s_n) = 0$ and $\forall s'$ prefix of s_n, $\Gamma_i(s') \geq 0$.
- Let $d_i^\circ(s) = \infty$, then there exists a decomposition $s = s_1.h_1 \dots s_n.h_n \dots$ with $\forall n\ h_n \in H_i$, $\Gamma_i(s_n) = 0$ and $\forall s'$ prefix of s_n, $\Gamma_i(s') \geq 0$.

W first prove that if a p-agglomerable net satisfies the H-independent hypothesis, then each fireable sequence can be reordered into a simulable fireable sequence having the same projection on $T_0 \cup F$ as the original one. This result is stated with the help of two propositions : one for the case of finite sequences (proposition 12) and one for the case of infinite sequences (proposition 13).

PROPOSITION 12 (H-FINITE INDEPENDENCE) *Let (N, m_0) be a p-agglomerable net which is H-independent. Let $J \subseteq I$, $H_J = \cup_{j\in J} H_j$ and $F_J = \cup_{j\in J} F_j$. Then for every sequence $s \in T^*$ such that $m_0[s_0\rangle m[s\rangle m'$ with $\forall j \in J\ \forall s'$ prefix of s, $\Gamma_j(s') \geq 0$ there exists a permutation of s, $\widehat{s}_{/J} = s_{\bowtie}.s_{\triangleleft}$ such that :*

(1) $m[\widehat{s}_{/J}\rangle m'$
(2) $\Pi_{T\setminus H_J}(s_{\bowtie}) = \Pi_{T\setminus H_J}(s)$ and $s_{\triangleleft} \in H_J^*$.
(3) $s_{\bowtie}$ is J-simulateable

We will denote by $\widehat{s}_{/J} = s_{\bowtie}.s_{\triangleleft}$ any sequence fulfilling the above requirements w.r.t. s and J, and when $J = I$ we simply denote $\widehat{s} = s_{\bowtie}.s_{\triangleleft}$.

Proof of proposition 12, page 33 We prove by induction on the length of $|s|_{H_J \cup F_J}$ that there exists at least one sequence $\widehat{s}_{/J}$. If $s \in (T \setminus (H_J \cup F_J))^*$

then the sequence $\widehat{s}_{/J} = s_{\bowtie} = s$ fulfills the conditions of the proposition.

Otherwise, the sequence s can be written $s = s'.t.s''$ with $s'' \in (T \setminus (H_J \cup F_J))^*$ and $t \in H_J \cup F_J$. There are two cases:

- $\exists j \in J,\ t \in H_j$
  As $|s''|_{F_j} = 0$, the H-independence hypothesis implies that $m[s'.s''.t\rangle m'$. Using the inductive hypothesis, there exists a sequence $\widehat{s'.s''}_{/J}$ and by construction $\widehat{s'.s''}_{/J}.t$ fulfils the conditions of the proposition w.r.t. s and J.
- $\exists j \in J,\ t \in F_j$
  By hypothesis, $\Gamma_j(s'.t) \geq 0$. So $\Gamma_j(s') > 0$. Let us pick the **longest** prefix s_1 of s' such that $\Gamma_j(s_1) = 0$. By definition of s_1, $s' = s_1.h.s_2$ with $h \in H_j$ and such that $\forall s^*$ prefix of s_2, $\Gamma_j(s^*) \geq 0$. Since the net is H-independent, $m[s_1.s_2.h.t.s''\rangle m'$.
  Again due to the definition of s_1, the inductive hypothesis applies to $s_1.s_2$. Thus $\widehat{s_1.s_2}_{/J} = s_{\bowtie}.s_{\triangleleft}$ with $s_{\triangleleft} \in H_J^*$ and $m[s_{\bowtie}.s_{\triangleleft}.h.t.s''\rangle m'$.
  Since $|s''|_{F_J} = 0$, the prefixes of the sequence $h.t.s''$ fulfills the Γ_j condition of H-independence for all $j \in J$. Thus we apply $|s_{\triangleleft}|$ times the H-independence hypothesis on the sequence $h.t.s''$ leading to $m[s_{\bowtie}.h.t.s''.s_{\triangleleft}\rangle m'$.
  By construction, $s_{\bowtie}.h.t.s''.s_{\triangleleft}$ fulfils the conditions of the proposition w.r.t. s and J.

PROPOSITION 13 (H-INFINITE INDEPENDENCE) *Let (N, m_0) be a p-agglomerable net which is H-independent. Then for every $s \in L^\infty(N, m_0)$ there exists a permutation of s, $\widehat{s}$ such that*

(1) $\forall s' \in Pref(\widehat{s}),\ m_0[s'\rangle$;
(2) $\Pi_{T\setminus H}(\widehat{s}) = \Pi_{T\setminus H}(s)$
(3) $\widehat{s} = s^0_{\bowtie}.s^0_{\triangleleft}.s^1_{\bowtie}.s^1_{\triangleleft} \ldots s^k_{\bowtie}.s^k_{\triangleleft} \ldots$ *where* $\forall k \geq 0$, $s^k_{\bowtie}$ *is a simulateable sequence and* $s^k_{\triangleleft} \in H^*$

Proof of the proposition 13, page 34 We will prove the existence of this sequence by induction on $J \subset I$. We suppose that for there is a permutation $\widehat{s}_J$ of s such that: $\forall s' \in Pref(\widehat{s}_J)$, $m_0[s'\rangle$, $\Pi_{T\setminus H_J}(\widehat{s}_J) = \Pi_{T\setminus H_J}(s)$ and $\widehat{s}_J = s^0_{\bowtie}.s^0_{\triangleleft}.s^1_{\bowtie}.s^1_{\triangleleft} \ldots s^k_{\bowtie}.s^k_{\triangleleft} \ldots$ where $\forall k \geq 0$, $s^k_{\bowtie}$ is a J-simulateable sequence and $s^k_{\triangleleft} \in H_J^*$. The basis case $J = \emptyset$ is straightforward. Let us suppose that J is strictly included in I and pick some $i \in I \setminus J$. We distinguish two cases:

(1) $d_i^\circ(s) = d$ and then $d_i^\circ(\widehat{s}_{/J}) = d$. Thus there is a decomposition of $\widehat{s}_{/J} = s_1.h_1.s_2.h_2.s_3.\ldots.h_d.s_{d+1}.s_{d+2}.s_{d+3}\ldots$ with $\forall k \geq 0,\ \Gamma_i(s_k) = 0$ and $\forall s'$ prefix de s_k, $\Gamma_i(s') \geq 0$ and $\{h_1, \ldots, h_d\} \subset H_i$. We refine this decomposition by taking into account the first decomposition of $\widehat{s}_{/J}$ giving: $\forall k \geq 0,\ s_k = s^{k,1}_{\bowtie}.s^{k,1}_{\triangleleft}.\ldots.s^{k,n_k}_{\bowtie}.s^{k,n_k}_{\triangleleft}$
We apply the previous proposition with $J = \{i\}$ on every s_k. This leads

to a new infinite sequence:
$\overline{s} = \overline{s}^{1,1}_{\bowtie}.s^{1,1}_{\lhd}.\ldots.\overline{s}^{1,n_1}_{\bowtie}.s^{1,n_1}_{\lhd}.h_1.\overline{s}^{2,1}_{\bowtie}.s^{2,1}_{\lhd}.\ldots.\overline{s}^{2,n_2}_{\bowtie}.s^{2,n_2}_{\lhd}.h_2.\ldots.h_d.\overline{s}^{d+1,1}_{\bowtie}.s^{d+1,1}_{\lhd}.\ldots$
with $\overline{s}^{k,1}_{\bowtie}.\ldots.\overline{s}^{k,n_k}_{\bowtie}$ a permutation of $s^{k,1}_{\bowtie}.\ldots.s^{k,n_k}_{\bowtie}$ which is $\{i\}$-simulateable)
So the decomposition of $\overline{s}$ fulfills the induction hypothesis for $J \cup \{i\}$

(2) $d^{\circ}_i(s) = \infty$ and then $d^{\circ}_i(\widehat{s}_{/J}) = \infty$. Thus there is a decomposition of $\widehat{s}_{/J} = s_1.h_1.s_2.h_2.s_3.\ldots.h_k.s_{k+1}.\ldots$ with $\forall k \geq 0,\ \Gamma_i(s_k) = 0$ and $\forall s'$ prefix de s_k, $\Gamma_i(s') \geq 0$ and $\{h_1, \ldots, h_k, \ldots\} \subset H_i$. So the proof of this case is similar to the proof of the first case.

We decompose the proof of the theorem 2 into different parts using progressively the five behavioral hypotheses.

LEMMA 1 *Let (N, m_0) be a p-agglomerable net which is H-independent and inter-HF-equivalent. Then*

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

Proof of lemma 1 We prove that given a reachable marking m of the reduced net $(m_0[s_{0_r}\rangle_r m)$, for any transition $t_r \in T_r$ there exists a fireable sequence from m that includes or enables the transition t_r. Using the theorem 1, we know that $m_0[s_0\rangle m$ with $s_0 = \phi(s_{0_r})$.

- If $t_r \in T_0$, as N is live, there exists a sequence s such that $m[s.t_r\rangle$. Since $t_r \notin H$, the definition of $\widehat{s.t} = s_{\bowtie}.s_{\lhd}$ implies that $t_r \in s_{\bowtie}$. As $s_{\bowtie}$ is a simulateable sequence, we have $m[\phi^{-1}(s_{\bowtie})\rangle_r$ and the proposition follows.
- If $t_r = hf$, $h \in H_i$, $f \in F_i$ we distinguish two cases depending on which inter-HF-equivalence hypothesis is fulfilled:

(1) The net fulfills point 1) of the inter-HF-equivalence hypothesis.
In this case, we choose $t = f$ (remember that $t_r = hf$). As N is live, there exists at least one firing sequence from the marking m that enables transition t. Let s be the shortest of these sequences $(m[s.t\rangle)$ and $\widehat{s.t} = s_{\bowtie}.s_{\lhd}$. As $t \notin H$, the minimality of $s.t$ implies that $\widehat{s.t} = s_{\bowtie}$ and that f is the last transition of $s_{\bowtie}$
Since $s_{\bowtie}$ is simulateable, $s_{\bowtie} = s1_{\bowtie}.h'.f$ with $s1_{\bowtie}$ a simulateable sequence and $h' \in H_i$. The inter-HF -equivalence hypothesis implies that $m[s1_{\bowtie}.h.f\rangle$ and then $m[\phi^{-1}(\widehat{s1_{\bowtie}}).t_r\rangle_r$.

(2) The net fulfills point 2) of the inter-HF-equivalence hypothesis.
In this case, we choose $t = h$ ($t_r = hf$). Using a same argument there exists a minimal sequence s such that $m[s.t\rangle$. Again, the minimality of s implies that in the decomposition $\widehat{s} = s_{\bowtie}.s_{\lhd}$, $s_{\lhd}$ is empty. Otherwise, $s_{\lhd} = h'.s'$ and the H-independence hypothesis would imply that $m[s_{\bowtie}.s'.h.h'\rangle$ which would contradict the minimality of s.
As N is live, there exists a minimal sequence s' such that $m[s_{\bowtie}.h.s'.f'\rangle$, with $f' \in F_i$. Since s' is minimal, $s' \in (T \setminus F_i)^*$. The H-Independence hy-

pothesis implies that $m[s_{\bowtie}.s'.h.f'\rangle$ and the Inter-HF Equivalence hypothesis implies that $m[s_{\bowtie}.s'.h.f\rangle$. We write $\widehat{s'} = s'_{\bowtie}.s'_{\lhd}$. As $m[s_{\bowtie}.s'_{\bowtie}.s'_{\lhd}.h.f\rangle$, because $\forall j \in I, \Gamma_j(h.f) = 0$, and because $s'_{\lhd} \in T_H^*$, the H-Independence hypothesis implies that $m[s_{\bowtie}.s'_{\bowtie}.h.f\rangle$ (we delay one by one the transitions of $s'_{\lhd}$). So, we obtain that $m[\phi^{-1}(s1_{\bowtie}).\phi^{-1}(s'_{\bowtie}).t_r\rangle_r$.

LEMMA 2 *Let (N, m_0) be a p-agglomerable net which is H-independent and divergent-free. Then:*

(1) $\Pi_{T_0 \cup T_F}(\phi(L^{\infty}(N_r, m_0))) = \Pi_{T_0 \cup T_F}(L^{\infty}(N, m_0))$
(2) $\Pi_{T_0 \cup T_F}(\phi(L^{max}(N_r, m_0))) \subset \Pi_{T_0 \cup T_F}(L^{max}(N, m_0))$

Proof of lemma 2 We successively prove each part of the lemma.

- **First claim of the lemma**
  From Theorem 1 $\phi(L^{\infty}(N_r, m_{0_r}) \subseteq L^{\infty}(N, m_0)$. So it is sufficient to prove that $\Pi_{T_0 \cup T_F}(L^{\infty}(N, m_0)) \subseteq \Pi_{T_0 \cup T_F}(\phi(L^{\infty}(N_r, m_0)))$

  Let s be an infinite sequence. Using proposition 13, we obtain a new infinite sequence $\widehat{s} = s^0_{\bowtie}.s^0_{\lhd}.s^1_{\bowtie}.s^1_{\lhd} \ldots s^k_{\bowtie}.s^k_{\lhd} \ldots$ such that $\Pi_{T_0 \cup T_F}(\widehat{s}) = \Pi_{T_0 \cup T_F}(s)$. Using the H-independence starting from the fireable sequence $s^0_{\bowtie}.s^0_{\lhd}.s^1_{\bowtie}.s^1_{\lhd} \ldots s^k_{\bowtie}.s^k_{\lhd}$, one obtains another firing sequence $s^0_{\bowtie}.s^1_{\bowtie}. \ldots s^k_{\bowtie}.s^0_{\lhd}.s^1_{\lhd} \ldots s^k_{\lhd}$.

  Thus $\overline{s}$ the simulateable sequence defined by $\overline{s} = s^0_{\bowtie}.s^1_{\bowtie}. \ldots s^k_{\bowtie}. \ldots$ is a sequence of (N, m_0). Since the net is divergent-free, $|s|_{T_0 \cup F} = \infty$ and the sequence $\overline{s}$ is infinite.

- **Second claim of the lemma**
  Let s_r such that $m_{0_r}[s_r\rangle_r m_r$ with m_r a dead marking of the reduced net. We know that $m_0[\phi(s_r)\rangle m_r$. Let s' a non empty maximal or infinite fireable sequence from m_r (i.e. $m_r[s'\rangle$). Let us suppose that $|s'|_{T_0 \cup F} \neq 0$ and denote s'' the smallest prefix of s' such that $|s''|_{T_0 \cup F} \neq 0$. From proposition 12, we have a new sequence $\widehat{s''} = s''_{\bowtie}.s''_{\lhd}$ with $s''_{\bowtie}$ a non empty simulateable sequence. Then $m_r[\phi^{-1}(s''_{\bowtie})\rangle_r$ which is impossible since m_r is a dead marking. Thus $|s'|_{T_0 \cup F} = 0$. Due to the divergence-freeness of the net, $\phi(s_r).s'$ is a finite maximal sequence of (N, m_0) with $\Pi_{T_0 \cup F}(\phi(s_r)) = \Pi_{T_0 \cup F}(s_r.s')$.

LEMMA 3 *Let (N, m_0) be a p-agglomerable net which is H-independent, quasi-persistent and H-similar. Then:*

$$(N_r, m_{0_r}) \textit{ live} \Longrightarrow (N, m_0) \textit{ live}$$

Proof of lemma 3 Let us pick a reachable marking m_1 in the original net (i.e. $m_0[s_0\rangle m_1$)and a transition $t \in T$. We will prove that t is necessarily fireable in the original net from m_1. Let $k = \sum_{i \in I} \Gamma_i(s_0)$, we proceed by induction on k.

- $k = 0$; i.e. s_0 is balanced and then $\widehat{s_0}$ is simulateable; so, $m_0[\phi^{-1}(\widehat{s_0})\rangle_r m_1$.
  Let t_r be the transition defined by :
  · $t_r = t$ if $t \in T_0$;
  · $t_r = tf$ if $t \in H_i$ and f some transition of F_i,
  · $t_r = ht$ if $t \in F_i$ and h some transition of H_i,
  Since (N_r, m_0) is live, there exists a sequence $s \in T_r^*$ such that $m_1[s.t_r\rangle_r$. So, $m_1[\phi(s.t_r)\rangle$ and by construction of t_r, $|\phi(s.t_r)|_t \neq 0$.
- $k > 0$; i.e. $\widehat{s_0} = s_{0\bowtie}.h_{i_1}.h_{i_2}\ldots h_{i_k}$ with $h_{i_j} \in H_{i_j}$.
  Let m the marking defined by $m_0[s_{0\bowtie}\rangle m$. As $s_{0\bowtie}$ is simulateable and as (N_r, m_0) is live, let us define s as a shortest sequence such that $m[s.hf\rangle_r$, for some $i \in I$, $h \in H_i$ and $f \in F_i$. Obviously $s \in T_0^*$. We also know that in the original net, $m[s.h.f\rangle$.
  The H-similarity hypothesis implies that $\exists s' \in T_0^*, f_{i_1} \in F_{i_1}$ such that $m[s'.h_{i_1}.f_{i_1}\rangle$. So, we have, $m[s'.h_{i_1}.f_{i_1}\rangle$ and also $m[h_{i_1}\rangle$.
  Since $s' \in T_0^*$, the quasi-persistence hypothesis implies that $\exists s'' \in T_0^*$ such that $m[h_{i_1}.s''\rangle$ with $W(s'') \geq W(s')$. So, $m[h_{i_1}.s''.f_{i_1}\rangle$.
  Let m' the marking defined by $m[h_{i_1}\rangle m'$. At m' we can fire h_{i_2} and $s''.f_{i_1}$. Since $s''.f_{i_1} \in (T_0 \cup T_F)^*$, using again the quasi-persistence hypothesis, $\exists s''' \in (T_0 \cup T_F)^*$ such that $m'[s'''.h_{i_2}\rangle$ and $\Pi_F(s''') = f_{i_1}$.
  By iteration of this argument, $\exists s_k \in (T_0 \cup T_F)^*$ such that $m[h_{i_1}.h_{i_2}\ldots h_{i_k}\rangle m_1[s_k\rangle m_2$ with $\Pi_F(s_k) = f_{i_1}$. Thus m_2 is reachable from m_1 and by construction $\sum_{i\in I}\Gamma_i(s_0.s_k) = k-1$. Using the induction hypothesis, the theorem follows.

LEMMA 4 *Let (N, m_0) be a p-agglomerable net which is H-independent, strongly quasi-persistent and H-similar. Then:*

$$\Pi_{T_0\cup T_F}(L^{max}(N, m_0)) \subseteq \Pi_{T_0\cup T_F}(\Phi(L^{max}(N_r, m_{0r})))$$

Proof of lemma 4 Let s be a finite maximal sequence ($m_0[s\rangle m_d$ with m_d a dead marking). Let then $\widehat{s} = s_{\bowtie}.s_{\lhd}$ and m be the marking defined by $m_0[s_{\bowtie}\rangle m$. Let us suppose that there exists a non empty sequence $s_r \in T_r^*$ with $m[s_r\rangle_r m'$; so we have $m[\phi(s_r)\rangle m'$ in the original net. We distinguish two cases :

(1) $\phi(s_r) = t.s'$, $t \in T_0$; We have $M[s_{\lhd}\rangle$ and $m[t\rangle$. Applying $|s_{\lhd}|$ times the strong quasi-persistence hypothesis there exists a non empty sequence s'' such that $W(s'') \geq W(t)$, and such that $m[s_{\lhd}\rangle m_d[s''\rangle$. which is impossible.
(2) $\phi(s_r) = h_i.f_i.s'$, for some $i \in I$, $h_i \in H_i$ and $f_i \in F_i$. We have $m[s_{\lhd}\rangle$ and $m[h_i.f_i\rangle$. If $s_{\lhd}$ is empty then $m_d[h_i.f_i\rangle$ which is impossible. Thus we note $s_{\lhd} = h_{i_1}\ldots h_{i_k}$. The H-similarity hypothesis implies that $\exists s' \in T_0^*, f_{i_1} \in F_{i_1}$ such that $m[s'.h_{i_1}.f_{i_1}\rangle$. So, we have, $m[s'.h_{i_1}.f_{i_1}\rangle$ and also $m[h_{i_1}\rangle$.
  Since $s' \in T_0^*$, the quasi-persistence hypothesis implies that $\exists s'' \in T_0^*$ such that $m[h_{i_1}.s''\rangle$ with $W(s'') \geq W(s')$. So, $m[h_{i_1}.s''.f_{i_1}\rangle$.
  Let m' the marking defined by $m[h_{i_1}\rangle m'$. At m' we can fire h_{i_2} and $s''.f_{i_1}$. Since $s''.f_{i_1} \in (T_0 \cup T_F)^*$, using again the quasi-persistence hy-

pothesis, $\exists s''' \in (T_0 \cup T_F)^*$ such that $m'[s'''.h_{i_2}\rangle$ and $\Pi_F(s''') = f_{i_1}$.

By iteration of this argument, $\exists s_k \in (T_0 \cup T_F)^*$ such that $m[h_{i_1}.h_{i_2} \dots h_{i_k}\rangle m_d[s_k\rangle m_2$ with $\Pi_F(s_k) = f_{i_1}$ which is impossible.

Thus m is a deadlock in (N_r, m_{0r}) and $\Pi_{T_0 \cup F}(s_{\bowtie}) = \Pi_{T_0 \cup F}(s)$.

Proof of theorem 2, page 12 A direct consequence of previous lemmas.

The following table relates the hypotheses to the preservation of the properties.

Original		Reduced	H-independence	inter-HF-equivalence	divergence-freeness	quasi-persistence	H-similarity
live	$\Longrightarrow$	live	•	•			
live	$\Longleftarrow$	live	•			•	•
$L^{\inf}$	$\supseteq$	$L^{\inf}$					
$L^{\inf}$	$\subseteq$	$L^{\inf}$	•		•		
L^{max}	$\supseteq$	$L^{\max}$	•		•		
L^{max}	$\subseteq$	$L^{\max}$	•			strong	•

As for the pre-agglomeration we first 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 14 (F-FINITE INDEPENDENCE) *Let (N, m_0) be a p-agglomerable net which is F-independent. Then for all sequence $s \in T^*$ 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, $\widehat{s} = s_{\bowtie}.s_{\triangleleft}$, such that :*

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

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

Proof of proposition 14, page 39 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 of the proposition w.r.t. s.
- $|s|_F > 0$: The hypothesis on the prefixes of s implies that 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. Furthermore a straightforward checking shows that the prefixes of the sequence $s_1.s_2$ fulfils the hypothesis of the proposition.

  Thus by induction $\widehat{s_1.s_2}$ exists and $(s'.h.f.(s_1.s_2)_{\bowtie}).(s_1.s_2)_{\triangleleft}$ fulfils the conditions of the proposition w.r.t. s.

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

PROPOSITION 15 (F-INFINITE INDEPENDENCE) *Let (N, m_0) be a p-agglomerable which is F-independent. Then for any infinite sequence $s \in L^{\infty}(N, m_0)$ there exists a permutation of s, $\widehat{s}$ such that*

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

Proof of proposition 15, page 39 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 de 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 $\widehat{s_k} = s^k_{\bowtie}$ (recall 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 de 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 proof of the first case.

As for the pre-agglomeration we establish progressively the results claimed in theorem 3.

LEMMA 5 *Let (N, m_0) be a p-agglomerable net which is F-independent and inter-HF-equivalent. Then*

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

Proof of lemma 5

Let $m_0[s_r\rangle m$ et $t_r \in T$.

We have $m_0[\phi(s_r)\rangle m$. We distinguish three cases.

- 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 one has for all prefixes s' of s_1, $\Gamma(s') \geq 0$. 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 $s'.s'' \in (T_0 \cup H)^*$. We apply the F-independence transformation leading to the firing sequence $m[s^1_{\bowtie}.h.f.s'.s''.t_r\rangle$ and $\Gamma(s_1) > \Gamma(s^1_{\bowtie}.h.f.s'.s'')$. 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 $m[\phi^{-1}(s^1_{\bowtie}).t_r\rangle$.
- Let $t_r = hf$ with $h \in H$ and $f \in F$ and suppose that the inter-HF-equivalence 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 agglomerable one has for all prefixes s' of s, $\Gamma(s') \geq 0$. 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 inter-HF-equivalence one substitutes h to h' leading to $m[s_1.h.f\rangle$. Thus in the reduced net, $m[\phi^{-1}(s_1).t_r\rangle$.

- Let $t_r = hf$ with $h \in H$ and $f \in F$ and suppose that the inter-HF-equivalence 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 minimizing $\Gamma(s_1)$. 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 inter-HF-equivalence 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 6 *Let (N, m_0) be a p-agglomerable net which is F-independent and and F-continuable. Then*

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

Proof of lemma 6 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, $s_{\bowtie}$. 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 7 *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 of lemma 7

(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 such that $m_{0r}[s_r\rangle_r m_d$ 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 8 *Let (N,m_0) be a p-agglomerable net which is strongly F-independent and F-continuable. Then*

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

Proof of lemma 8

(1) $\Pi_{T_0\cup H}(L^{\infty}(N,m_0)) \subseteq \Pi_{T_0\cup H}(\phi(L^{\infty}(N_r,m_0)))$

Let $s \in L^{\infty}(N,m_0)$. Then from proposition 15, there exists a sequence $\widehat{s}$ such that

(a) $\forall s' \in Pref(\widehat{s})$, $m_0[s'\rangle$;

(b) $\Pi_{T_0\cup H}(\widehat{s}) = \Pi_{T_0\cup H}(s)$

(c) $\exists (s^i_{\bowtie})_{i\geq 0}$ 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 $s^n_{\lhd} \in H^*$

By (possible) insertions of empty sequences, the sequence $\widehat{s}$ may be rewritten as :

- when $d(s) = n \in \mathbb{N}$, $\widehat{s} = s^1_{\bowtie}.h_1.s^2_{\bowtie}.h_2\ldots s^n_{\bowtie}.h_n.s^{n+1}_{\bowtie}$ with $\forall i, h_i \in H$
- when $d(s) = \infty$, $\widehat{s} = s^1_{\bowtie}.h_1.s^2_{\bowtie}.h_2\ldots s^n_{\bowtie}.h_n\ldots$ with $\forall i, h_i \in H$

At first let us suppose that $d(s) = n$ is finite. Then we build by induction a simulateable infinite sequence s' with $\Pi_{T_0\cup H}(s') = \Pi_{T_0\cup H}(s)$. The induction hypothesis is the following one: there exists a 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 basis case is handled by taking $s_0 = \widehat{s}$.

Now let us look at h_{k+1}. Then $s_k{=}s'_k.h_{k+1}.s"_k$ with s'_k a balanced sequence and $s"_k$ a infinite suffix of $\widehat{s}$ s.t. $\forall s_p$ finite prefixe of $s"_k$, $\Gamma(s_p) \geq 0$. 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 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.

Now let us suppose that $d(s)$ is infinite. We proceed in the same way

as for the finite case. Thus we produce an infinite set of sequences $s_n = s'_n.h_{n+1}.s"_n$ s.t. s'_n is a strict prefix of s'_{n+1}. The infinite sequence defined by its infinite set of prefixes s'_n is the balanced sequence we look for.

(2) $\Pi_{T_0 \cup H}(L^{\infty}(N, m_0)) \supseteq \Pi_{T_0 \cup H}(\phi(L^{\infty}(N_r, m_0)))$
Follows straightforwardly from theorem 1.

Proof of theorem 3, page 13 A direct consequence of previous lemmas.

B.2.1 Synthesis

Original		Reduced	$\mathcal{F}$-Independence	Inter-HF Equivalence	$\mathcal{F}$-Continuation
live	$\Longrightarrow$	live	•	•	
live	$\Longleftarrow$	live	•		•
$L^{\inf}$	$\supseteq$	$L^{\inf}$			
$L^{\inf}$	$\subseteq$	$L^{\inf}$	*strong*		•
L^{max}	$\supseteq$	$L^{\max}$	•		•
L^{max}	$\subseteq$	$L^{\max}$	•		•

Proof of the proposition 3, page 17 (H-independence) Let us suppose that there is a reachable marking m and a sequence s such that $\forall s' \in Pref(s)\ \Gamma_i(s') \geq 0$ and $m[h.s\rangle$ with $h \in H_i$.

We claim that no transition of $(H_i^\bullet \setminus \{p_i\})^\bullet$ occurs in s.

At first, pick up $t \in (H_i^\bullet \setminus \{p_i\})^\bullet \setminus F_i$, s' a prefix of s and note $m[h.s'\rangle m'$. Due to the condition on s, $\Gamma_i(s') \geq 0$, thus $m'(p_i) = m(p_i) + W^+(p_i, h) + \Gamma_i(s') > 0$. As p_i is marked, t is not fireable from m' due to the hypothesis a).

Then suppose that $f \in (H_i^\bullet \setminus \{p_i\})^\bullet \cap F_i$ occurs in s. In this case, the condition of the hypothesis b) holds. Let us pick $s'.f$ the appropriate prefix of s. Due to the condition on s, $\Gamma_i(s') = \Gamma_i(s'.f) + 1 \geq 1$. So there is some $h' \in H_i$ occuring in s' i.e. there is a prefix $s".h'$ of s s.t. $m[h.s"\rangle m'[h'\rangle$ As $m'(p_i) = m(p_i) + W^+(p_i, h) + \Gamma_i(s") > 0$, p_i is marked and freezes h' which leads to a contradiction.

We prove now that $m[s\rangle$ by induction on the length of the prefixes of s. Let us pick a prefix $s_0.t$ of s and suppose that we have proven that $m[s_0\rangle m'$. We know that $m[h.s_0\rangle m"[t\rangle$. $\forall p \in P\ m'(p) = m"(p) - W(p, h)$. Let $p \in {}^\bullet t$.Due to the fact that no $t \in (H_i^\bullet \setminus \{p_i\})^\bullet$ occurs in s, we have to examine two cases:

- If $p \notin h^\bullet$, we have $m'(p) \geq m"(p) \geq W^-(p, t)$.
- If $p = p_i$ then $t \in F_i$. Moreover $\Gamma_i(s_0) = \Gamma_i(s_0.t) + 1 \geq 1$ Thus $m'(p_i) = m(p_i) + \Gamma_i(s_0) \geq 1 = W^-(p, t)$. The last equality is due to the sp-agglomerability.

Consequently $m[s_0\rangle m'[t\rangle$.

Let $s = s_1.s_2$ be a decomposition of s, we prove by induction on the length of s_1 that $m[s_1.h.s_2\rangle$. Suppose that $s = s_1.h.t.s_2$ and that $m[s_1\rangle m_1[h\rangle m_h[t\rangle m_{ht}[s_2\rangle$. From the previous paragraph, we also know that $m[s_1\rangle m_1[t\rangle m_t$ and that $t \notin (H_i^\bullet \setminus \{p_i\})^\bullet$.

- Let $p \notin {}^\bullet t$, $m_t(p) = m_1(p) + W(p, t) \geq m_1(p) \geq W^-(p, h)$.
- Let $p = p_i$, then $p \notin {}^\bullet h$ $m_t(p) \geq 0 = W^-(p, h)$.
- Let $p \in {}^\bullet t, p \neq p_i$, then $p \notin h^\bullet$ due to our claim.
  $m_t(p) = m_h(p) - W(p, h) + W(p, t) \geq W^-(p, t) - W(p, h) + W(p, t) = W^+(p, t) - W(p, h) \geq -W(p, h) = W^-(p, h)$

So we have $m_t[h\rangle m_{ht}[s_2\rangle$

Proof of the proposition 4, page 18 (divergence freeness) Let s be an infinite sequence such that $|s|_{H_i} = \infty$ and let $h \in H_i$ such that $|s|_h = \infty$ (it exists since H_i is finite).

Let us suppose that the first condition is fulfilled. Since p_i occurs in a positive flow, p_i is structurally bounded. We denote such a bound by B. Let $m_0[s_n\rangle m$ be a firing sequence where s_n is the prefix of length n of s. One has: $m_0(p_i) + |s_n|_{H_i} - |s_n|_{F_i} = m(p_i) \leq B$. Thus $|s_n|_{H_i} - B + m_0(p_i) \leq |s_n|_{F_i}$. Taking the limit as n goes to ∞, one obtains $|s|_{F_i} = \infty$.

Let us suppose that the second condition is fulfilled. Let q be the place associated with h. With the same notations, one has: $m_0(q) + \Sigma_{t \in {}^\bullet q} |s_n|_t \cdot W^+(q,t) - |s_n|_h \cdot W^-(q,h) \geq m(q) \geq 0$. Thus $\Sigma_{t \in {}^\bullet q} |s_n|_t \cdot W^+(q,t) \geq |s_n|_h \cdot W^-(q,h) - m_0(q)$. Taking the limit as n goes to ∞, one obtains $|s|_{{}^\bullet q} = \infty$ with ${}^\bullet q \subset T_0 \cup F$.

Proof of the proposition 5, page 18 (quasi-persistence) Let m be a reachable marking such that $m[h\rangle$ and $m[s\rangle$ with $s \in (T_0 \cup F)^*$. At first we delete the neutral transitions which share input places with h leading to a sequence s'. Obviously, $\forall p \in P\ W(p,s') = W(p,s)$ and $\Pi_F(s') = \Pi_F(s)$

We claim that no transition which shares an input place with h occurs in s'. Let us suppose that such a transition occurs and let us focus on the first occurrence, i.e. $s' = s_1.t.s_2$ with t being the first occurrence. Since s_1 does not consume any token in an input place oh h, due to the second hypothesis there are two mutually exclusive input places of h and t. Since the first place is still marked, the second one is unmarked which prevents the firing of t. Since h and s' are structurally not conflicting, one has $m[h.s'\rangle$. Of course, if we have not deleted any neutral transition, the strong quasi-persistence is fulfilled.

Proof of the proposition 6, page 19 (H-similarity) Let us suppose that for some reachable marking m, one has: $m[h_i\rangle$ and $m[s\rangle m_1[h_j.f_j\rangle$ with $s \in (T_0)^*$. Since the net is quasi-persistent $m[h_i.s'\rangle$ with $s' \in (T_0)^*$ and $\forall p \in P$ $W(p,s') \geq W(p,s)$. Since the net is independent $m[s'\rangle m_2[h_i\rangle$ and as $W(s') \geq W(s)$, $m_2 \geq m_1$. Then there exists $f_i \in F_i$ such that

(1) $\forall p \in {}^\bullet f_i \setminus \{p_i\}$, $m_2(p) \geq m_1(p) \geq W^-(p, h_j.f_j) \geq W^-(p, h_i.f_i)$
(2) $m_2(p_i) \geq 0 = W^-(p_i, h_i.f_i)$
(3) $\forall p \in {}^\bullet h_i \setminus {}^\bullet f_i$, $m_2(p) \geq W^-(p,h_i) = W^-(p, h_i.f_i)$.

Thus $m_2[h_i.f_i\rangle$

Proof of the proposition 8, page 19 (F-independence) Let us suppose that for $m \in Reach(N, m_0)$, one has $m[h.s.f\rangle$ with $s \in (T_0 \cup H)^*$.

We prove by induction on the length of s that $m[h.f.s\rangle$ The base case is trivial. Now let us suppose that $s = s'.t$ and that $m[h.s'\rangle m'[t\rangle m_t[f\rangle$. We claim that $m[h.s'\rangle m'[f.t\rangle$.

Suppose that the first point is fulfilled. Since m' is a reachable marking such that $m'(p) > 0$ and p freezes the set ${}^\bullet({}^\bullet F \setminus \{p\}) \setminus F$, t does not belong to this subset. Thus $\forall q \in {}^\bullet f \setminus \{p\}\ W^-[q, f] \leq m_t(q) = m'(q) + W[q, t] \leq m'(q)$. Hence $m'[f\rangle m_f$.

Now let $q \in {}^\bullet t$, then since $t \in T_0 \cup H, q \neq p$.
If $q \notin {}^\bullet f$ then $m_f(q) \geq m'(q) \geq W^-[q, t]$.
If $q \in {}^\bullet f$ then as $t \notin {}^\bullet({}^\bullet F \setminus \{p\})$ one has $q \notin t^\bullet$. So $W^+[q, t] = 0$.
$m_f(q) = m_t(q) + W[q, f] - W[q, t] \geq W^-[q, f] + W[q, f] - W[q, t]$. So
$m_f(q) = W^+[q, f] + W^-[q, t] - W^+[q, t] = W^+[q, f] + W^-[q, t] \geq W^-[q, t]$.
Thus $m_f[t\rangle$.

Let us suppose that for $m \in Reach(N, m_0)$, one has $m[h.s.f\rangle$ with $\forall s' \in Pref(s), \Gamma(s') \geq 0$. Any intermediate marking (say m') reached by $h.s$ except the initial one fulfills $m'(p) > 0$. Thus no transition of H occurs in s since p freezes H. Now due to the conditions on the prefixes, no transition of F occurs in s. Thus $s \in T_0^*$ and the previous proof is still valid.

Suppose now that the second point is fulfilled. Condition 2.a) ensures that $m'[f\rangle m''$. Indeed, let $q \in {}^\bullet f$; if $W^-(q, t) \geq W^-(q, f)$ then $m'[t\rangle \implies m'[f\rangle$. Otherwise, by hypothesis, $W^-(q, t) \geq W^+(q, t)$. So, $m_t(q) \leq m'(q)$ and as $m_t[f\rangle$ then $m'[f\rangle$.

Now, 2.b) ensures that $m''[t\rangle$ $(m'[f\rangle m'')$. Let $q \in {}^\bullet t$; if $W^+(q, f) \geq W^-(q, f)$ then $m''(q) \geq m'(q)$. So, $m'[t\rangle \implies m''[t\rangle$. Otherwise, by hypothesis,
$W^+(q, f) \geq W^+(q, t)$. So, as $m'[t.f\rangle$, we have that $m'(q) \geq W^-(q, f) - W(q, t)$.
As $m''(q) = m'(q) - W^-(q, f) + W^+(q, f)$ it comes that $m''(q) \geq W^+(q, f) - W(q, t) \geq W^+(q, t) - W(q, t) = W^-(q, t)$. So $m''[t\rangle$.

DEFINITION **B.2 (Original Berthelot's pre-agglomeration conditions)** *A transition h is pre-agglomerable with a set of transitions F ($h \notin F$) if there exists a place p such that:*

(1) $m_0(p) = 0$, $^\bullet(p) = \{h\}$, $p^\bullet = F$;
(2) $^\bullet h \neq \emptyset$ *and* $\forall q \in {}^\bullet h, q^\bullet = \{h\}$;
(3) $h^\bullet = \{p\}$;
(4) $W^+(p,h) = 1$ *and* $\exists m \in \mathbb{N}^+ \mid \forall f \in F, W^-(p,f) = m$.

DEFINITION **B.3 (Original Berthelot's pre-agglomeration transformation)** *The reduced net* $\langle P_r, T_r, W_r^+, W_r^-, m_{0r} \rangle$ *is defined by*

- $P_r = P \setminus \{p\}$ *and* $\forall q \in P_r, m_{0r}(q) = m_0(q)$;
- $T_r = T \setminus \{h\}$;
- $\forall t \in T_r \setminus F$, $\forall q \in P_r$, $W_r^+(q,t) = W^+(q,t)$ *and* $W_r^-(q,t) = W^-(q,t)$.
- $\forall f \in F$, $\forall q \in P_r$, $W_r^+(q,f) = W^+(q,f)$ *and* $W_r^-(q,f) = m.W^-(q,f)$.

Berthelot demonstrated [Ber83] that, if a net fulfils previous conditions, this transformation preserves the liveness, the boundedness and some other general properties less interesting. Note that if $m = 1$ our reduction covers immediately Berthelot's one.

If $m > 1$, we transform the net N into the net N' with the following transformation (see Fig.B.1): we create m places, $p^0, p^1, \ldots p^{m-1}$, m transitions $h^0, h^1, \ldots h^{m-1}$, we suppress the transition h and we link new places and new transitions to the net as follows:

- $\forall i, (p^i)^\bullet = \{h^i\}$ and $\forall i > 0$, $^\bullet(p^i) = \{h^{i-1}\}$ and $^\bullet(p^0) = \{h^{m-1}\}$
- $\forall i$, $^\bullet(h^i) = {}^\bullet h \cup \{p^i\}$, $\forall q \in {}^\bullet h$, $W'^-(q,h^i) = W^-(q,h^i)$ and $W'^-(p^i,h^i) = 1$;
- $\forall i > 0$, $(h^i)^\bullet = \{p^{i+1}\}$ and $W'^+(h^i,p^{i+1}) = 1$;
- $(h^{m-1})^\bullet = \{p^0\} \cup \{p\}$, $W'^+(h^{m-1},p^0) = 1$ and $W'^+(h^{m-1},p) = 1$;
- $m'_0(p^0) = 1$ and $\forall i > 0, m'_0(p^i) = 0$;
- the rest of the net is unchanged.

At first remark that any fireable sequence s of N can be transformed in a fireable sequence of N'. For doing this we count the number of h occurrences in s; if it's the k^{th} occurrence, then we rename it into $h^{k \bmod m}$. In the other way, we replace any occurrence of h^i in a firing sequence of N' into an occurrence of h and we obtain a firing sequence of N. Thus, it's obvious to demonstrate that this transformation preserves all properties preserved by our pre-agglomeration (which include those ones of Berthelot).

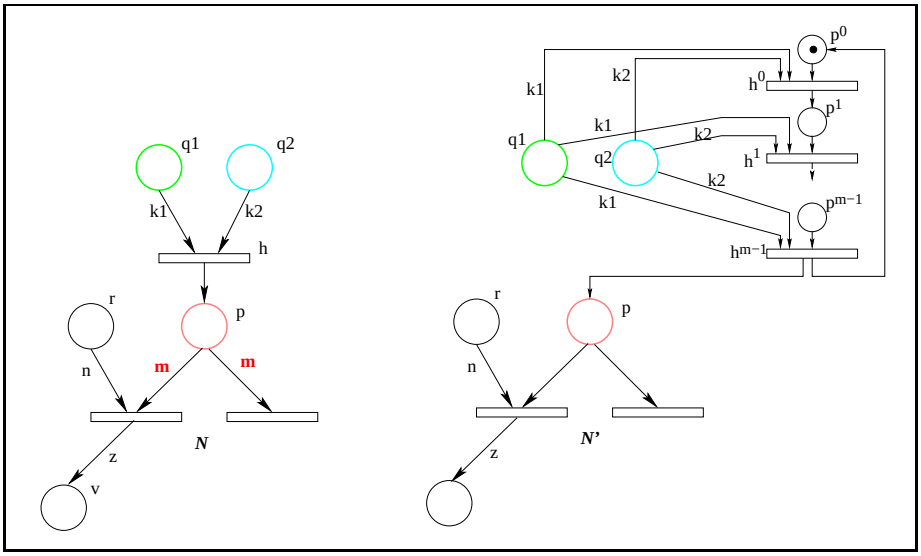

Fig. B.1. Net valuation transformation

Now, as each h^i are not in conflict (almost one is fireable at a given marking), and as $\forall i < m-1, (h^i)^\bullet = \{p^i\}$ we can perform m pre-agglomeration with our rules (we begin by h^{m-2} with h^{m-1}). We obtain the net depicted in the left of Fig B.2.

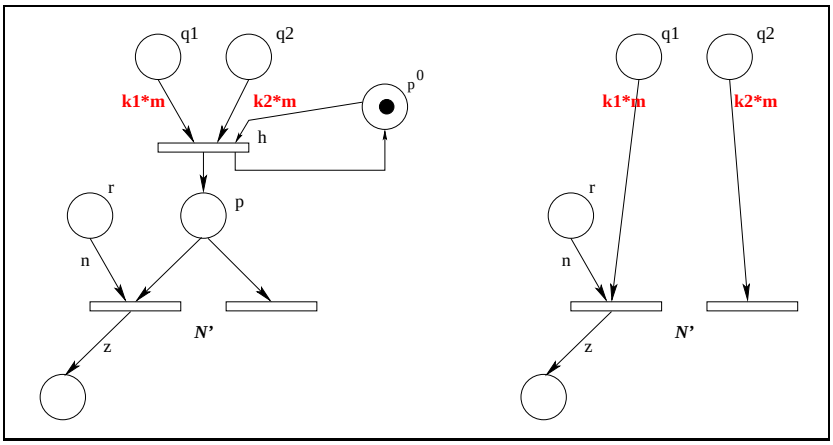

Fig. B.2. Net valuation transformation (follows)

In this net, we can suppress the place p_0 and perform one pre-agglomeration. We obtain the net depicted in the right of Fig. B.2 which is exactly the one obtained when applying Berthelot's transformation.

DEFINITION **B.4 (Original Berthelot's post-agglomeration conditions)** *A set of transitions H are post-agglomerable with a set of transitions F ($H \cap F = \emptyset$) if there exists a place p such that:*

(1) $m_0(p) = 0$, $^\bullet(p) = H$, $p^\bullet = F$, $F^\bullet \neq \emptyset$;
(2) $^\bullet F = \{p\}$ *and* $\exists m \in \mathbb{N}^+ \mid \forall f \in F, W^-(p,f) = m$;
(3) $\forall h \in H, \exists k_h \in \mathbb{N}^+ \mid W^+(p,h) = m * k_h$.

For the following definition we denote by $Rep(k,E)$, where k is a positive integer and E a finite set, the set of mappings g from E to $\mathbb{N}$ such that, $\sum_{e \in E} g(e) = k$ (each mapping characterises a repartition of k items between E). This notation is useful since, when h produces k tokens which can be used in different way by the transitions of F.

DEFINITION **B.5 (Original Berthelot's post-agglomeration transformation)** *The reduced net* $\langle P_r, T_r, W_r^+, W_r^-, m_{0r} \rangle$ *is defined by*

- $P_r = P \setminus \{p\}$ *and* $\forall q \in P_r, m_{0r}(q) = m_0(q)$;
- $T_r = (T \setminus (H \cup F)) \cup_{h \in H, g \in Rep(k_h,F)} \{t_{h,g}\}$;
- $\forall t \in T_r, t \neq t_{h,g}$, $\forall q \in P_r$, $W_r^+(q,t) = W^+(q,t)$ *and* $W_r^-(q,t) = W^-(q,t)$.
- $\forall t_{h,g} \in T_r$,
  - $\forall q \in P_r$, $W_r^-(q,t_{h,g}) = W^-(q,h)$;
  - $\forall q \in P_r \setminus F^\bullet$, $W_r^+(q,t_{h,g}) = W^+(q,h)$;
  - $\forall f \in F$, $\forall q \in P_r \cap f^\bullet$, $W_r^+(q,t_{h,g}) = g(f).W^+(q,h)$;

Berthelot demonstrated [Ber83] that, if a net fulfils previous conditions, this transformation preserves the liveness, the boundedness and some other general properties less interesting.

We suppose that $m = 1$ (this does not change the general nature of our proof). Given a net N fulfilling Berthelot's post-agglomerations conditions, we transform it into a net N' defined as follows:

- we note $K = max_{h \in H} k_h$, $F = \{f_1, f_2, \ldots, f_n\}$;
- we suppress transitions of F and place p;
- we create K new places $p^1, \ldots, p^K$ and we rename each $h \in H$ into h';
- for each transition $f_i \in F$ we create K new transitions $f_i^1, \ldots, f_i^K$;
- $\forall i \in 1..N$, $\forall k \in 1..K$, $^\bullet(f_i^k) = \{p^k\}$ and $W^-(p^k, f_i^k) = 1$; $(f_i^k)^\bullet = f_i^\bullet$ and $\forall q \in f_i^\bullet$, $W^+(q, f_i^k) = W^+(q, f_i)$;
- $\forall h \in H$, $h'^\bullet = (\{p^{k_h}\} \cup h^\bullet) \setminus \{p\}$ and $W^+(h, p^{k_h}) = 1$;
- the rest of the net is unchanged.

Again, we can easily prove that N and N' have the same properties (among those considered for the post-agglomeration). Now, N' can be reduced by our post-agglomeration and we obtain the same net than the one produced by Berthelot's transformation.