Synthesis and Analysis of Product-form Petri Nets

S. Haddad[1], J. Mairesse[2], H-T. Nguyen[2]

[1] ENS Cachan, LSV, CNRS UMR 8643, INRIA, Cachan, France
haddad@lsv.ens-cachan.fr
[2] Université Paris 7, LIAFA, CNRS UMR 7089, Paris, France
{mairesse,ngthach}@liafa.jussieu.fr

Abstract. For a large Markovian model, a "product form" is an explicit description of the steady-state behaviour which is otherwise generally untractable. Being first introduced in queueing networks, it has been adapted to Markovian Petri nets. Here we address three relevant issues for product-form Petri nets which were left fully or partially open: (1) we provide a sound and complete set of rules for the synthesis; (2) we characterise the exact complexity of classical problems like reachability; (3) we introduce a new subclass for which the normalising constant (a crucial value for product-form expression) can be efficiently computed.

Keywords: Petri nets, product-form, synthesis, complexity analysis, reachability, normalising constant

1 Introduction

Product-form for stochastic models. Markovian models of discrete events systems are powerful formalisms for modelling and evaluating the performances of such systems. The main goal is the equilibrium performance analysis. It requires to compute the stationary distribution of a continuous time Markov process derived from the model. Unfortunately the potentially huge (sometimes infinite) state space of the models often prevents the modeller from computing explicitly this distribution. To cope with the issue, one can forget about exact solutions and settle for approximations, bounds, or even simulations. The other possibility is to focus on subclasses for which some kind of explicit description is indeed possible. In this direction, the most efficient and satisfactory approach may be the *product-form* method: for a model composed of modules, the stationary probability of a global state may be expressed as a product of quantities depending only on local states divided by a *normalising constant*.

Such a method is applicable when the interactions between the modules are "weak". This is the case for queueing networks where the interactions between queues are described by a random routing of clients. Various classes of queueing networks with product-form solutions have been exhibited [11, 3, 12]. Moreover efficient algorithms have been designed for the computation of the normalising constant [18].

Product-form Petri nets. Due to the explicit modelling of competition and synchronisation, the Markovian Petri nets formalism [1] is an attractive modelling paradigm. Similarly to queueing networks, product-form Markovian Petri Nets were introduced to cope with the combinatorial explosion of the state space. Historically, works started with purely behavioural properties (i.e. by an analysis of the reachability graph) as in [13], and then progressively moved to more and more structural characterisations [14, 10]. Building on the work of [10], the authors of [9] establish the first purely structural condition for which a product form exists and propose a polynomial time algorithm to check for the condition, see also [15] for an alternative characterisation. These nets are called Π^2-nets.

Open issues related to product-form Petri nets.

- From a modelling point of view, it is more interesting to design specific types of Petri nets by modular constructions rather than checking a posteriori whether a net satisfies the specification. For instance, in [7], a sound and complete set of rules is proposed for the synthesis of live and bounded free-choice nets. Is it possible to get an analog for product-form Petri nets?
- From a qualitative analysis point of view, it is interesting to know the complexity of classical problems (reachability, coverability, liveness, etc.) for a given subclass of Petri nets and to compare it with that of general Petri nets. For product-form Petri nets, partial results were presented in [9] but several questions were left open. For instance, the reachability problem is PSPACE-complete for safe Petri nets but in safe product-form Petri nets it is only proved to be NP-hard in [9].
- From a quantitative analysis point of view, an important and difficult issue is the computation of the normalising constant. Indeed, in product-form Petri nets, one can directly compute relative probabilities (e.g. available versus unavailable service), but determining absolute probabilities requires to compute the normalising constant (i.e. the sum over reachable states of the relative probabilities). In models of queueing networks, this can be efficiently performed using dynamic programming. In Petri nets, it has been proved that the efficient computation is possible when the linear invariants characterise the set of reachable markings [6]. Unfortunately, all the known subclasses of product-form nets that fulfill this characterisation are models of queueing networks!

Our contribution. Here we address the three above issues. In Section 3, we provide a set of sound and complete rules for generating any Π^2-net. We also use these rules for transforming a general Petri net into a related product-form Petri net. In Section 4, we solve relevant complexity issues. More precisely, we show that the reachability and liveness problems are PSPACE-complete for safe product-form nets and that the coverability problem is EXPSPACE-complete for general product-form nets. From these complexity results, we conjecture that the problem of computing the normalising constant does not admit an efficient solution for the general class of product-form Petri nets. However, in Section 5, we introduce a large subclass of product-form Petri nets, denoted Π^3-nets, for which the normalising constant can be efficiently computed. We emphasise

that contrary to all subclasses related to queueing networks, Π^3-nets may admit *spurious* markings (i.e. that fufill the invariants while being unreachable).

The above results may change our perspective on product-form Petri nets. It is proved in [15] that the intersection of free-choice and product-form Petri nets is the class of Jackson networks [11]. This may suggest that the class of product-form Petri nets is somehow included in the class of product-form queueing networks. In the present paper, we refute this belief in two ways. First by showing that some classical problems are as complex for product-form Petri nets as for general Petri nets whereas they become very simple for product-form queueing networks. Second by exhibiting the class of Π^3-nets, see the above discussion.

A version of the present paper including proofs can be found on arXiv.

Notations. We often denote a vector $u \in \mathbb{R}^S$ by $\sum_s u(s)s$. The *support* of vector u is the subset $S' \equiv \{s \in S \mid u(s) \neq 0\}$.

2 Petri nets, product-form nets, and Π^2-nets

Definition 2.1 (Petri net) *A* Petri net *is a 5-tuple* $\mathcal{N} = (P, T, W^-, W^+, m_0)$ *where:*

- P *is a finite set of* places*;*
- T *is a finite set of* transitions*, disjoint from* P*;*
- W^-*, resp.* W^+*, is a* $P \times T$ *matrix with coefficients in* $\mathbb{N}$*;*
- $m_0 \in \mathbb{N}^P$ *is the* initial marking*.*

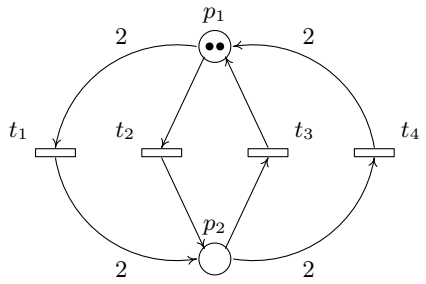
Below, we also call *Petri net* the unmarked quadruple (P, T, W^-, W^+). The presence or absence of a marking will depend on the context.

A Petri net is represented in Figure 1. The following graphical conventions are used: places are represented by circles and transitions by rectangles. There is an arc from $p \in P$ to $t \in T$ (resp. from $t \in T$ to $p \in P$) if $W^+(p,t) > 0$ (resp. $W^-(p,t) > 0$), and the weight $W^+(p,t)$ (resp. $W^-(p,t)$) is written above the corresponding arc except when it is equal to 1 in which case it is omitted. The initial marking is materialised: if $m_0(p) = k$, then k tokens are drawn inside the circle p. Let $P' \subset P$ and m be a marking then $m(P')$ is defined by $m(P') \equiv \sum_{p \in P'} m(p)$.

The matrix $W = W^+ - W^-$ is the *incidence matrix* of the Petri net. The *input bag* ${}^\bullet t$ (resp. *output bag* $t^\bullet$) of the transition t is the column vector of W^- (resp. W^+) indexed by t. For a place p, we define ${}^\bullet p$ and $p^\bullet$ similarly. A *T-semi-flow* (resp. *S-semi-flow*) is a $\mathbb{Q}$-valued vector v such that $W.v = 0$ (resp. $v.W = 0$).

A *symmetric* Petri net is a Petri net such that: $\forall t \in T,\ \exists t^- \in T,\ \ {}^\bullet t = (t^-)^\bullet, t^\bullet = {}^\bullet t^-$. A *state machine* is a Petri net such that: $\forall t \in T,\ |{}^\bullet t| = |t^\bullet| = 1$.

Definition 2.2 (Firing rule) *A transition* t *is* enabled *by the marking* m *if* $m \geq {}^\bullet t$ *(denoted by* $m \xrightarrow{t}$*); an enabled transition* t *may* fire *which transforms the marking* m *into* $m - {}^\bullet t + t^\bullet$*, denoted by* $m \xrightarrow{t} m' = m - {}^\bullet t + t^\bullet$*.*



$$W = \begin{pmatrix} -2 & -1 & 1 & 2 \\ 2 & 1 & -1 & -2 \end{pmatrix} .$$
$$m_0 = (2, 0) .$$

Fig. 1. Petri net.

A marking m' is *reachable* from the marking m if there exists a *firing sequence* $\sigma = t_1 \dots t_k$ $(k \geq 0)$ and a sequence of markings $m_1, \dots, m_{k-1}$ such that $m \xrightarrow{t_1} m_1 \xrightarrow{t_2} \cdots \xrightarrow{t_{k-1}} m_{k-1} \xrightarrow{t_k} m'$. We write in a condensed way: $m \xrightarrow{\sigma} m'$.

We denote by $\mathcal{R}(m)$ the set of markings which are reachable from the marking m. The *reachability graph* of a Petri net with initial marking m_0 is the directed graph with nodes $\mathcal{R}(m_0)$ and arcs $\{(m, m') | \exists t \in T : m \xrightarrow{t} m'\}$.

Given $(\mathcal{N}, m_0)$ and m_1, the *reachability problem* is to decide if $m_1 \in \mathcal{R}(m_0)$, and the *coverability problem* is to decide if $\exists m_2 \in \mathcal{R}(m_0), m_2 \geq m_1$.

A Petri net $(\mathcal{N}, m_0)$ is *live* if every transition can always be enabled again, that is: $\forall m \in \mathcal{R}(m_0), \forall t \in T,\ \exists m' \in \mathcal{R}(m),\ m' \xrightarrow{t}$. A Petri net $(\mathcal{N}, m_0)$ is bounded if $\mathcal{R}(m_0)$ is finite. It is *safe* or *1-bounded* if: $\forall m \in \mathcal{R}(m_0),\ m(p) \leq 1$.

2.1 Product-form Petri nets

There exist several ways to define timed models of Petri nets, see [2]. We consider the model of Markovian Petri nets with *race policy*. Roughly, with each enabled transition is associated a "countdown clock" whose positive initial value is set at random according to an exponential distribution whose rate depends on the transition. The first transition to reach 0 fires, which may enable new transitions and start new clocks.

Definition 2.3 (Markovian PN) *A* Markovian Petri net (with race policy) *is a Petri net equipped with a set of* rates $(\mu_t)_{t \in T}$, $\mu_t \in \mathbb{R}_+ \setminus \{0\}$. *The firing time of an enabled transition t is exponentially distributed with parameter μ_t. The marking evolves as a continuous-time jump Markov process with state space $\mathcal{R}(m_0)$ and infinitesimal generator $Q = (q_{m,m'})_{m,m' \in \mathcal{R}(m_0)}$, given by:*

$$\forall m,\ \forall m' \neq m,\ q_{m,m'} = \sum_{t : m \xrightarrow{t} m'} \mu_t, \qquad \forall m,\ q_{m,m} = - \sum_{m' \neq m} q_{m,m'} . \tag{2.1}$$

W.l.o.g., we assume that there is no transition t such that ${}^\bullet t = t^\bullet$. Indeed, the firing of such a transition does not modifiy the marking, so its removal does not modify the infinitesimal generator. We also assume that $({}^\bullet t_1, t_1^\bullet) \neq ({}^\bullet t_2, t_2^\bullet)$

for all transitions $t_1 \neq t_2$. Indeed, if it is not the case, the two transitions may be replaced by a single one with the summed rate.

An *invariant measure* is a non-trivial solution ν to the *balance equations*: $\nu Q = 0$. A *stationary measure (distribution)* π is an invariant probability measure: $\pi Q = 0$, $\sum_m \pi(m) = 1$.

Definition 2.4 (Product-form PN) *A Petri net is a* product-form Petri net *if for all rates $(\mu_t)_{t\in T}$, the corresponding Markovian Petri net admits an invariant measure ν satisfying:*

$$\exists (u_p)_{p\in P}, u_p \in \mathbb{R}_+, \quad \forall m \in \mathcal{R}(m_0), \qquad \nu(m) = \prod_{p\in P} u_p^{m_p} \,. \tag{2.2}$$

The existence of ν satisfying (2.2) implies that the marking process is irreducible (in other words, the reachability graph is strongly connected). In (2.2), the mass of the measure, i.e. $\nu(\mathcal{R}(m_0)) = \sum_m \nu(m)$, may be either finite or infinite. For a bounded Petri net, the mass is always finite. But for an unbounded Petri net, the typical situation will be as follows: structural conditions on the Petri net will ensure that the Petri net is a product-form one. Then, for some values of the rates, ν will have an infinite mass, and, for others, ν will have a finite mass. In the first situation, the marking process will be either transient or recurrent null (unstable case). In the second situation, the marking process will be positive recurrent (stable or ergodic case).

When the mass is finite, we call $\nu(\mathcal{R}(m_0))$ the *normalising constant*. The probability measure $\pi(\cdot) = \nu(\mathcal{R}(m_0))^{-1}\nu(\cdot)$ is the unique stationary measure of the marking process. Computing explicitly the normalising constant is an important issue, see Section 5.

The goal is now to get sufficient conditions for a Petri net to be of product-form. To that purpose, we introduce three notions: *weak reversibility*, *deficiency*, and *witnesses*.

Let (N, m_0) be a Petri net. The set of *complexes* is defined by $\mathcal{C} = \{{}^\bullet t \mid t \in T\} \cup \{t^\bullet \mid t \in T\}$. The *reaction graph* is the directed graph whose set of nodes is $\mathcal{C}$ and whose set of arcs is $\{({}^\bullet t, t^\bullet) | t \in T\}$.

Definition 2.5 (Weak reversibility: Π-nets) *A Petri net is* weakly reversible (WR) *if every connected component of its reaction graph is strongly connected. Weakly reversible Petri nets are also called* Π-nets.

The notion and the name "WR" come from the chemical literature. In the Petri net context, it was introduced in [4, Assumption 3.2] under a different name and with a slightly different but equivalent formulation. WR is a strong constraint. It should not be confused with the classical notion of "reversibility" (the marking graph is strongly connected). In particular, WR implies reversibility! Observe that all symmetric Petri nets are WR.

The notion of deficiency is due to Feinberg [8].

Definition 2.6 (Deficiency) *Consider a Petri net with incidence matrix W and set of complexes $\mathcal{C}$. Let ℓ be the number of connected components of the reaction graph. The* deficiency *of the Petri net is defined by: $|\mathcal{C}| - \ell - \mathrm{rank}(W)$.*

The notion of witnesses appears in [9].

Definition 2.7 (Witness) *Let c be a complex. A* witness *of c is a vector $wit(c) \in \mathbb{Q}^P$ such that for all transition t:*

$$\begin{cases} wit(c) \cdot W(t) = -1 & \text{if } {}^\bullet t = c \\ wit(c) \cdot W(t) = 1 & \text{if } t^\bullet = c \\ wit(c) \cdot W(t) = 0 & \text{otherwise}\,, \end{cases}$$

where $W(t)$ denotes the column vector of W indexed by t.

Examples. Consider the Petri net of Figure 1. First, it is WR. Indeed, the set of complexes is $\mathcal{C} = \{p_1, p_2, 2p_1, 2p_2\}$ and the reaction graph is:

$$p_1 \leftrightarrow p_2\,,\ 2p_1 \leftrightarrow 2p_2\,,$$

with two connected components which are strongly connected. Second, the deficiency is 1 since $|\mathcal{C}| = 4$, $\ell = 2$, and $\mathrm{rank}(W) = 1$. Last, one can check that none of the complexes admit a witness.

Consider now the Petri net of Figure 3. It is WR and it has deficiency 0.

Proposition 2.8 (deficiency 0 $\iff$ witnesses, in [15, Prop. 3.9]) *A Petri net admits a witness for each complex iff it has deficiency 0.*

Next Theorem is a combination of Feinberg's Deficiency zero Theorem [8] and Kelly's Theorem [12, Theorem 8.1]. (It is proved under this form in [15, Theorem 3.8].)

Theorem 2.9 (WR + deficiency 0 $\implies$ product-form) *Consider a Markovian Petri net with rates $(\mu_t)_{t\in T}$, $\mu_t > 0$, and assume that the underlying Petri net is WR and has deficiency 0. Then there exists $(u_p)_{p\in P}$, $u_p > 0$, satisfying the equations:*

$$\forall c \in \mathcal{C}, \qquad \prod_{p:c_p\neq 0} u_p^{c_p} \sum_{t:{}^\bullet t = c} \mu_t = \sum_{t:t^\bullet = c} \mu_t \prod_{p:{}^\bullet t_p \neq 0} u_p^{{}^\bullet t_p}\,. \tag{2.3}$$

The marking process has an invariant measure ν s.t.: $\forall m,\ \nu(m) = \prod_{p\in P} u_p^{m_p}$.

Checking the WR, computing the deficiency, determining the witnesses, and solving the equations (2.3), all of these operations can be performed in polynomial-time, see [9, 15].

Summing up the above, it seems worth to isolate and christen the class of nets which are WR and have deficiency 0. We adopt the terminology of [9].

Definition 2.10 (Π^2-net) *A* Π^2-net *is a Petri net which is WR and has deficiency 0.*

3 Synthesis and regulation of Π^2-nets

The reaction graph, defined in Section 2.1, may be viewed as a Petri net (state machine). Let us formalise this observation. The *reaction Petri net* of $\mathcal{N}$ is the Petri net $\mathcal{A} = (\mathcal{C}, T, \overline{W}^-, \overline{W}^+)$, with for every $t \in T$:
- $\overline{W}^-({}^\bullet t, t) = 1$ and $\forall u \neq {}^\bullet t,\ \overline{W}^-(u,t) = 0$
- $\overline{W}^+(t^\bullet, t) = 1$ and $\forall u \neq t^\bullet,\ \overline{W}^+(u,t) = 0$

3.1 Synthesis

In this subsection, we consider unmarked nets. We define three rules that generate all the Π^2-nets. The first rule adds a strongly connected state machine.

Definition 3.1 (State-machine insertion) *Let $\mathcal{N} = (P_\mathcal{N}, T_\mathcal{N}, W^-_\mathcal{N}, W^+_\mathcal{N})$ be a net and $\mathcal{M} = (P_\mathcal{M}, T_\mathcal{M}, W^-_\mathcal{M}, W^+_\mathcal{M})$ be a strongly connected state machine disjoint from $\mathcal{N}$. The rule* `S-add` *is always applicable and $\mathcal{N}' =$ `S-add`$(\mathcal{N}, \mathcal{M})$ is defined by:*
- $P' = P_\mathcal{N} \sqcup P_\mathcal{M}$, $T' = T_\mathcal{N} \sqcup T_\mathcal{M}$;
- $\forall p \in P_\mathcal{N},\ \forall t \in T_\mathcal{N},\ W'^-(p,t) = W^-_\mathcal{N}(p,t),\ W'^+(p,t) = W^+_\mathcal{N}(p,t)$;
- $\forall p \in P_\mathcal{M},\ \forall t \in T_\mathcal{M},\ W'^-(p,t) = W^-_\mathcal{M}(p,t),\ W'^+(p,t) = W^+_\mathcal{M}(p,t)$;
- *All other entries of W'^- and W'^+ are null.*

The second rule consists in substituting to a complex c the complex $c + \lambda p$. However in order to be applicable some conditions must be fulfilled. The first condition requires that $c(p) + \lambda$ is non-negative. The second condition ensures that the substitution does not modify the reaction graph. The third condition preserves deficiency zero. Observe that the third condition can be checked in polynomial time, indeed it amounts to solving a system of linear equations in $\mathbb{Q}$ for every complex.

Definition 3.2 (Complex update) *Let $\mathcal{N} = (P, T, W^-, W^+)$ be a Π^2-net, c be a complex of $\mathcal{N}$, $p \in P$, $\lambda \in \mathbb{Z}^*$. The rule* `C-update` *is applicable when:*
1. $\lambda + c(p) \geq 0$;
2. $c + \lambda p$ *is not a complex of* $\mathcal{N}$;
3. *For every complex c' there exists a witness $wit(c')$ s.t. $wit(c')(p) = 0$.*

The resulting net $\mathcal{N}' =$ `C-update`$(\mathcal{N}, c, p, \lambda)$ is defined by:
- $P' = P$, $T' = T$;
- $\forall t \in T$ *s.t.* $W^-(t) \neq c,\ W'^-(t) = W^-(t)$, $\forall t \in T$ *s.t.* $W^-(t) = c,\ W'^-(t) = c + \lambda p$
- $\forall t \in T$ *s.t.* $W^+(t) \neq c,\ W'^+(t) = W^-(t)$, $\forall t \in T$ *s.t.* $W^+(t) = c,\ W'^+(t) = c + \lambda p$.

The last rule "cleans" the net by deleting an isolated place. We call this operation `P-delete`.

Definition 3.3 (Place deletion) *Let $\mathcal{N} = (P, T, W^-, W^+)$ be a net and let p be an isolated place of $\mathcal{N}$, i.e. $W^-(p) = W^+(p) = 0$. Then the rule* `P-delete` *is applicable and $\mathcal{N}' =$ `P-delete`$(\mathcal{N}, p)$ is defined by:*

– $P' = P \setminus \{p\}$, $T' = T$;
– $\forall q \in P',\ W'^{-}(q) = W^{-}(q),\ W'^{+}(q) = W^{+}(q)$.

Proposition 3.4 shows the interest of the rules for synthesis of Π^2-nets.

Proposition 3.4 (Soundness and Completeness) *Let $\mathcal{N}$ be a Π^2-net.*
– *If a rule* `S-add`, `C-update` *or* `P-delete` *is applicable on $\mathcal{N}$ then the resulting net is still a Π^2-net.*
– *The net $\mathcal{N}$ can be obtained by successive applications of the rules* `S-add`, `C-update`, `P-delete` *starting from the empty net.*

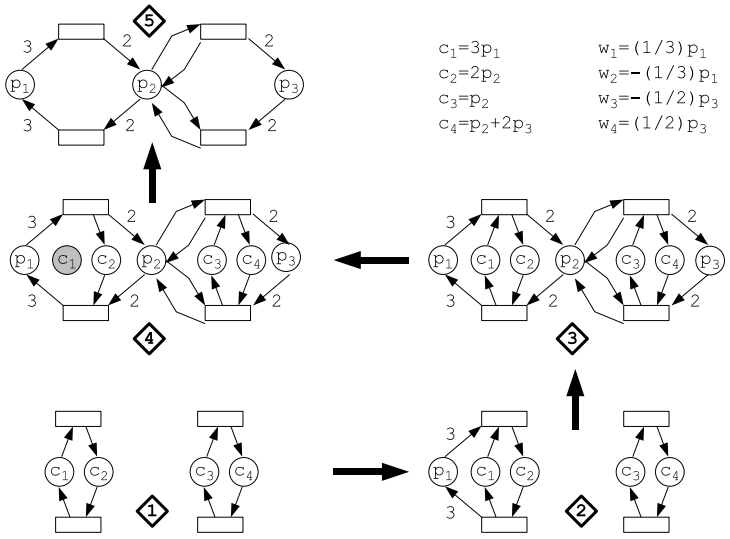


Fig. 2. How to synthetise a Π^2-net.

We illustrate the synthesis process using our rules on the net numbered 5 in Figure 2. We have also indicated on the right upper part of this figure, the four complexes and their witnesses. Since the reaction Petri graph of this net has two state machines, we start by creating it using twice the insertion of a state machine (net 1). Then we add the place p_1 (a particular state machine). We update the complex c_1 (the single one where p_1 appears in the original net) by adding $3p_1$ (net 2). The new complex cannot appear elsewhere due to the presence of c_1. Iterating this process, we obtain the net 3. Observe that this net is a fusion (via T the set of transitions) of the original net and its reaction Petri net. We now iteratively update the complexes. The net 4 is the result of transforming $c_1 + 3p_1$ into $3p_1$. This transformation is applicable since all the

complexes are witnessed by witnesses of the original net. For instance, $c_1 + 3p_1$ is witnessed by $(1/3)p_1$. Once c_1 is isolated, we delete it. Iterating this process yields the original net.

For modelling purposes, we could define more general rules like the refinement of a place by a strongly connected state machine. Here the goal was to design a minimal set of rules.

3.2 From non Π^2-nets to Π^2-nets

Below we propose a procedure which takes as input any Petri net and returns a Π^2-net. The important disclaimer is that the resulting net, although related to the original one, has a different structural and timed behaviour. So it is up to the modeller to decide if the resulting net satisfies the desired specifications. In case of a positive answer, the clear gain is that all the associated Markovian Petri nets have a product form.

Consider a Petri net $\mathcal{N} = (P, T, W^-, W^+, m_0)$ with set of complexes $\mathcal{C}$. Assume that $\mathcal{N}$ is not WR. For each transition t, add a reverse transition t^- such that ${}^\bullet t^- = t^\bullet$ and $(t^-)^\bullet = {}^\bullet t$ (unless such a transition already exists). The resulting net is WR. In the Markovian Petri net, the added reverse transitions can be given very small rates, to approximate more closely the original net.

Now, to enforce deficiency 0, the idea is to compose a general Petri net with its reaction graph as in the illustration of Proposition 3.4.

Definition 3.5 *Consider a Petri net $\mathcal{N} = (P, T, W^-, W^+, m_0)$. Let $\overline{m}_0$ be an initial marking for the reaction Petri net $\mathcal{A}$. The* regulated *Petri net associated with $\mathcal{N}$ is defined as follows:*

$$\mathcal{A} \odot \mathcal{N} = \big(P \sqcup \mathcal{C}, T, \widetilde{W}^-, \widetilde{W}^+, (m_0, \overline{m}_0)\big), \qquad \widetilde{W}^- = \begin{bmatrix} W^- \\ \overline{W}^- \end{bmatrix}, \quad \widetilde{W}^+ = \begin{bmatrix} W^+ \\ \overline{W}^+ \end{bmatrix}.$$

Proposition 3.6 *The regulated Petri net $\mathcal{A} \odot \mathcal{N}$ is WR iff $\mathcal{N}$ is WR. The regulated Petri net $\mathcal{A} \odot \mathcal{N}$ has deficiency 0.*

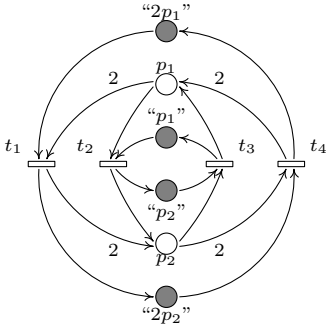


Fig. 3. Regulated Petri net associated with the Petri net of Fig 1.

The behaviours of the original and regulated Petri nets are different. In particular, the regulated Petri net is bounded, even if the original Petri net is unbounded. Roughly, the regulation imposes some control on the firing sequences. Consider the example of Figures 1 (original net) and 3 (regulated net). The transitions t_1 and t_4 belong to the same simple circuit in the reaction graph. Let w be an arbitrary firing sequence. The quantity $|w|_{t1} - |w|_{t4}$ is unbounded for the original net, and bounded for the regulated net.

4 Complexity analysis of Π^2-nets

All the nets that we build in this section are symmetric hence WR. For every depicted transition t, the reverse transition exists (sometimes implicitly) and is denoted t^-. It is well known that reachability and liveness of safe Petri nets are PSPACE-complete [5]. In [9], it is proved that reachability and liveness are PSPACE-hard for safe Π-nets and NP-hard for safe Π^2-nets. Next theorem and its corollary improve on these results by showing that the problem is not easier for safe Π^2-nets than for general safe Petri nets.

Theorem 4.1 *The reachability problem for safe Π^2-nets is* PSPACE*-complete.*

Proof. Our proof of PSPACE-hardness is based on a reduction from the QSAT problem [17]. QSAT consists in deciding whether the following formula is true

$$\varphi \equiv \forall x_n \exists y_n \forall x_{n-1} \exists y_{n-1} \ldots \forall x_1 \exists y_1 \psi$$

where ψ is a propositional formula over $\{x_1, y_1 \ldots, x_n, y_n\}$ in conjunctive normal form with at most three literals per clause.

Observe that in order to check the truth of φ, one must check the truth of ψ w.r.t. the 2^n interpretations of $x_1, \ldots, x_n$ while the corresponding interpretation of any y_i must only depend of the interpretation of $\{x_n, \ldots, x_i\}$.

Counters modelling. First we design a Π^2-net $\mathcal{N}_{cnt}$ that "counts" from 0 to $2^k - 1$. This net is defined by:

- $P = \{p_0, \ldots, p_{k-1}, q_0, \ldots, q_{k-1}\}$;
- $T = \{t_0, \ldots, t_{k-1}\}$;
- For every $0 \leq i < k$, ${}^\bullet t_i = p_i + \sum_{j<i} q_j$ and $t_i^\bullet = q_i + \sum_{j<i} p_j$;
- For every $0 \leq i < k$, $m_0(p_i) = 1$ and $m_0(q_i) = 0$.

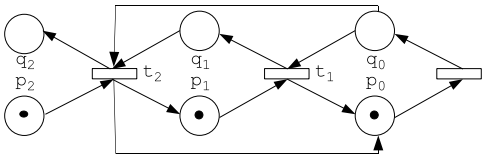
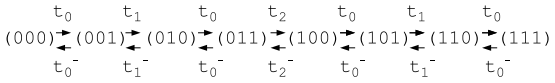


Fig. 4. A 3-bit counter (without the reverse transitions).

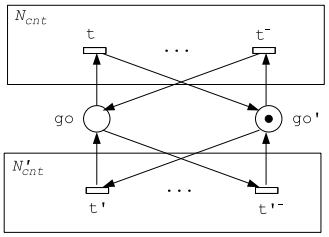
Observe that for every reachable marking m and every index i, we have $m(p_i) + m(q_i) = 1$. Therefore m can be coded by the binary word $\omega = \omega_{k-1} \dots \omega_0$ in which $\omega_i = m(q_i)$. The word ω is interpreted as the binary expansion of an integer between 0 and $2^k - 1$. We denote by $val(\omega)$ the integer value associated with w. Consider $w \notin \{0^k, 1^k\}$, there are two markings reachable from w which are $w+$ and $w-$ such that $val(w-) = val(w) - 1$ and $val(w+) = val(w) + 1$.

The figure below represents the reachability graph of the 3-bit counter. For a k-bit counter, the shortest firing sequence from 0^k to 1^k is σ_k defined inductively by: $\sigma_1 = t_0$ and $\sigma_{i+1} = \sigma_i t_i \sigma_i$.



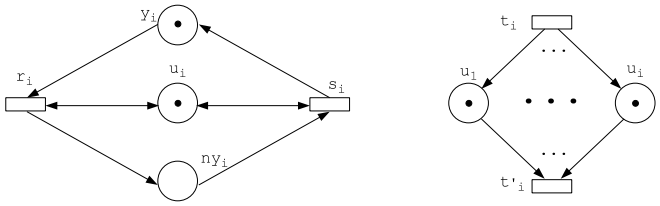
For every complex $c \equiv p_i + \sum_{j<i} q_j$ (resp. $c \equiv q_i + \sum_{j<i} p_j$), a possible witness is $wit(c) \equiv p_i + \sum_{j>i} 2^{j-i-1} p_j$ (resp. $wit(c) \equiv q_i + \sum_{j>i} 2^{j-i-1} q_j$). Thus this subnet has deficiency 0.

To manage transition firings between the update of counters, we duplicate the counter subnet and we synchronize the two subnets as indicated in the figure below. For a duplicated k-bit counter, the shortest firing sequence from the marking with the two counters set to 0^k and place go marked to the marking with the two counters set to 1^k and place go marked is obtained by: $\overline{\sigma}_1 = \overline{t}_0$ and $\overline{\sigma}_{n+1} = \overline{\sigma}_n \overline{t}_n \overline{\sigma}_n$ where $\overline{t}_i = t_i t'_i$.



This net has still deficiency 0 since the complexes are just enlarged by the places go or go' and their witnesses remain the same.

Variable modelling. For reasons that will become clear later on, the two counter subnets contain $n + 3$ bits indexed from 0 to $n + 2$. The bits $1, \dots, n$ of counter cnt correspond to the value of variables $x_1, \dots, x_n$. Managing the value of variables $y_1, \dots, y_n$ is done as follows. For every variable y_i, we add the subnet described below on the left (observe that $s_i = r_i^{-1}$) and modify the two counter subnets as described on the right.



When place y_i (resp. ny_i) is marked, this corresponds to interpreting variable y_i as **true** (resp. **false**). Changes of the interpretation are possible when place u_i is marked. This is the role of the modification done on the counter subnet: between a firing of t_i and t'_i places $\{u_j\}_{j \leq i}$ are marked. With this construction, we get the expected behaviour: the interpretation of a variable y_i can only be modified when the interpretation of a variable x_j with $j \geq i$ is modified. The complexes of the counter subnet are enlarged with places u_i and their witnesses remain the same since places in the support of these witnesses are not modified by transitions s_i and r_i. The new complex $y_i + u_i$ (resp. $ny_i + u_i$) has for witness y_i (resp. ny_i). Thus the new net has still deficiency 0.

Modelling the checking of the propositional formula. We now describe the subnet associated with the checking of propositional formula $\psi \equiv \bigwedge_{j \leq m} C_j$ where we assume w.l.o.g.: (1) that every clause $C_j \equiv l_{j,1} \vee l_{j,2} \vee l_{j,3}$ has exactly three literals (i.e. variables or negated variables); and (2) that every variable or negated variable occurs at least in one clause. The left upper part of Figure 5 shows the Petri net which describes clause C_j of the formula ψ. Places $\ell_{j,k} (k = 1, 2, 3)$ represent the literals while places $n\ell_{j,k}$ represent the literal *used as a proof of the clause*, the place $mutex_j$ avoids to choose several proofs of the clause (and thus ensuring safeness), and finally place $success_j$ can be marked if and only if the evaluation of the clause yields true for the current interpretation and one of its true literal is used as a proof.

The complexes of this subnet are $mutex_j + \ell_{j,k}$ (resp. $success_j + n\ell_{j,k}$) with witness $-n\ell_{j,k}$ (resp. $n\ell_{j,k}$). So the subnet has deficiency 0.

We now synchronise the clause subnets with the previous subnet in order to obtain the final net. Observe that in the previous subnet, transition t_0 (and t'_0) must occur after every interpretation change. This is in fact the role of bit 0 of the counter. Thus we constrain its firing by requiring the places $success_j$ to be marked as presented in the right upper part of Figure 5. Adding loops simply enlarges the complexes associated with t_0 and does not modify the incidence matrix. So the net has still deficiency 0.

It remains to synchronise the value of the variables and the values of the literals where the variables occur either positively or negatively. This is done in two steps. First $\ell_{j,k}$ is initially marked if the interpretation of the initial marking satisfies $\ell_{j,k}$. Then we synchronize the value changes as illustrated in the lower part of Figure 5. Once again the complexes are enlarged and the witnesses are still valid since the places $\ell_{j,k}$ do not belong to the support of any witness.

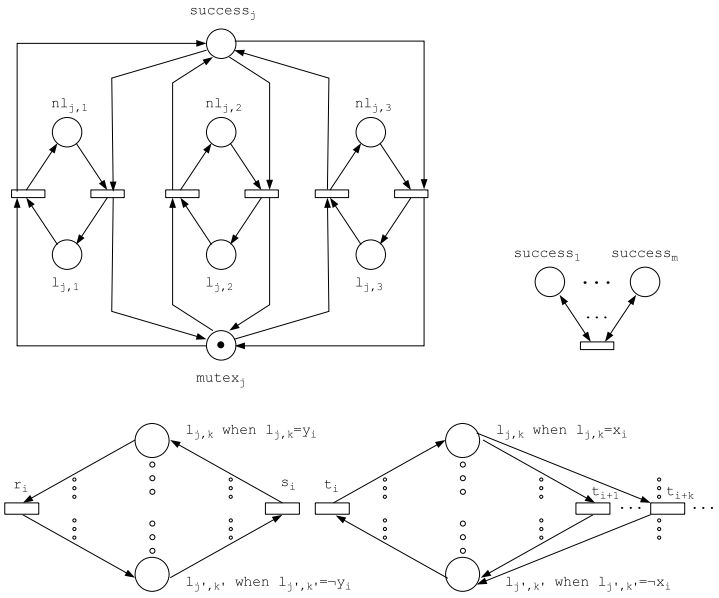


Fig. 5. Clause C_j (left), synchronisation with t_0 (right) and with variables (below)

Choice of the initial and final marking for the net. Let us develop a bit the sequence $\overline{\sigma}_{n+3}$ in the two counter subnet in order to explain the choice of initial marking for this subnet: $\overline{\sigma}_{n+3} = \overline{\sigma}_{n+1}t_{n+1}t'_{n+1}\overline{\sigma}_{n+1}t_{n+2}t'_{n+2}\overline{\sigma}_{n+1}t_{n+1}t'_{n+1}\overline{\sigma}_{n+1}$

We want to check all the interpretations of x_i's guessing the appropriate values of y_i's (if they exist). We have already seen that changing from one interpretation to another one (i.e. a counter incrementation or decrementation) allows to perform the allowed updates of y_i. However given the initial interpretation of the x_i's we need to make an initial guess of all the y_i's. So our initial marking restricted to the counter subnet will correspond to the marking reached after $\overline{\sigma}_{n+1}t_{n+1}$, i.e. corresponding to $cnt = 2^{n+1}$ (i.e. word $010\ldots0$), $cnt' = 2^{n+1} - 1$ (i.e. word $001\ldots1$) with in addition places go', u_i's, $mutex_j$'s and y_i's 1-marked; places $\ell_{j,k}$ are marked according to the initial marking of places x_i's and y_i's as explained before. All the other places are unmarked. This explains the role of bit $n+1$.

Furthermore, if we have successfully checked all the interpretations of the x_i's, the counters will have reached the value $2^{n+2} - 1$ (corresponding to a firing sequence obtained from $t'_{n+1}\overline{\sigma}_{n+1}$ with possible updates of y_i during change of interpretations). However we do not know what is the final guess for the y_i's. So firing transition t_{n+2} allows to set the y_i's in such a way that the final marking will correspond to $cnt = 2^{n+2}$ (i.e. word $10\ldots0$), $cnt' = 2^{n+2} - 1$ (i.e. word $01\ldots1$) with in addition places go', u_i's $mutex_j$'s and y_i's 1-marked; places $\ell_{j,k}$

are marked accordingly. All the other places are unmarked. This explains the role of bit $n+2$.

By construction, the net reaches the final marking iff the formula is satisfied. Observe that the checking of clauses can be partially done concurrently with the change of interpretation. However as long as, in the net, a clause C_j is "certified" by a literal $\ell_{j,k}$ (i.e. marking place $success_j$ and unmarking place $\ell_{j,k}$) the value of the variable associated with the literal cannot change, ensuring that when t_0 is fired, the marking of any place $success_j$ corresponds to the evaluation of clause C_j with the current interpretation. □

Corollary 4.1. *The liveness problem for safe* Π^2*-nets is* PSPACE*-complete.*

Proof. Observe that the transitions of the net of the previous proof are fireable at least once (and so live by weak reversibility) iff φ is true. □

Let us now consider general (non-safe) Petri nets. Reachability and coverability of symmetric nets is EXPSPACE-complete [16]. In [9], it is proved that both problems are EXPSPACE-complete for WR nets (which include symmetric Petri nets). Next proposition establishes the same result for the coverability of Π^2-nets.

Proposition 4.1. *The coverability problem for* Π^2*-nets is* EXPSPACE*-complete.*

The complexity of reachability for Π^2-nets remains an open issue (indeed the proof of EXPSPACE-hardness does not work for reachability).

5 The subclass of Π^3-nets

In this section, we introduce Π^3-nets, a subclass of product-form Petri nets for which the normalising constant can be efficiently computed. The first subsection defines the subclass; the second one studies its structural properties and the third one is devoted to the computation of the normalising constant.

5.1 Definition and properties

Definition 5.1 (Ordered Π-net) *Consider an integer $n \geq 2$. An n-level* ordered Π-net *is a Π-net $\mathcal{N} = (P, T, W^-, W^+)$ such that:*

1. $P = \bigsqcup_{1 \leq i \leq n} P_i$, $T = \bigsqcup_{1 \leq i \leq n} T_i$ *and* $P_i \neq \emptyset$ *for all* $1 \leq i \leq n$,
2. $\mathcal{M}_i = (P_i, T_i, W^-_{|P_i \times T_i}, W^+_{|P_i \times T_i})$ *is a strongly connected state machine,*
3. $\forall 1 \leq i \leq n\,, \forall t \in T_i\,, \forall p \in P$, ${}^\bullet t(p) > 0$ *implies* $p \in P_i$ *or* $p \in P_{i-1}$ *($P_0 = \emptyset$),*
4. $\forall 2 \leq i \leq n\,, \exists t \in T_i\,, \exists p \in P_{i-1}$ *s.t.* ${}^\bullet t(p) > 0$,
5. $\forall 1 \leq i \leq n\,, \forall t, t' \in T_i$, $({}^\bullet t \cap {}^\bullet t') \cap P_i \neq \emptyset$ *implies* ${}^\bullet t = {}^\bullet t'$.

We call $\mathcal{M}_i$ the level i state machine. The elements of P_i (resp. T_i) are level i places (resp. transitions). The complexes ${}^\bullet t$ with $t \in T_i$ are level i complexes.

By weak reversibility, the constraints 3, 4, and 5 also apply to the output bags $t^\bullet$. An *ordered* Π*-net* is a sequence of strongly connected state machines. Connections can only be made between a level i transition and a level $(i-1)$ place (points 1, 2, 3). By construction, an *ordered* Π*-net* is connected (point 4). Each level i place belongs to one and only one level i complex (point 5).

Lemma 5.2 *The reaction net of $\mathcal{N}$ is isomorphic to the disjoint union of state machines $\mathcal{M}_i$. So a T-semi-flow of $\mathcal{M}_i$ is also a T-semi-flow of $\mathcal{N}$. If a transition of T_i is enabled by a reachable marking then every transition of T_i is live.*

An ordered Π-net may be interpreted as a multi-level system. The transitions represent jobs or events while the tokens in the places represent resources or constraints. A level i job requires resources from level $(i-1)$ and relocates these resources upon completion. Conversely, events occurring in level $(i-1)$ may make some resources unavailable, hence interrupting activities in level i. The dependency of an activity on the next level is measured by *potentials*, defined as follows.

Definition 5.3 (Interface, potential) *A place $p \in P_i$, $1 \leq i \leq n-1$, is an* interface place *if $p \in t^\bullet$ for some $t \in T_{i+1}$. For a place $p \in P_i$, $2 \leq i \leq n$, and a place $q \in P_{i-1}$, set:*

$$pot(p,q) = \begin{cases} t^\bullet(q) & \text{if } p \text{ and } q \text{ have a common input transition } t \in T_i \\ 0 & \text{otherwise.} \end{cases}$$

The potential *of a place $p \in P_i$, $2 \leq i$, is defined by: $pot(p) = \sum_{q \in P_{i-1}} pot(p,q)$. By convention, $pot(p) = 0$ for all $p \in P_1$.*

By the definition of ordered Π-nets, the quantity $t^\bullet(q)$ does not depend on the choice of t, so the potential is well-defined.

Example. The Petri net in Figure 6 is a 3-level ordered Π-net. The potentials are written in parentheses. To keep the figure readable, the arcs between the place p_1 and the level 2 transitions are omitted.

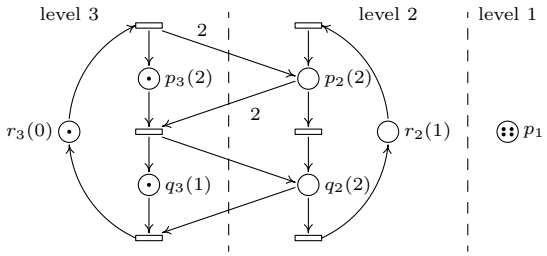


Fig. 6. Ordered Π-net.

The behaviour of the state machines $\mathcal{M}_i$ is embedded in the behaviour of $\mathcal{N}$, in the sense that the marking stripped off of the potentials evolves like a marking of the state machines.

Definition 5.4 (Effective marking) *The* effective marking *of a marking m, denoted by $\widetilde{m}$, is defined as follows. For all $i \leq n$ and $p \in P_i$,*

$$\widetilde{m}(p) = m(p) + \sum_{j=1}^{n-i}\big((-1)^j \sum_{\substack{r_1 \in P_{i+1} \\ \dots \\ r_j \in P_{i+j}}} m(r_j)\big(\prod_{k=1}^{j-1} pot(r_{k+1}, r_k)\big)pot(r_1, p)\big)\,. \tag{5.1}$$

Remark. Note that an effective marking is not necessarily non-negative. It can be showed by induction that:
$\forall p \in P_n\,, \widetilde{m}(p) = m(p)$ and $\forall p \in P_i\,, i < n\,, \widetilde{m}(p) = m(p) - \sum_{r \in P_{i+1}} \widetilde{m}(r)pot(r,p)$

Lemma 5.5 *Let m, m' be two vectors such that $m' = m + W(t)$ for some $t \in T_i$ $(1 \leq i \leq n)$. Let p_1 and p_2 denote the input place and the output place of t in P_i, respectively. Then for every place p:*

$$\widetilde{m}'(p) = \widetilde{m}(p) - 1 \text{ if } p \text{ is } p_1, \quad \widetilde{m}(p) + 1 \text{ if } p \text{ is } p_2, \quad \widetilde{m}(p) \text{ otherwise.} \tag{5.2}$$

The above lemma applies in particular when m and m' are markings such that $m \xrightarrow{t} m'$. Eqns (5.2) look like the equations for witnesses. Since each level i complex contains exactly one level i place, one guesses that every complex admits a witness, i.e. that $\mathcal{N}$ is a Π^2-net. This is confirmed by the next proposition.

Proposition 5.6 *Let B denote the $P \times P$ integer matrix of the linear transformation $m \mapsto \widetilde{m}$ defined by (5.1). For $p \in P_i$, the line vector $B(p)$ is a witness for the i-level complex containing p. In particular, $\mathcal{N}$ is a Π^2-net.*

Lemma 5.5 allows to derive the S-invariants of $\mathcal{N}$ induced by S-semi-flows.

Corollary 5.7 *Let m_0 be the initial marking of $\mathcal{N}$. We have:*
$$\forall m \in \mathcal{R}(m_0), \quad \forall i \in \{1, \dots, n\}, \qquad \widetilde{m}(P_i) = \widetilde{m}_0(P_i)$$
More generally, for all i, the vector $v_i = \sum_{p \in P_i} B(p)$ is a S-semi-flow of $\mathcal{N}$.

Example. Consider the ordered Π-net in Figure 6 with the initial marking $m_0 = p_3 + q_3 + r_3 + 4p_1$. The effective marking of m_0 is $\widetilde{m_0} = p_3 + q_3 + r_3 - 2p_2 - q_2 + 10p_1$. Any reachable marking m satisfies the invariants:
$$m(P_3) = 3$$
$$m(P_2) - 2m(p_3) - m(q_3) = -3$$
$$m(p_1) - 2m(p_2) - 2m(q_2) - m(r_2) + 4m(p_3) + 2m(q_3) = 10$$
It can be shown that $\{v_i, 1 \leq i \leq n\}$ is a basis of the S-semi-flows of $\mathcal{N}$.

Proposition 5.8 *Let v be an S-semi-flow of $\mathcal{N}$, i.e. $v.W = 0$. There exist rational numbers $a_1, \dots, a_n$ such that $v = \sum_{i=1}^{n} a_i v_i$.*

The independence of the set $\{v_i\,, 1 \leq i \leq n\}$ follows from the fact that the vectors $v_i B^{-1}$ have non-empty disjoint supports.

We now consider only ordered Π-nets in which the interface places in P_i have maximal potential among the places of P_i. From the technical point of view, this assumption is crucial for the reachability set ananlysis presented later. From the modelling point of view, it is a reasonable restriction. Consider the multi-level model, the assumption means that during the executions of level i jobs, the level $(i-1)$ is idle, therefore the amount of available resource is maximal.

Definition 5.9 (Π^3-net) *An ordered Π-net $\mathcal{N}$ is a Π^3-net if:*

$$\forall i, \forall p \in P_i \,:\, p \in {}^\bullet T_{i+1} \implies pot(p) = max\{pot(q), q \in P_i\}\,.$$

5.2 The reachability set

From now on, $\mathcal{N}$ is a n-level Π^3-net with $\mathcal{M}_1, \ldots, \mathcal{M}_n$ its state machines.

Definition 5.10 (Minimal marked potential) *Consider $i \in \{2, \ldots, n\}$. The* level i minimal potential marked by m *is:*

$$\varphi_i(m) = \begin{cases} max\{pot(p), p \in P_i\} & \text{if } m(P_i) = 0\,, \\ min\{pot(p), p \in P_i, m(p) > 0\} & \text{if } m(P_i) > 0\,. \end{cases}$$

Next lemma gives a necessary condition for reachability.

Lemma 5.11 *If $\varphi_i(m) \leq m(P_{i-1})$ then $\varphi_i(m') \leq m'(P_{i-1})$ for all $m' \in \mathcal{R}(m)$.*

For our purposes, we now define the partial liveness and partial reachability.

Definition 5.12 (i-reachability set, i-liveness) *Let m be a marking. The i-reachability set of m, denoted by $\mathcal{R}_i(m)$, is the set of all markings reachable from m by a firing sequence consisting of transitions in $\bigcup_{1 \leq j \leq i} T_j$. We say that m is* i-live *if for any transitions t in $\bigcup_{1 \leq j \leq i} T_j$, there exists a marking in $\mathcal{R}_i(m)$ which enables t. By convention, $\mathcal{R}_0(m) = \{m\}$ and every marking is 0-live.*

The i-live markings are characterised by the following proposition.

Proposition 5.13 *A marking m is i-live if and only if it satisfies the following inequalities, called the* i-condition*:*

$$m(P_i) > 0 \wedge \forall 2 \leq j \leq i \,:\, m(P_{j-1}) \geq \varphi_j(m) \tag{5.3}$$

If m satisfies the i-condition then for every $p, q \in P_i$ such that $p \neq q$, $m(p) > 0$ and $pot(p) \leq m(P_{i-1})$, there exists $m' \in \mathcal{R}_i(m)$ such that:

$$m'(p) = m(p) - 1\,,\ m'(q) = m(q) + 1\,,\ \forall r \in P_i \setminus \{p, q\},\ m'(r) = m(r)\,. \tag{5.4}$$

A marking is live if and only if it satisfies the n-condition.

Example: The ordered Π-net in Figure 6 is a Π^3-net. Consider two markings: $m_1 = p_3 + q_3 + r_3 + 4p_1$ and $m_2 = 3q_3 + 4p_1$. These markings agree on all the S-invariants, but only m_1 satisfies the 3-condition. It is easy to check that m_1 is live while m_2 is dead.

We conclude this subsection by showing that the reachability problem for Π^3-nets can be efficiently decided as well.

Theorem 5.14 *Suppose that the initial marking m_0 is live. Then the reachability set $\mathcal{R}(m_0)$ coincides with the set $\mathcal{S}(m_0)$ of markings which satisfy the n-condition and agree with m_0 on the S-invariants given by Corollary 5.7.*

5.3 Computing the normalising constant

The normalising constant of a product-form Petri net (see Section 2.1) is $G = \sum_m \mathbb{1}_{m\in\mathcal{R}(m_0)} \prod_{p\in P} u_p^{m(p)}$. It is in general a difficult task to compute G, as can be guessed from the complexity of the reachability problem. However, efficient algorithms may exist for nets with a well-structured reachability set. Such algorithms were known for Jackson networks [18] and the *S-invariant reachable* Petri nets defined in [6]. We show that is is also the case for the class of live Π^3-nets which is strictly larger than the class of Jackson networks (which correspond to 1-level ordered nets) and is not included in the class of S-invariant reachable Petri nets.

Suppose that m_0 is a live marking. Suppose that the places of each level are ordered by increasing potential: $P_i = \{p_{i1}, \ldots, p_{ik_i}\}$ such that $\forall 1 \leq j < k_i$, $pot(p_{ij}) \leq pot(p_{i(j+1)})$.

Let V denote the $n \times P$-matrix the i-th row of which is the S-invariant v_i defined in Corollary 5.7. For $1 \leq i \leq n$, set $C_i = v_i m_0 = \widetilde{m}_0(P_i)$. Then the reachability set consists of all n-live markings m such that $Vm = {}^t(C_1, \ldots, C_n)$.

For $1 \leq i \leq n$, $1 \leq j \leq k_i$ and $c_1, \ldots, c_i \in \mathbb{Z}$, define $E(i, j, c_1, \ldots, c_i)$ as the set of markings m such that

$$\begin{cases} m(p_{i\nu}) = 0 \text{ for all } \nu > j \\ Vm = {}^t(c_1, \ldots, c_i, 0 \ldots, 0) \\ \varphi_\nu(m) \leq m(P_{\nu-1}) \text{ for all } 2 \leq \nu \leq i\,. \end{cases}$$

The elements of $E(i, j, c_1, \ldots, c_i)$ are the markings which satisfy the second part of the i-condition and the S-invariants constraints $(c_1, \ldots, c_i, 0, \ldots, 0)$ and concentrate tokens in $P_1, \ldots, P_{i-1}$ and $\{p_{i1}, \ldots, p_{ij}\}$.

With each $E(i, j, c_1, \ldots, c_i)$ associate

$$G(i, j, c_1, \ldots, c_i) = \pi(E(i, j, c_1, \ldots, c_i)) = \sum \prod_{p\in P} u_p^{m(p)}$$

the sum being taken over all $m \in E(i, j, c_1, \ldots, c_i)$.

We propose to compute $G(n, k_n, C_1, \ldots, C_n)$ by dynamic programming. It consists in breaking each $G(i, j, c_1, \ldots, c_i)$ into smaller sums. This corresponds to a partition of the elements of $E(i, j, c_1, \ldots, c_i)$ by the number of tokens in p_{ij}.

Proposition 5.15 *Let be given $E = E(i, j, c_1, \ldots, c_i)$. If $c_i < 0$ then $E = \emptyset$. If $c_i \geq 0$ then for every non-negative integer a:*

1. *If $a > c_i$ then $E \cap \{m | m(p_{ij}) = a\} = \emptyset$.*
2. *If $a < c_i$ and $j = 1$ then $E \cap \{m | m(p_{ij}) = a\} = \emptyset$.*
3. *If $a < c_i$ and $j \geq 2$ then $E \cap \{m | m(p_{ij}) = a\} = E(i, j-1, c_1 - v_1(ap_{ij}), \ldots, c_i - v_i(ap_{ij}))$:*
4. *If $a = c_i$ and $i = 1$ then $E \cap \{m | m(p_{ij}) = a\} = \{c_1 p_{1j}\}$.*
5. *If $a = c_i$ and $i > 1$ then $E \cap \{m | m(p_{ij}) = a\} = E(i-1, k_{i-1}, c_1 - v_1(ap_{ij}), \ldots, c_{i-1} - v_{i-1}(ap_{ij}))$:*

Proposition 5.15 induces the following relations between the sums $G(i, j, c_1, \ldots, c_i)$.

Corollary 5.16 *If $c_i < 0$ then $G(i, j, c_1, \ldots, c_i) = 0$. If $c_i \geq 0$ then:*

- *Case $2 \leq i \leq n$, $2 \leq j \leq k_i$:*

$$G(i, j, c_1, \ldots, c_i) = \sum_{\nu=0}^{c_i - 1} u_{p_{ij}}^{\nu} G(i, j-1, c_1 - v_1(\nu p_{ij}), \ldots, c_i - v_i(\nu p_{ij})) + u_{p_{ij}}^{c_i} G(i-1, k_{i-1}, c_1 - v_1(c_i p_{ij}), \ldots, c_{i-1} - v_{i-1}(c_i p_{ij}))\,.$$

- *Case $2 \leq i \leq n$, $j = 1$:*

$$G(i, 1, c_1, \ldots, c_i) = u_{p_{i1}}^{c_i} G(i-1, k_{i-1}, c_1 - v_1(c_i p_{i1}), \ldots, c_{i-1} - v_{i-1}(c_i p_{i1}))\,.$$

- *Case $i = 1$, $j \geq 2$: $G(1, j, c_1) = \sum_{\nu=0}^{c_1 - 1} u_{p_{1j}}^{\nu} G(1, j-1, c_1 - \nu) + u_{p_{1j}}^{c_1}$.*
- *Case $i = 1$, $j = 1$: $G(1, 1, c_1) = u_{p_{11}}^{c_1}$.*

Complexity. Since $i \leq n$, $j \leq K = \max\{k_1, \ldots, k_n\}$, the number of evaluations is bounded by $n \times K \times \gamma$, where γ upper bounds the c_i's. Let α denote the global maximal potential. From (5.1), we obtain $\gamma = \mathcal{O}(m_0(P) K^n \alpha^n)$. So the complexity of a dynamic programming algorithm using Cor. 5.16 is $\mathcal{O}(m_0(P) n K^{n+1} \alpha^n)$, i.e. pseudo-polynomial for a fixed number of state machines.

6 Perspectives

This work has several perspectives. First, we are interested in extending and applying our rules for a modular modelling of complex product-form Petri nets. Then we want to validate the formalism of Π^3-nets showing that it allows to express standard patterns of distributed systems. Finally we conjecture that reachability is EXPSPACE-complete for Π^2-nets and we want to establish it.

Acknowledgements. We would like to thank the anonymous referess whose numerous and pertinent suggestions have been helpful in preparing the final version of the paper.

References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1995.

[2] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. John Wiley & Sons, New York, 1992.

[3] F. Baskett, K. M. Chandy, R. R. Muntz, F. Palacios. Open, closed and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.

[4] R. J. Boucherie, M. Sereno. On closed support T-invariants and traffic equations. *Journal of Applied Probability*, (35): 473–481, 1998.

[5] J. Esparza and M. Nielsen. Decidability issues for Petri nets - a survey. *Journal of Informatik Processing and Cybernetics*, 30(3):143-160, 1994.

[6] J.L. Coleman, W. Henderson, P.G. Taylor. Product form equilibrium distributions and a convolution algorithm for stochastic Petri nets. *Performance Evaluation*, 26(3):159–180, September 1996.

[7] J. Esparza. Reduction and Synthesis of Live and Bounded Free Choice Petri Nets. *Information and Computation*, 114(1):50–87, 1994

[8] M. Feinberg. Lectures on chemical reaction networks. *Given at the Math. Research Center, Univ. Wisconsin, 1979*. Available online at `http://www.che.eng.ohio-state.edu/~feinberg/LecturesOnReactionNetworks`.

[9] S. Haddad, P. Moreaux, M. Sereno, M. Silva Product-form and stochastic Petri nets: a structural approach. *Performance Evaluation*, 59: 313-336, 2005.

[10] W. Henderson, D. Lucic, P.G. Taylor. A net level performance analysis of stochastic Petri nets. *Journal of Australian Mathematical Soc. Ser. B*, 31:176–187, 1989.

[11] J. R. Jackson. Jobshop-like Queueing Systems. *Management Science*, 10(1): 131–142, 1963.

[12] F. Kelly. *Reversibility and Stochastic Networks*. Wiley, New-York, 1979.

[13] A. A. Lazar, T. G.Robertazzi. Markovian Petri Net Protocols with Product Form Solution. *Proc. of INFOCOM 87*, pp. 1054–1062, San Francisco, CA, USA, 1987.

[14] M. Li, N. D. Georganas. Parametric Analysis of Stochastic Petri Nets, *Fifth International Conference on Modelling and Tools for Computer Performance Evaluation*, Torino, Italy, 1991,

[15] J. Mairesse, H-T. Nguyen. Deficiency Zero Petri Nets and Product Form. *Petri Nets 2009*, LNCS 5606, 103-122, 2009.

[16] E. Mayr, A. Meyer. The complexity of the word problem for commutative semigroups an polynomial ideals. *Advances in Math*, 46 (1982), 305-329.

[17] C. Papadimitriou. Computational Complexity. *Addison Wesley*, 1994.

[18] M. Reiser, S.S. Lavenberg. Mean Value Analysis of Closed Multichain Queueing Networks. *Journal of the ACM*, 27(2): 313-322, 1980.