



HAL
open science

Non-Ambiguous Petri Nets

Bruno Gaujal, Stefan Haar

► **To cite this version:**

Bruno Gaujal, Stefan Haar. Non-Ambiguous Petri Nets. [Research Report] RR-3691, INRIA. 1999. <inria-00072978>

HAL Id: inria-00072978

<https://inria.hal.science/inria-00072978v1>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Non-Ambiguous Petri Nets

Bruno Gaujal , Stefan Haar

No 3691

May 19, 1999

————— THÈME 1 —————

 ***rapport
de recherche***


Non-Ambiguous Petri Nets

Bruno Gaujal ^{*}, Stefan Haar [†]

Thème 1 — Réseaux et systèmes
Projet TRIO

Rapport de recherche n° 3691 — May 19, 1999 — 30 pages

Abstract: In this paper, we investigate conditions for *non-ambiguous behaviour* of a timed Petri net which also contains immediate transitions. Non-ambiguity of the net behaviour means that it is uniquely determined by a given set of *control parameters* defined in this paper. To this effect, we describe two semantics, that deal with non-determinism in different ways. The first policy, called the *limit policy* considers immediate transitions as limits of timed transitions with timing ϵ . The second policy, called the *global priority policy* defines a dynamical order among the transitions and fires them along that order. The paper also investigates the advantages, the drawbacks and the relations between those two policies.

Key-words: Timed Petri nets, Semantics, Nondeterminism.

(Résumé : *tsvp*)

^{*} Loria, 615, rue du Jardin botanique, 54602 Villers-les-Nancy, France. Email: Bruno.Gaujal@loria.fr

[†] Loria-Ensem, 2 avenue de la Forêt de la Haye, 54516 Vandoeuvre, France. Email: Stefan.Haar@loria.fr This author is supported by the European TMR project ALAPEDES, Contract reference ERB-FMRX-CT-96-0074

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY (France)
Téléphone : 03 83 59 30 30 - International : +33 3 3 83 59 30 30
Télécopie : 03 83 27 83 19 - International : +33 3 83 27 83 19
Antenne de Metz, technopôle de Metz 2000, 4 rue Marconi, 55070 METZ
Téléphone : 03 87 20 35 00 - International: +33 3 87 20 35 00
Télécopie : 03 87 76 39 77 - International : +33 3 87 76 39 77

Non-Ambiguous Petri Nets

Résumé : Dans cet article, nous analysons les conditions pour avoir un comportement non ambigu d'un réseau de Petri temporisé qui contient des transitions immédiates. La non-ambiguïté du comportement du réseau signifie que le comportement est déterminé de façon unique par un ensemble de paramètres de contrôle qui sont définis dans cet article. Pour ce faire, nous décrivons deux sémantiques qui traitent le non-déterminisme de manière différente. La première politique, appelée *politique limite* considère les transitions immédiates comme des limites de transitions temporisées par ϵ . La deuxième politique, appelée la *politique à priorité globale* définit un ordre dynamique sur les transitions qui sont tirées selon cet ordre. L'article cherche aussi à montrer les avantages, les inconvénients et les dangers des deux politiques ainsi que leurs relations.

Mots-clé : Réseaux de Petri temporisés, Sémantique, non-déterminisme

1 Introduction

Petri nets constitute a model of discrete systems that combines concurrency and competition at the node level. They were primarily used for analysing logical properties of systems involving parallelism and synchronisations, see for example [7, 3, 5]. More recently, the notion of time has been introduced in Petri nets to make performance analysis possible. However, this addition of time may alter the semantics of a Petri net. In particular, the presence of *immediate* transitions, whose firing takes no time at all, besides timed transitions, makes the semantics surprisingly complex. The difficulties arising here have led to the introduction of *vanishing states* and to very strong restrictions imposed on the structure allowed for such nets (cf. [2]). This paper will deal with the problem of finding a possible semantics for general timed Petri nets, under weak assumptions.

A central problem in Petri nets – which remains even after the introduction of time – is *non-determinism*. It is important to be able to control the non-determinism in timed Petri nets, for example for simulation or analysis purposes. This non-determinism comes, in general, from the following sources:

1. Conflicts over tokens between transitions sharing a pre-place;
2. “races” among tokens and transitions firing in parallel.

A standard approach to overcoming non-determinism is to introduce a probabilistic framework where non-deterministic choices are replaced by probabilistic ones. Many authors deal with this issue using firing times with exponential distributions (or constant firing times) and breaking conflicts through the race policy. Under these conditions, the evolution of the net can be modeled by a (semi)-Markovian process [2].

This approach, however, makes it very difficult to use priority levels among transitions, to deal with the existence of immediate transitions anywhere in the net and with routing of tokens.

Here, we will present a different approach to this problem based on a sample path model. We focus on a single trajectory of the system obtained for a given *fixed* set of parameters. This study can then be followed by an investigation of the effect that changing those parameters has on the behaviour. These changes can, but need not, be determined by probability distributions.

For this, however, it is crucial to ensure functionality of the behaviour w.r.t. the parameters. We will present a model of Petri Nets with *timing*, *routing*, and *priority*

ordering, and investigate the conditions under which these parameters determine the behaviour of the net uniquely; such a net will be called *non-ambiguous*.

Also, two *policies* for computing the behaviour are presented; they can handle different structural classes of nets. Wherever they are jointly applicable they differ substantially and it is up to the user to decide which one is more adequate for a given context.

The paper is structured as follows. In Section 2, we introduce the model of timed Petri nets we will consider. The main feature is the presence of routing functions which allow to make routing decisions for tokens (or not). In Section 3, we introduce priorities and the enabling of the transitions. Sections 5.1 and 6.1 present two different semantics, each of them defining a unique behaviour of the net (under appropriate general conditions). An algorithm to compute their respective behaviour is provided in each case.

2 The model

We shall now formally define the notion of timed Petri Nets that will be used in the following and which is as general as possible. In particular, it will include classical Petri nets, as defined in [4]. A timed Petri net with routing is a tuple $\mathcal{S} = (\mathcal{P}, \mathcal{Q}, \mathcal{F}, M_0, \nu, \Phi, H)$, where

- \mathcal{P} is a finite set of places,
- \mathcal{Q} a finite set of transitions,
- $\mathcal{F} \subset (\mathcal{P} \times \mathcal{Q}) \cup (\mathcal{Q} \times \mathcal{P})$ the set of arcs. We will denote by p^\bullet (${}^\bullet p, q^\bullet, {}^\bullet q$), the set of output (input) transitions (places) of place (transition) p (q). An essential restriction we need to impose is that every transition has at least one pre-place, i.e. ${}^\bullet q \neq \emptyset$ for all q ; otherwise, q can fire at unpredictable times and an unpredictable number of times.
- M_0 is the initial marking,
- $\Phi = \{\phi_q, q \in \mathcal{Q}\}$ is the set of the firing times of the transitions. The firing of transition q is assumed to take ϕ_q units of time. All these durations are constant; in the algorithms, we will assumed them to be multiples of a common time unit δ , which can be taken as unity without loss of generality.

Note first that the case with a finite number of different durations for the same q can be modelled as well: delete q from the net structure, and for each value

d of the duration of q , create a transition q_d such that q_d has the same input and output places as q , and $\phi_{q_d} = d$.

Note, further, that some transitions can have a zero firing time. Such transitions are called *immediate transitions*. The set of all immediate transition is denoted \mathcal{I} , and its cardinality by I . The set of non-immediate transitions, $\mathcal{Q} \setminus \mathcal{I}$, is denoted \mathcal{U} , with cardinality U .

- $\nu = \{\nu_p(n, t), p \in \mathcal{P}, n \in \mathbb{N}, t \in \mathbb{R}\}$ is the set of routing decisions. The routing decision $\nu_p(n, t)$ gives the sub-set of p^\bullet for which the n -th token to enter place p is visible. This routing decisions are given *a priori* (in particular, they are independent of the marking in other places). A token routed to $m \subset p^\bullet$ cannot be fired by any transition $q' \notin m$ and stays in place p until one of the transitions in m fires. When the set m is always reduced to a single element, then the routing decisions are called *unequivocal*. This is the case in [1]. When this subset is always the full set p^\bullet , then, this is the classical case where no routing decision is taken.
- $H = \{H_{p,i}(n), p \in \mathcal{P}, i \in p^\bullet, n \in \mathbb{N}\}$ is the set of the routing functions. The routing function $H_{p,i}(n)$ gives the number of tokens routed to each transition $i \in p^\bullet$ among the first n tokens to have arrived in place p . By definition, we have

$$H_{p,i}(n) = \sum_{k=1}^n \mathbf{1}_{\{i \in \nu_p(k, t_k)\}},$$

where t_k is the date of arrival of the k th token in place p . The two extreme cases are when $H_{p,i}(\cdot)$ is the identity function for all $i \in p^\bullet$ (all tokens are available for all output transitions), this is the no routing case, and when $\sum_{i \in p^\bullet} H_{p,i}(n) = n$ (each token is routed to a single transition), this is the *unequivocal* case.

3 Priorities

In a Petri net as defined above, several transitions may be enabled at the same time. Among those transitions, some can be in conflict, meaning that they require tokens on a shared place for being enabled.

One obvious way to break local conflicts in Petri nets is to give static priorities to transitions that share an input place.

More formally, $\Pi = \{\pi_p(i), p \in \mathcal{P}, i \in p^\bullet\}$ is the set of priorities for the output transitions of each place:

$$\pi_p : p^\bullet \rightarrow \mathbb{N}$$

is injective for every $p \in \mathcal{P}$. If $i, j \in p^\bullet$ and if $\pi_p(i) < \pi_p(j)$, we say that i has higher priority than j . This will also be denoted $i <_p j$.

The aim of this paper is to address the following problem: Is this local conflict resolution sufficient to give a unique behaviour (in a sense that will be made clearer below) to the whole net?

The priorities of the places impose a relation (which we require to be an order) on the set of the transitions. If two transitions, t_1 and t_2 , are output transitions of a place p , and if t_1 has higher priority than t_2 , then we have that $t_1 <_p t_2$.

(C_1) All the priority orders $\{<_p, p \in \mathcal{P}\}$ are compatible, i.e. the union of all the priority orders is a partial order $<_{\mathcal{P}}$ on the set of transitions.

If the priority orders are not compatible, then the net (and/or the priority assignments) are badly designed, and no valid numbering of the transitions can be found. In the following, we will assume that condition (C_1) is always satisfied.

Note that, under (C_1), the order $<_{\mathcal{P}}$ induces (in general) several admissible numbering of the transitions, by any of its total order extensions

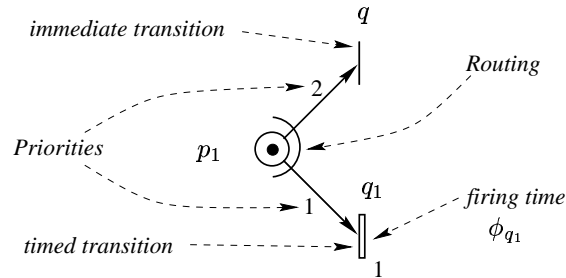


Figure 1: The general Petri net model

Figure 1 shows an example of a Petri net.

4 Semantics

To describe the state of the net, we will use as state variables, the *right continuous* functions $X_i(t)$ that give the number of firings initiated by transition i at time t .

In the following, the sequences: $\{X_i(t), i \in T, t \in \mathbb{R}^+\}$ will be called the *behaviour* of the Petri net. Note that these sequences determine exactly the evolution of the net. Also note that, if all firing times are integer, it follows from the right continuity assumption that

$$\forall i \in T, \forall t \in \mathbb{R}^+, X_i(t) = X_i(\lfloor t \rfloor).$$

This means that giving the behaviour at integer times suffices to obtain the complete behaviour of the net.

Enabling - A transition q is *enabled* at time t if each input place p of q contains a token which is routed to p at time t .

Firing - If a transition q starts firing at time t , then one token is removed from all input places (for routing places, this token is the token routed to q) at time t . The transition ends the firing at time $t + \phi_q$ and one token is added in all output places of q at time $t + \phi_q$.

Note that for immediate transitions, although the firing takes no time, the sequence of events still happens in that order.

Also note that the marking in a place p with input transitions $\{i_1, i_2, \dots, i_k\}$ and output transitions $\{o_1, o_2, \dots, o_m\}$ does not satisfy $M_p(t) = \sum_{i_j} X_{i_j}(t - \phi_{i_j}) - \sum_{o_j} X_{o_j}(t) + M_p(0)$ at *any* instant t . However, the equality becomes true at integer times:

$$\forall n \in \mathbb{N}, \quad M_p(n) = \sum_{j=1}^k X_{i_j}(n - \phi_{i_j}) - \sum_{j=1}^m X_{o_j}(n) + M_p(0).$$

Firing Sequence - The most delicate part in the construction of the semantics of a Petri nets is to define the choice of the next transition to be fired. Note that firing sequences list all firings of transitions *one after the other*, whereas in the *behaviour*, transitions may fire simultaneously. Therefore, there are in general several different sequences giving the same behaviour. At any stage in the evolution of the net, the net obeys the following firing rules:

- (R_1) Fire the next transition to be enabled under the current marking.
- (R_2) If several transitions are enabled under the current marking, fire one of the transitions with the highest priority.

Two remarks are in order regarding the rules R_1 and R_2 :

- Once a transition q has been chosen according to R_1 and R_2 , the remaining transitions are immediately reconsidered under the new marking that resulted from q 's removing tokens from its input places, and other transitions may start firing. In other words, the first event moves the system into a new *logical* state, but *physical* time has not been advanced yet.
- Any transition enabled and chosen as above *immediately* starts firing. This will be referred to as the *earliest firing rule* (although it is merely a consequence of the above rules rather than a rule of its own).

It is possible that several enabled transitions have the highest priority; under (C1), this is of course only possible when they do not share an input place. In that case there are two possible continuations of the firing sequence, either by firing one or the other first; the question is whether there may also be differences in the behaviour.

Definition 1. *A Petri net is non-ambiguous if it admits a unique behaviour satisfying the firing rules.*

Proposition 1. *Under condition (C1), if the net does not contain any immediate transition, then it is non-ambiguous.*

Proof. Suppose there exist two different behaviours for \mathcal{S} , $\{X^1(t) : t \in \mathbb{R}^+\}$ and $\{X^2(t) : t \in \mathbb{R}^+\}$. Since counter functions are right continuous and change values by integer jumps, there exists a smallest $t \in \mathbb{R}^+$ such that $X^1(t) \neq X^2(t)$. For $i \in \{1, 2\}$ and $s \in \mathbb{R}^+$, denote by $Enabled_s^i$ the set of all transitions that are enabled to fire at s and that are minimal with respect to all \prec_p . Then $Enabled_t^1 = Enabled_t^2 =: Enabled_t$, and $Enabled_t$ contains no conflicting pair of transitions by virtue of (C1). Therefore, under the earliest firing rule, exactly the transitions in $Enabled_t$ are fired at t , hence $X^1(t) = X^2(t)$ contradicting the assumption. \square

However, when a Petri net contains immediate transitions, it may have several behaviours. To illustrate this fact, let us consider the Petri net displayed in Figure 2.

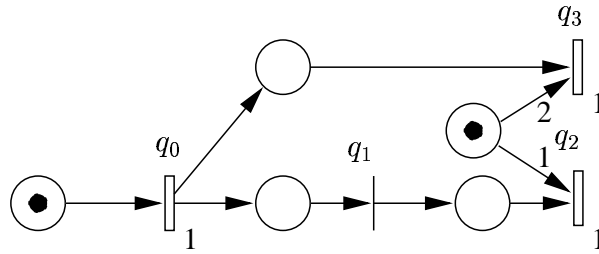


Figure 2: An Ambiguous Petri net

At time 0, only transition t_0 fires. At time 1, transition t_1 and t_3 are enabled. The local priority does not help to decide which one to fire.

If one chooses to fire t_1 then the next transition to fire will be t_2 , and t_3 will not fire. We have thus

$$X_0(1) = 1, X_1(1) = 1, X_2(1) = 1, X_3(1) = 0$$

If one chooses to fire t_3 then no other transition will fire. We have

$$X_0(1) = 1, X_1(1) = 0, X_2(1) = 0, X_3(1) = 1$$

Therefore, in order to impose a unique behavior on the net, we have to strengthen the conditions on firing a transition. In the following, we will present two possible policies, each of them either yielding a different behavior and different features.

5 Limit Policy

5.1 Definition and Properties

This policy (denoted (LP)) consists in considering the behaviour of immediate transitions as limits of that of timed transitions as their duration tends to 0. Given $\epsilon > 0$ and a Petri Net \mathcal{S} which contains immediate transitions, we construct a Petri net \mathcal{S}_ϵ from \mathcal{S} by assigning to all its immediate a firing time equal to ϵ .

Since \mathcal{S}_ϵ does not contain any immediate transitions, it is non-ambiguous because of Proposition 1 (for ϵ arbitrary but fixed). Its behaviour is denoted by $X^\epsilon(t)$.

Definition 2. Whenever the limit $X^0(t) \stackrel{\text{def}}{=} \lim_{\epsilon \rightarrow 0} X^\epsilon(t)$ exists, the behavior of \mathcal{S} up to time T under policy (LP) is given by $X^{LP}(t) \stackrel{\text{def}}{=} \lim_{\delta \rightarrow 0} X^0(t+\delta)$ for $0 \leq t \leq T$.

Note that this limit w.r.t. δ always exists because $X^\epsilon(T+\delta)$ is decreasing in δ . Defining the limiting behavior as above takes into account all those events which occur at times smaller than $T+k\epsilon$, which can be considered as occurring no later than T for the limit, because of the right continuity of the behavior.

In the following, the focus of this section will be to give conditions under which the limit $X^0(t)$ exists.

First, we introduce a topological condition which is easy to verify (but, as we will see, quite restrictive):

(C₂) We say that a net satisfies condition (C₂) if it does not contain a cycle made of immediate transitions.

Lemma 1. Under condition (C₂), the total number of firings in S_ϵ before time T is bounded, uniformly in ϵ .

Proof. We will show that there exists a uniform (in ϵ)

upper bound $K = K(T) \in \mathbb{N}$ on the total number of tokens created in S_ϵ before time T (including those present in the initial marking M_0); from that, the claim follows immediately.

We first partition the interval $[0, T]$ into a finite number of (more easily tractable) subintervals $Int(n)$. Since \mathcal{U} is finite, $\eta_1 := \min\{\phi_q : q \in \mathcal{U}\}$ is positive; set $\eta := \frac{\eta_1}{2}$ and $m = m(T) := \lceil \frac{T}{\eta} \rceil$, and for $n \in \{0, \dots, m-1\}$ define

$$Int(n) := [n\eta, (n+1)\eta].$$

Then $[0, T] \subset \bigcup_{n=0}^{m-1} Int(n)$, and for any $n \in \{0, \dots, m-1\}$ and any firing event e of a transition in \mathcal{U} , e cannot both start and end within $Int(n)$. Now, each transition $q \in Q$ has a finite token reproduction factor independent of time, marking, or duration, namely

$$\kappa_q := \frac{\sum_{p \in q^\bullet} W(q, p)}{\sum_{p \in \bullet q} W(p, q)}.$$

Set $\kappa := \lceil \max\{\kappa_q : q \in Q\} \rceil$; then κ is finite since Q is. Now, let $\tilde{\Theta}_n$ denote the number of tokens present in S_ϵ at the beginning of $Int(n)$ (where $\Theta_0 := \sum_{p \in P} M_0(p)$), and

$$\Theta_n := \sup_{k=0, \dots, n} \tilde{\Theta}_k$$

(which could, in principle, be infinite; it will follow inductively from the following that all Θ_n are finite). During $Int(n)$, the transitions from \mathcal{U} can create at most $\Theta_n^{\mathcal{U}} := \kappa|U|\Theta_n$ new tokens, since they can only release tokens during $Int(n)$ by ending a firing that was started before time $n\eta$. Since, by (C2), the length of every chain of immediate transitions is bounded from above by I , within $Int(n)$, the number of tokens can be increased at most I times by a factor of at most κ ; thus we have

$$\Theta_{n+1} \leq \Theta_n^{\mathcal{U}} \left(\kappa^I \right)$$

and thus $\Theta_n \leq \Theta_0 (U\kappa^{I+1})^n$ (from which it follows that, as claimed above, all Θ s are finite). Then

$$K(T) := \Theta_0 \sum_{n=0}^{m(T)} (U\kappa^{I+1})^n$$

is a finite upper bound for the number of tokens that are

- initially present in S^ϵ or
- created during $[0, T]$,

and $K(T)$ is independent of ϵ . Hence Lemma 1 follows. \square

For the following result, we need the notion of a *trajectory*: A trajectory τ is an ordered multi-sequence, which gives the order of events (beginning and end of firing of the transitions). If several events occur at the same time, then they are considered simultaneous in τ and called multi-events.

Lemma 2. *Under the assumption that the net does not contain immediate cycles (C₂), the limit $X^0(t) \stackrel{\text{def}}{=} \lim_{\epsilon \rightarrow 0} X^\epsilon(t)$ exists.*

Proof. By Lemma 1, the number of events up to time T is bounded uniformly in ϵ . Therefore, the number of possible trajectories is also bounded uniformly in ϵ . Hence there exists trajectories of accumulation, when ϵ goes to 0.

Let τ be one such trajectory, and let $(\epsilon_k)_{k \in \mathbb{N}}$ be a decreasing sequence going to 0 and such that X^{T, ϵ_k} has the trajectory τ for all k .

In the first part of the proof, we show that on τ , the behavior X^{T, ϵ_k} has a limit. We introduce a new variable, $Y_q^{T, \epsilon_k}(n)$ which gives the number of times transition q has started firing among the first n multi-events on trajectory τ .

We show by induction on n that $Y_q^{\tau, \epsilon_k}(n)$ is decreasing in k , i.e.

$$Y_q^{\tau, \epsilon_k}(n) \leq Y_q^{\tau, \epsilon_{k+1}}(n). \quad (1)$$

First, consider the first multi-event on trajectory T . It is the same for ϵ_k and for ϵ_{k+1} and takes place at time 0. Therefore, $Y_q^{\tau, \epsilon_k}(1) \leq Y_q^{\tau, \epsilon_{k+1}}(1)$ holds with equality. Let us assume that (1) is true after $n - 1$ events on trajectory τ . The n -th multi-event e is the same for ϵ_k and for ϵ_{k+1} because we are on the same trajectory τ in both cases. Now e is triggered by the same set of earlier multi-events. By induction, the triggering multi-events occur no later under timing ϵ_k than under ϵ_{k+1} . Furthermore, the firing times of the triggering multi-events are larger under timing ϵ_k than under ϵ_{k+1} . Therefore, the n -th multi-event occurs no later under timing ϵ_k than under ϵ_{k+1} . This implies directly that (1) holds.

Therefore, $Y_q^{\tau, \epsilon_k}(n)$ has a limit for $k \rightarrow \infty$, called $Y_q^{\tau, 0}(n)$ in the following.

In the second part of the proof, we will show that there is a single trajectory of accumulation. Let us assume that τ_1 and τ_2 are two trajectories of accumulation with corresponding subsequences $(\epsilon_k^1)_{k \in \mathbb{N}}$ and $(\epsilon_k^2)_{k \in \mathbb{N}}$ respectively.

Let n be the smallest index such that τ_1 and τ_2 differ (n may be 0). Up to n , both trajectories are equal, that is, the order in which events occur from the first instant up to the $(n - 1)$ th occurrence of an event are the same for all $(\epsilon_k^1)_{k \in \mathbb{N}}$ and for all $(\epsilon_k^2)_{k \in \mathbb{N}}$.

On both trajectories and up to event n , the limits of $Y_q^{\tau_1, 0}(k)$ and $Y_q^{\tau_2, 0}(k)$ coincide. The proof of this assertion is done by induction. At time 0, the marking is the same in both cases, therefore, exactly the same transitions will start firing (see Proposition 1). Let us assume that $Y_q^{\tau_1, 0}(j)$ and $Y_q^{\tau_2, 0}(j)$ coincide up to $j = k$. Then, since the trajectories τ^1 and τ^2 are still the same, the $k + 1$ -th multi-event is the same under both trajectories, triggered by the same earlier multi-set e' . By assumption, e' occurs – at the limit – at the same time on both trajectories. Also, at the limit, both systems have the same firing times (0 for immediate transitions). Therefore, the limits coincide for the $k + 1$ -th event, $Y_q^{\tau_1, 0}(k + 1) = Y_q^{\tau_2, 0}(k + 1)$.

Now, we have $Y_q^{\tau_1, 0}(k) = Y_q^{\tau_2, 0}(k)$ for all $k < n$. This means that before the n -th event, the situation is exactly the same under both systems (at the limit). Therefore, the n -th event must be the same on both trajectories.

This means that we have a single trajectory τ of accumulation. On this trajectory, we have by definition

$$X_q^\epsilon(t) = X_q^{\tau, \epsilon}(t) = \sum_{n=0}^{\infty} \mathbf{1}_{\{Y_q^{\tau, \epsilon}(n) \leq t\}}.$$

If $K(t)$ is the bound on the total number of firings whose existence is guaranteed by Lemma 1, we have

$$X_q^\epsilon(t) = \sum_{n=0}^{K(t)} \mathbf{1}_{\{Y_q^{\tau, \epsilon}(n) \leq t\}}$$

and thus

$$X_q^0(t) = \lim_{\epsilon \rightarrow 0} X_q^\epsilon(t) = \sum_{n=0}^{K(t)} \mathbf{1}_{\{Y_q^{\tau, 0}(n) \leq t\}}.$$

□

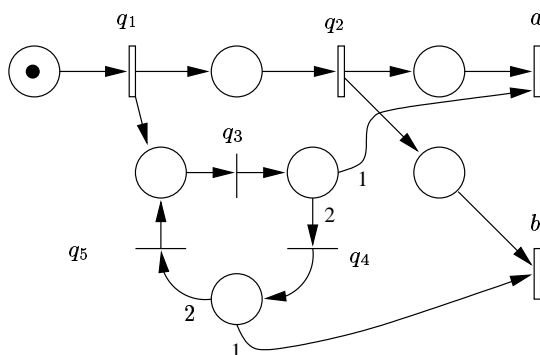


Figure 3: The limit policy does not exist

Dropping Condition (C2) in Lemma 2 is not possible in general, as Figure 3 shows. Indeed, when ϵ goes to 0, either transition a or transition b fires, according to the value of the temporisation ϕ_1 with respect to ϵ . If $-\epsilon \leq \phi_1 < \epsilon \bmod 3\epsilon$, transition a fires. However, if $\epsilon \leq \phi_1 < 2\epsilon \bmod 3\epsilon$, then transition b fires. Therefore, the limit when ϵ goes to 0 does not exist.

Lemma 3. *The limit $X^0(t) = \lim_{\epsilon \rightarrow 0} X^\epsilon(t)$ exists without condition (C2) as long as the total number of events remains finite, uniformly bounded for all ϵ .*

Proof. The proof of Lemma 2 only uses the fact that the total number of events is finite which implies that the number of trajectories is also finite. Condition (C2) *per se* is never used. □

Now, we will present a more general structural condition under which the total number of events is finite even if the net contains immediate cycles.

We construct inductively a set of transitions $\mathcal{C} \subset \mathcal{Q}$; all transitions in \mathcal{C} will be called *controlled* transitions in the following.

Definition 3. *The set \mathcal{C} is the smallest set satisfying:*

1. $\mathcal{U} \subset \mathcal{C}$.
2. For all places p , $\bullet p \subset \mathcal{C} \Rightarrow p^\bullet \subset \mathcal{C}$.
3. For all places p , partition p^\bullet into two sets, namely, the set \mathcal{A} of transitions chosen by infinitely many routing decisions,

$$\mathcal{A} \stackrel{\text{def}}{=} \{j \in p^\bullet : H_{p,j}(n) \rightarrow_{n \rightarrow \infty} \infty\},$$

and those with a finite number of routing decisions,

$$\mathcal{B} \stackrel{\text{def}}{=} \{j \in p^\bullet : \forall n, H_{p,j}(n) \leq K_{p,j} \in \mathbb{N}\}.$$

Then $\mathcal{B} \subset \mathcal{C}$.

(C₃) If $\mathcal{Q} \subset \mathcal{C}$ (and hence $\mathcal{Q} = \mathcal{C}$), the net \mathcal{S} is called **controlled** and said to satisfy Condition (C₃).

First, note that the set \mathcal{C} is well defined. It can be constructed iteratively, starting with $\mathcal{C} = \mathcal{U}$ and adding new transitions using rules 2 and 3. Then, the order in which transitions are considered does not matter since the recursive conditions 2 and 3 are monotonic in \mathcal{C} , i.e. can not be rendered false by adding new elements to \mathcal{C} .

For the example in Figure 3, we have $\mathcal{C} = \mathcal{U} = \{q_1, q_2, a, b\}$, so the net is not controlled. Figure 4 shows a controlled net.

Remark 1. *If condition (C2) is satisfied in a net \mathcal{S} , then \mathcal{S} is controlled. This follows by induction over the length of chains of immediate transition; note that, for any place p , $\bullet p = \emptyset$ implies that $p^\bullet \subset \mathcal{C}$.*

Lemma 4. *If $\mathcal{C} = \mathcal{Q}$, then $\lim_{\epsilon \rightarrow 0} X^\epsilon(t)$ exists.*

In other words, the lemma says that if all transitions are controlled, then the net is non ambiguous with respect to the limit policy.

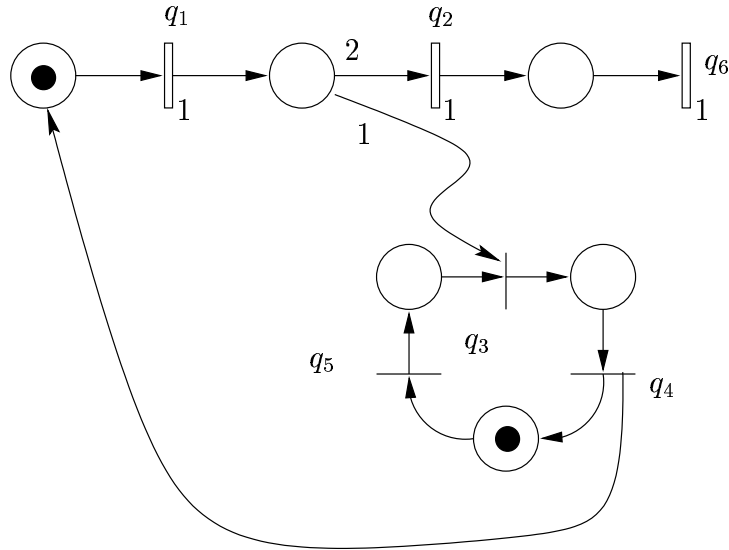


Figure 4: A controlled net

Proof. Using Lemma 3, it is enough to prove that if $\mathcal{Q} \subset \mathcal{C}$, the total number of firings up to time T is bounded uniformly in ϵ . The proof is now similar to the proof of Lemma 1 (we use the same notations): Let the number of firings before and up to the beginning of $Int(n)$ be bounded, and let \mathcal{F} be the set of transitions that complete only finitely many firings during $Int(n)$; it then suffices to show that $\mathcal{Q} \subset \mathcal{F}$. Since $\mathcal{U} \subset \mathcal{F}$, it suffices to show that \mathcal{F} is closed under the extension rules given by 2 and 3 of Definition 3. Now, if $\bullet p \in \mathcal{F}$, only finitely many tokens will ever reach p , and thus $p^\bullet \in \mathcal{F}$; and if only a finite number of tokens on some preplace of q is ever routed towards q , q must be in \mathcal{F} . Hence we are done. \square

The limit policy is applicable for some nets that do not fall into the categories we described here. Then, however, one has to accept *infinite* behaviors with *transfinite* properties. As an example, take the net obtained by deleting transition b in Figure 3. This net is still uncontrolled. Transitions q_1 and q_2 will fire exactly once, and, during the firing period of q_2 , transitions q_3 , q_4 , and q_5 will fire infinitely often; immediately after q_2 completes its firing, a will fire, killing the net. So the net is non-ambiguous with respect to $X(t)$. However, none of the policies described here

capture this behaviour; in our opinion, a non-anticipatory policy accounting for all such cases is not likely to exist.

The finitary case of Lemma 1 appears, therefore, to be the most general class of non-ambiguous nets that are still of practical interest; it is, however, defined in *semantical* rather than *structural* terms. The sub-class of *controlled* nets according to Definition 3 is identifiable by structural criteria (note that the routing is given *a priori*).

Choosing *LP* for the behavior of \mathcal{S} has several consequences.

Remark 2. *Note that in general the asymptotic behavior of \mathcal{S} (that is when the time grows to infinity) cannot be studied as the limit of the asymptotic behavior of \mathcal{S}_ϵ , since in general,*

$$\lim_{t \rightarrow +\infty} \lim_{\epsilon \rightarrow 0} X^\epsilon(t) \neq \lim_{\epsilon \rightarrow 0} \lim_{t \rightarrow +\infty} X^\epsilon(t).$$

In fact, the difference can even be unbounded, as illustrated by the net displayed in Figure 5: In net \mathcal{S} , transition q_5 never fires, therefore, $X_{q_5}^0(t) = 0$ for all t , and thus

$$\lim_{t \rightarrow \infty} X^0 t_5(t) = 0.$$

However, in \mathcal{S}_ϵ , transition q_5 fires at time $2k + k\epsilon$, for each k such that $1/2 \leq k\epsilon \leq 1 \pmod{2}$, which happens for infinitely many values of k . Therefore, for each $\epsilon < 1/2$, $\lim_{t \rightarrow +\infty} X^\epsilon q_5(t) = +\infty$.

5.2 Algorithm Limit(T)

This algorithm only works under Condition (C_3) and when the firing durations of the non-immediate transitions are all integer numbers. First we define the quantity ϵ by the inverse of the maximal number of firings that can happen during T units of time, which can be computed a priori.

Note that the main feature of this algorithm is that it uses two independent time scales: the integer numbers (with a one unit interval) AND the ϵ scale with ϵ so small that no occurring sequence firings with duration ϵ exceeds the unit interval; to avoid the accumulation of the ϵ 's to quantities that would not be negligible with respect to the integer scale, these time steps are “forgotten” once we move to the next unit interval. The variable Y_i computes the number of firings within one interval and is added to X_i at the end of the interval. Put differently, the firing times are treated as positive infinitesimals.

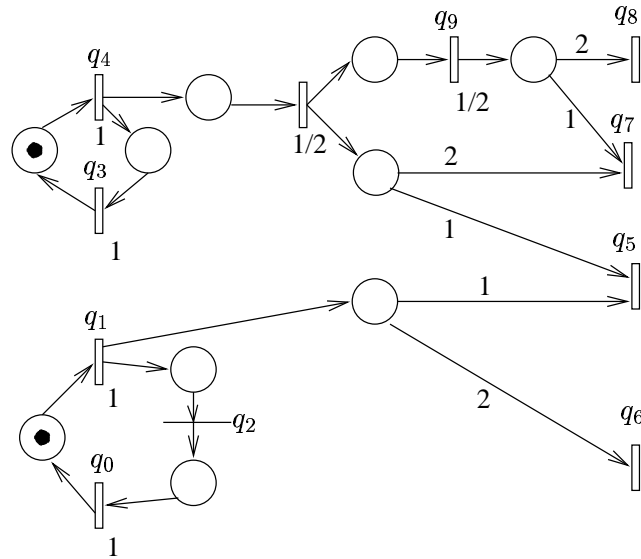


Figure 5: Example of a net \mathcal{S} whose asymptotic behavior differs from the limit of the asymptotic behavior of the ϵ -timed net \mathcal{S}_ϵ ; trivial routing

```

Limit(T)
  Order  $Q$  with any linear extension of the priorities. // numbering
   $\epsilon := 1/K(T)$ 
  for  $i$  from 1 to  $Q$  do if  $\phi_i := 0$  then  $\phi_i := \epsilon$  fi od
  for  $i$  from 1 to  $Q$  do  $Y_i(n) := X_i(0) := 0$  od // Initializations
  for  $n$  from 1 to  $T$  do
    for  $i$  from 1 to  $Q$  do  $X_i(n) := X_i(n-1)$  od // Initializations
    for  $i$  from 1 to  $Q$  do  $Y_i(0) := 0$  od // Initializations
     $D_p := \sum_{j \in p^\bullet} [X_j(n) + Y_j(t)]$  // Initialization: departures from  $p$  up to time
  n
    for  $t$  from 0 to 1 by step  $\epsilon$  do
      for  $i$  from 1 to  $Q$  do  $Y_i(t) := Y_i(t-\epsilon)$  od // Initializations
      for  $i$  from 1 to  $Q$  do
        for all  $p$  in  $\bullet i$  do
           $C_p := \sum_{j \in \bullet p} X_j(\lceil n - \phi_j \rceil) + Y_j(t - \phi_j)$  // accumulated arrivals in  $p$ 
           $A_{p,i} := H_{p,i}(C_p) - H_{p,i}(D_p + Y_j(t))$  // tokens currently available
        on  $p$  for  $i$ 
          od
           $Y_i(t) := \min_{p \in \bullet i} \{A_{p,i}\} + Y_i(t)$  // # firings started by  $i$  at time  $n + t$ 
        od
      for  $i$  from 1 to  $Q$  do  $X_i(n) := X_i(n) + Y_i(1)$  od
    od
  end

```

Figure 6: Algorithm Limit(T).

Theorem 1. *The algorithm $\text{Limit}(T)$ computes $X^{LP}(T)$.*

Proof. Follows from the finiteness of $K(T)$ above and the fact that the non- ϵ firing times in S have a positive lower bound. \square

5.3 Stepwise computation

Under (C_3) , the limit policy can also be computed in an iterative way which treats the ϵ firing times as logical rather than physical time steps.

In order to decide which transitions to fire at time t , we construct the multi-set $E(t)$ (with multiplicities) of all transitions that will fire at time t under this policy.

First, we define $<_{\mathcal{P}}$ (which is a partial order by virtue of (C_1)) by

$$<_{\mathcal{P}} \stackrel{\text{def}}{=} \bigcup_{p \in \mathcal{P}} <_p .$$

Second, we define by $(t_k)_{k \in \mathbb{N}}$ the sequence of time instants of the form $\sum_{q \in \mathcal{U}} c_q \phi_q$, with $c_q \in \mathbb{N}$, in increasing order ($t_0 = 0$). Note that the sequence t_k is sporadic, i.e. does not contain any accumulation point; however, it contains all instants at which non-immediate transitions end firing.

By right continuity of the sequences X , we can define the marking at time t_{k-} using the formula:

$$M_p(t_{k-}) = M_p(t_{k-1}) + \sum_{i_j} X_{i_j}(t_{k-1} - \phi_{i_j}) - \sum_{o_j} X_{o_j}(t_{k-1}).$$

For any transition $q \in \mathcal{Q}$, we call the enabling degree of q the maximal number of times q can fire simultaneously at the current marking. $EA(M)$ denotes the set of transitions enabled in M . Set $M^0(t) := M(t_-)$ and define:

$$E_0(t)(q) \stackrel{\text{def}}{=} \begin{cases} \text{enabling degree of } q \text{ under } M^0(t) & \text{if } q \text{ is minimal for } <_{\mathcal{P}} \\ & \text{in } EA(M^0(t_-)) \\ 0 & \text{otherwise.} \end{cases}$$

The marking $M^i(t)$ is obtained from $M^{i-1}(t)$ by

1. firing all transitions in $E_{i-1}(t) \cap \mathcal{I}$ and
2. removing the tokens consumed by the transitions in $E_{i-1}(t) \cap \mathcal{U}$; since these transitions will take at least one time unit firing, the tokens they will produce play no role yet !

Formally, set

$$M_p^i(t) := M_p^{i-1}(t) + \sum_{q \in \bullet p \cap \mathcal{I}} E_{i-1}(t)(q) - \sum_{q \in p \bullet} E_{i-1}(t)(q).$$

We then define

$$E_i(t)(q) \stackrel{\text{def}}{=} \begin{cases} \text{enabling degree of } q \text{ under } M^i(t) & \text{if } q \text{ is minimal for } <_{\mathcal{P}} \\ & \text{in } EA(M^i(t)) \\ 0 & \text{otherwise.} \end{cases}$$

Finally,

$$E(t)(q) \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} E_i(t)(q), \quad (2)$$

$$M_p(t) \stackrel{\text{def}}{=} M_p(t_-) + \sum_{q \in \bullet p \cap \mathcal{I}} E(t)(q) - \sum_{q \in p \bullet} E(t)(q). \quad (3)$$

Note that by definition, $E(t)(q) = 0$ if $t \notin \{t_k, k \in \mathbb{N}\}$. E and M are well-defined because of the following

Lemma 5. *Under condition (C_3) , the multiset $E(t)$ is finite.*

Proof. The proof uses induction on k and i . At time $t_0 = 0$, $E(t_0)$ is finite since the marking is finite. Because of (C_3) , the firing of $E(t_0)$ will produce again a finite marking. Now, suppose the marking at t_k is finite. Then so is $E_0(t_k)$; if $E_{i-1}(t_k)$ is finite, its firing produces only finitely many tokens; thus $E_i(t_k)$ is finite as well. By (C_3) , there is an $n = n(k)$ such that

$$E(t_k) = \sum_{i=0}^n E_i(t)(q),$$

and thus $E(t)$ is finite. □

We now give an algorithm to stepwise compute the limiting behavior under (C_3) . The computation – as for all algorithms given in this paper – is done under the assumption that all non-immediate transitions have a firing time which is an integer number, which does not constitute a real loss of generality.

Theorem 2. *Under (C_3) , Steps(T) computes $X^{LP}(T)$.*

```

Steps(T)
  for  $i$  from 1 to  $Q$  do  $X_i(0) := 0$  od // Initializations
  for  $n$  from 1 to  $T$  do
    for  $i$  from 1 to  $Q$  do  $X_i(n) := X_i(n-1)$  od
    Compute  $E := \{ \text{enabled transitions} \}$ 
    while  $E$  is not empty do
      Order  $E$  according to the priorities.
      Let  $\{q_1, \dots, q_e\}$  be the smallest elements in  $E$ ;
      for  $k$  from 1 to  $e$  do  $X_{q_k}(n) := X_{q_k}(n) + 1$ 
      for all  $p$  in  $\bullet i$  do
         $C_p := \sum_{j \in \bullet p} X_j(n - \phi_j) + M_0(p)$  // accumulated arrivals in  $p$ 
         $D_p := \sum_{j \in p \bullet} [X_j(n) + Y_j(t)]$  // departures from  $p$  up to time  $n$ 
         $A_{p,i} := H_{p,i}(C_p) - H_{p,i}(D_p)$  // tokens currently available on  $p$  for  $i$ 
      od
      Compute  $E := \{ \text{enabled transitions} \}$ 
    od;
  od;
end;

```

Figure 7: Algorithm Steps(T).

Proof. The proof is done by induction on n , the number of steps on the trajectories followed by the net under both policies.

The first multi-event is the same on both trajectories. If we assume that this holds for all multi-events up to n , then the fact that the firing times are the same under each policy implies that the multi-event $n + 1$ is also the same under each policy, and takes place at the same instant. \square

Remark 3. *This algorithm – as well as the other algorithms below – have been designed for the case of routing decisions independent of time, i.e. $\nu_p(n, t) \equiv \nu_p(n)$. This causes no loss in generality since the algorithms can be adapted to the general case by the following procedure: introduce additional variables $t_{p,k}$ to denote, respectively, the time of the k -th arrival on place p ; in each round of **Steps** corresponding to n set $t_{p,k} := n$ iff*

$$\sum_{q \in \bullet p} X_q(n - \phi) < n \quad \text{and} \quad \sum_{q \in \bullet p} X_q(n) \geq n.$$

We have omitted these steps here and below to simplify the algorithms.

5.4 Generalizations

The convergence to zero of the firing times for immediate transitions proceeds, as it stands, in a very controlled way; what happens if more freedom is allowed ?

- An immediate generalization is obtained by setting the firing time of q_ϵ (the transition in \mathcal{S}^ϵ corresponding to an immediate transition q in \mathcal{S}) equal to $\alpha(q)\epsilon$, where $\alpha(q)$ is a positive number depending only on q . With some small adjustments, the results of this section still hold. A unique limit behaviour exists under the same conditions as for $\alpha \equiv 1$ and can be computed with an adjusted version of the algorithms given above.

Note that the *value* of the limit behavior $X^0(t, \alpha)$ may change if α is changed.

- If, on the other hand, one allows firing times ϵ_q for each immediate q varying independently of one another, one may lose the existence of limit behaviors even for well-behaved nets. Consider Figure 8. Depending on the convergence rates, one would hope to have either $X^0(1) = (1, 0)$ or $X^0(1) = (0, 1)$. However,

choosing the sequences $\epsilon_1(n)$ and $\epsilon_2(n)$ such that

$$\epsilon_1(n) = \begin{cases} 2 & : n = 1 \\ \frac{1}{n} & : n \text{ even} \\ \frac{1}{n-1} & : n \text{ odd, } n > 1 \end{cases}$$

$$\epsilon_2(n) = \begin{cases} \frac{1}{n-1} & : n \text{ even} \\ \frac{1}{n-1} & : n \text{ odd} \end{cases},$$

one has $\epsilon_1(n) < \epsilon_2(n)$ whenever n is even and $\epsilon_1(n) > \epsilon_2(n)$ whenever n is odd; therefore, the value of $X^{\epsilon(n)}$ alternates forever between $(0, 1)$ and $(1, 0)$, and no limit exists.

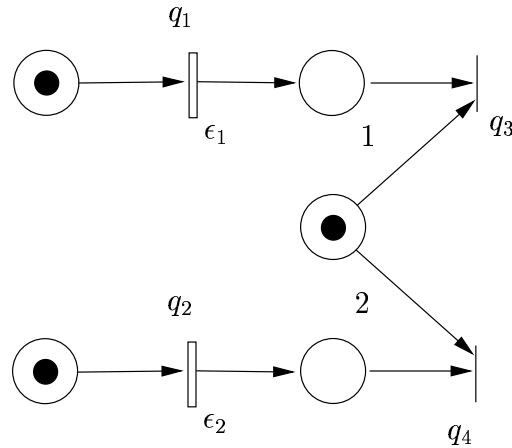


Figure 8: On generalised limit policies

6 Global Priority Policy (GPP)

6.1 Definition and Properties

In this section, we present an alternative to the limiting behavior (LP) by making the priority ordering more global. This will provide a completely different behavior

that, in a certain way, emphasizes the priorities of the transitions. However, under this firing policy some nets will remain ambiguous.

We construct an asymmetric relation that *depends on the current marking M* , between the transitions in the following way. The relation \triangleleft_M will depend on M . First, we construct an auxiliary Petri net \mathcal{S}_M – which we will discard after computing \triangleleft_M – by

- removing in \mathcal{S} all arcs entering a timed transition;
- and removing all arcs (i, p) such that $M(p) > 0$.

If two transitions q_1 and q_2 in \mathcal{S} are such that there exists an oriented path in \mathcal{S}_M between q_1 and q_2 then we say that, under the current marking, q_1 precedes q_2 (w.r.t. M).

Definition 4. *The relation \triangleleft_M is defined as follows:*

1. *If q_1 has priority over q_2 then $q_1 \triangleleft_M q_2$.*
2. *If q_1 precedes q_2 w.r.t. M , then $q_1 \triangleleft_M q_2$.*
3. *If $q_1 \triangleleft_M q_2 \triangleleft_M q_3$ then $q_1 \triangleleft_M q_3$.*

Remark 4. *The relation \triangleleft_M has a close relation with the topological order defined in [6]. However, the relation \triangleleft_M is dynamic and changes with the state of the net, while the topological order is structural, and in some sense, less refined.*

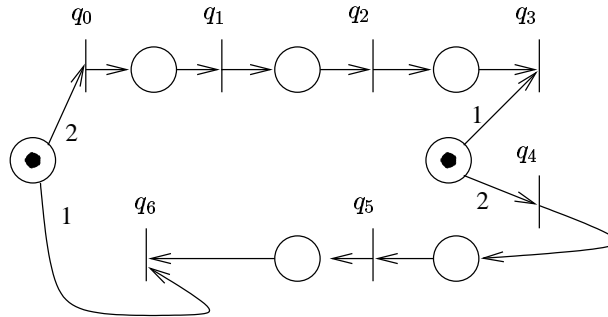


Figure 9: Example of a net where \triangleleft_M is not an order

In the net displayed in Figure 9 – where trivial routing is assumed –, \triangleleft_M is not an order because we have

$$q_0 \triangleleft_M q_1 \triangleleft_M q_2 \triangleleft_M q_3 \triangleleft_M q_4 \triangleleft_M q_5 \triangleleft_M q_6 \triangleleft_M q_0. \quad (4)$$

Remark 5. The relation \triangleleft_M heavily depends on the marking, therefore, it changes with the evolution of the net. There even exist examples for which \triangleleft_M is an order under some reachable marking and not an order under some other reachable marking, sooner or later in the evolution.

(C₄) We say that \mathcal{S} satisfies (C₄) iff, under any reachable marking M , \triangleleft_M is an order.

Lemma 6. We replace the priority ordering by \triangleleft_M for the firing rules (R₁) and (R₂). Then, under (C₄), for any immediate transitions q_1 and q_2 incomparable and minimal for \triangleleft_M , and any sequence W of immediate transitions, $M \xrightarrow{q_1 W} M'$ implies that q_2 is still enabled under M' .

Proof. The proof is done by induction on the length of W . First, if W is empty, the firing of q_1 will not involve tokens in places in $\bullet q_2$ since q_1 and q_2 are incomparable for $<_{\mathcal{P}}$.

Similarly, the firing of q_1 will not generate tokens in places that would enable transitions in competition with q_2 and with a higher priority. This is because q_1 and q_2 are incomparable for \triangleleft_M .

Finally, the firing of q_1 will not modify the routing of the tokens enabling q_2 .

Therefore, under marking M' , the order $\triangleleft_{A(M')}$ will still have q_2 minimal and q_2 will still be enabled under M' .

The general step of the proof is similar to the first step. Under marking M' , the transition q_2 is minimal for the order $\triangleleft_{A(M')}$ and will still be enabled under M' . \square

Theorem 3. Let \mathcal{S} be a net satisfying (C₄). Then, replacing the priority ordering by \triangleleft_M in the rules (R₁) and (R₂) makes the net non-ambiguous. We denote the resulting behavior by X^{GPP} .

Proof. By Lemma 6, any set θ of transitions pairwise incomparable and minimal with respect to \triangleleft_M can fire in arbitrary order with no consequence whatsoever on the future behavior of the net. This defines a unique behavior of the net. \square

Remark 6. It is an open problem to give a good characterization of Condition (C₄).

Remark 7. *If Condition (C_2) is satisfied and if the topological ordering (as defined in [6]) is compatible with the priority ordering then, (C_4) is satisfied. Indeed, the relation \triangleleft_M is always included in the union of the priority ordering and the topological ordering and does not contain cycles as long as the two latter orderings are compatible.*

6.2 Computation of the behavior under GPP

In the algorithm `Global(T)`, we denote the initial marking by M_0 and assume $\mathcal{Q} = \{1, \dots, Q\}$ and $\mathcal{P} = \{1, \dots, P\}$. During each round, the algorithm uses a different numbering of \mathcal{Q} to choose the next transition to be inspected. In order to maintain the ordering of the components of X , the algorithm has to account for the change of numbering.

Theorem 4. *The algorithm `Global(T)` computes $X^{GPP}(T)$ whenever it terminates, in particular, if \mathcal{S} satisfies (C_3) .*

Proof. By Lemma 6 and induction, any linear ordering will lead to a unique firing count for finite firing sequences. So the crucial condition is the termination of the `While`-loop of `Global`, which is guaranteed under (C_3) . \square

6.3 Comparing the Policies

Remark 8. *Conditions (C_2) and (C_4) are incomparable; there exist nets satisfying one but not the other condition. For instance, the net of Figure 9 satisfies (C_2) but not (C_4) . On the other hand, a net formed by one cycle of immediate transitions with one token (see figure 11) does not satisfy (C_2) or C_3 ; however, this net is nonambiguous ($X_q(t) = \infty$ for all $t > 0$ and all transitions q) and satisfies (C_4) .*

Remark 9. *In the first example of Figure 2, (LP) yields the behavior $X_0(1) = 1, X_1(1) = 0, X_2(1) = 0, X_3(1) = 1$, and (GPP) leads to the behavior $X_0(1) = 1, X_1(1) = 1, X_2(1) = 1, X_3(1) = 0$. Both policies solve the non-determinism in a different way. It is in general hard to give preference for one policy or the other since depending on the case at hand one or the other looks more “natural”.*

Remark 10. *Let us return to the example of an ambiguous net in Figure 3. It is possible to adjust the routing in such a way that ambiguity is eliminated. Any unequivocal routing will achieve this; sometimes, this condition is not even necessary: for the net in Figure 3, it suffices to have some n such that only the n th token having*

```

Global(T)
begin
  for  $i$  from 1 to  $Q$  do
     $X_i(0) := 0$ 
    for  $p$  from 1 to  $P$  do       $A_{p,i} := H_{p,i}(M_0(p))$ 
    od
  od
  Compute  $E := \{ \text{Transitions enabled in } A \}$ .           // Initializations
  for  $n$  from 1 to  $T$  do
    Compute  $\triangleleft_A$  and a permutation  $\sigma$  of  $Q$  that orders the elements of  $Q$ 
    according to some linear extension of  $\triangleleft_A$ .
    While  $E \neq \emptyset$  do
      for  $i$  from 1 to  $Q$  do  $X_{\sigma^{-1}(i)}(n) := X_{\sigma^{-1}(i)}(n-1)$  od
      for  $i$  from 1 to  $Q$  do
        for all  $p \in \bullet\sigma^{-1}(i)$  do
           $C_p := \sum_{j \in \bullet p} X_j(n - \phi_j)$            // accumulated number of arrivals on  $p$ 
           $D_p := \sum_{j \in p\bullet} X_j(n)$                    // departures from  $p$  up to time  $n$ 
           $A_{p,\sigma^{-1}(i)} := H_{p,\sigma^{-1}(i)}(C_p) - H_{p,\sigma^{-1}(i)}(D_p)$  // tokens currently available
                                                                // on  $p$  for  $\sigma^{-1}(i)$ 
        od
         $X_{\sigma^{-1}(i)}(n) := \min_{p \in \bullet\sigma^{-1}(i)} \{ A_{\sigma^{-1}(i)} \} + X_{\sigma^{-1}(i)}(n)$  // firings started by  $\sigma^{-1}(i)$ 
                                                                // before or at  $n$ 
      od
      Compute  $E := \{ \text{Transitions enabled in } A \}$ .
    od
  od
end

```

Figure 10: Algorithm Global(T).

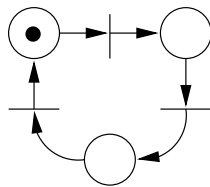


Figure 11: A net satisfying C_4 and not C_3 .

to choose between a and q_4 is routed unequivocally to a ; the net will then stop in that state until a can fire. However, we have used here our knowledge of the semantics: we were sure never to have more than one token on the immediate cycle. In the general case, where the number of tokens in the subnet spanned by the immediate transitions may not even be bounded, we cannot find such a routing a priori, by structural considerations alone.

7 Conclusion

We have shown that the initial data of a Petri net \mathcal{S} – structure, initial marking, priority ordering, routing function, transition durations – suffice to ensure uniqueness of \mathcal{S} 's behavior for a large class of nets given by structural conditions. Within this class, nondeterminism is impossible once those data are fixed; stochastic behavior can be obtained by “randomizing” the timing, the priorities, or the routing. Non-ambiguity may hold also if cycles of immediate transitions exist but the net is controlled; if, however, an uncontrolled independent cycle of immediate transitions exist, the state of any place within that cycle is undefined for any time t later than the first enabling of transitions in the cycle. The firing numbers of transitions depending entirely on such states will then be ambiguous except for special circumstances, such as unequivocal routing. Figure 12 shows the relations between the different conditions; for instance, the net displayed in Figure 3 without transition b is an example of a net that violates Condition (C_3) but for which the limit policies exist. As Figure 12 shows, the examples presented in this paper comprise almost all possible cases, i. e. combinations of conditions studied in this paper.

Also, two firing strategies to resolve conflicts between alternative behaviors have been presented; they are defined on a large, common class of well-behaved nets given by (C_1) , (C_3) and (C_4) and, possibly, beyond that class. However they substantially differ and the choice can only be made on a case by case basis.

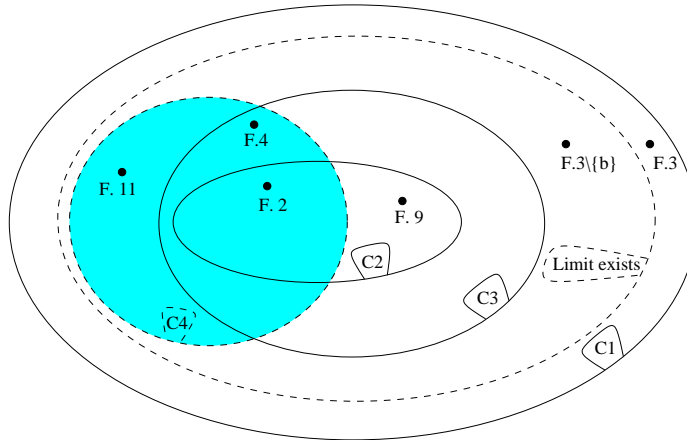


Figure 12: Illustrations of connections between Conditions C_1 , C_2 , C_3 and C_4

The non-ambiguity studied here is taken with respect to $X(t)$. Non-ambiguity (up to isomorphism) of the *process* of \mathcal{S} requires in addition the possibility to uniquely recover the transitions that put a given token onto a place p (whereas the routing determines the transition through which the token leaves p).

References

- [1] F. Baccelli, S. Foss, and B. Gaujal. Free choice nets, an algebraic approach. *IEEE Transaction on Automatic Control*, 41(12), 1996.
- [2] F. Bause and P.S. Kritzinger. *Stochastic Petri Nets, An introduction to the Theory*. Verlag Vieweg, 1996.
- [3] E. Best and C. Fernández. *Nonsequential Processes. A Petri Net view*, volume 13 of *EATCS Monographs*. Springer, Berlin, New York, 1988.
- [4] J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge University Press, 1995.
- [5] J. Esparza. Model Checking Using Net Unfoldings. *Science of Computer Programming*, 23:151–195, 1994.

- [6] B. Gaujal, A. Jean-Marie, and G. Siegel. Non-ambiguous petri nets and their application to the modeling of atm switches. In *WODES*, Cagliari, Italy, 1998. IEE.
- [7] W. Reisig. *Petri Nets*, volume 4 of *EATCS Monographs on Theoretical Computer Science*. Springer, Berlin, New York, 1985.



Unit e de recherche INRIA Lorraine, Technop ole de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS L ES NANCY
Unit e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit e de recherche INRIA Rh one-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

 diteur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399