Petri Nets with Marking-Dependent Arc Cardinality: Properties and Analysis

Gianfranco Ciardo

Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795, USA
ciardo@cs.wm.edu

Abstract. We discuss P/T-nets where the arc cardinalities are allowed to be marking-dependent expressions of various types, resulting in a hierarchy of subclasses. Some of the language and decidability properties of these classes have been studied before, but we focus on the practical implications in systems modeling, adding some new insight to the known results about the relative expressive power of the subclasses.
We show how the p-semiflows of a P/T-net with marking-dependent arc cardinality can be obtained from the p-semiflows of a related ordinary P/T-net and how bounds on the relative throughputs of the transitions can be obtained, a weaker condition than t-semiflows.
Finally, we briefly discuss several modeling applications where these subclasses are used.

1 Introduction

Petri nets were introduced by Petri [22] to model concurrent behavior and, since then, they have been studied for their theoretical properties and adopted as an effective description formalism to model discrete-state systems.

Informally, a Petri net is a finite bipartite directed graph where the nodes are either places or transitions. Tokens reside in places, and move according to the firing rule. Three classes of Petri nets have been considered [23]: condition-event nets (C/E-nets), where at most one token can reside in each place, resulting in a finite state-space; place-transition nets (P/T-nets), which we consider in this paper, where this restriction is lifted, resulting in a possibly infinite state-space; and predicate-event nets (P/E-nets) and high-level nets, where tokens have individual identities.

Two main directions of Petri net research are well represented in two cycles of conferences: the International (formerly European) Conference on Applications and Theory of Petri Nets, held annually, with a focus on the theoretical computer science aspects of nets, and the International Workshop on Petri Nets and Performance Models, with a focus on the use of nets augmented with stochastic or deterministic timing for the quantitative analysis of systems (stochastic Petri nets, or SPNs). While these two areas span a wide range of interests in computer science, from formal languages, decidability, and complexity theory, to stochastic processes, numerical analysis, and optimization, there has been a good amount of cross-fertilization, as attested by the number of researchers active in both conferences.

The standard definition of Petri net assumes that the cardinality of the input and output arcs is constant but, at times, the reality being modeled behaves differently: a given event might require to remove from a place, or add to a place, a number of tokens which varies according to the marking of the net. Assuming that the behavior can be modeled at all, extraneous places and transitions must be used, whose only purpose is to model explicitly the movement of a variable number of tokens.

From a practical point of view this is both unpleasant, since it adds additional clutter to the model without adding any useful detail, and inefficient, since it requires to process sequentially what is essentially a bulk movement of tokens. For this reason, SPNs with marking-dependent arc cardinality were introduced in [8], adopted in several types of SPN definitions [9, 10, 11], and implemented in a computer-based analysis tool, SPNP [13]. More recently other SPN modeling tools have adopted them as well [18, 19].

Much earlier, though, researchers interested in the expressive power of (untimed) P/T-nets, introduced similar extensions: reset nets, where the firing of a transition can empty a place [3], and self-modifying nets (SM-nets) and their subclasses [24, 25], where the cardinality of input and output arcs can be any nonhomogeneous linear combination of the number of tokens in each place.

In this paper, we consider P/T-nets with various restrictions on the type of arc cardinality (Sect. 2), obtaining a hierarchy of nets. For these, we consider both the computation of minimal p-semiflows (Sect. 3), and the analysis of their expressive power (Sect. 4). Finally, we list some examples from the modeling literature where P/T-nets with marking-dependent arc cardinality have been used, or could have been used, advantageously (Sect. 5).

2 P/T-Nets with Marking-Dependent Arc Cardinality

A P/T-net with marking-dependent arc cardinality is a tuple

$$N = (P, T, D^-, D^+, \mu^{[0]})$$

where:

- $P = \{p_1, p_2, \ldots p_{|P|}\}$ is a finite set of places, which can contain tokens. A marking $\mu = (\mu_1, \mu_2, \ldots \mu_{|P|}) \in \mathbb{N}^{|P|}$ describes an assignment of tokens to each place[1]. A marking-dependent expression is a function of the marking, $f(\mu) = f(\mu_1, \mu_2, \ldots, \mu_{|P|})$.
- $T = \{t_1, t_2, \ldots t_{|T|}\}$ is a finite set of transitions, $P \cap T = \emptyset$.
- $\forall i, 1 \le i \le |P|, \forall j, 1 \le j \le |T|$, $D^-_{i,j} : \mathbb{N}^{|P|} \to \mathbb{N}$ and $D^+_{i,j} : \mathbb{N}^{|P|} \to \mathbb{N}$ are the marking-dependent cardinalities of the input arc from p_i to t_j and of the output arc from t_j to p_i, respectively.
- $\mu^{[0]} \in \mathbb{N}^{|P|}$ is the initial marking.

[1] We use the symbols $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ to indicate the non-negative integers, the integers, and the rational numbers, respectively

Places and transitions are drawn as circles and rectangles, respectively. The number of tokens in a place is written inside the place itself (default is zero). Input and output arcs have an arrowhead on their destination. The cardinality is written on the arc (default is the constant function one). A missing arc indicates that the cardinality is the constant function zero.

A transition $t_j \in T$ is enabled in marking μ iff $D^-_{\bullet,j}(\mu) \leq \mu$ ($A_{\bullet,j}$ indicates the j-th column of A). The set of transitions enabled in marking μ is denoted by $\mathcal{E}(\mu) = \{t_j \in T : D^-_{\bullet,j}(\mu) \leq \mu\}$. A transition $t_j \in \mathcal{E}(\mu)$ can fire, we denote the new marking $\mathcal{M}(t,\mu) = \mu - D^-_{\bullet,j}(\mu) + D^+_{\bullet,j}(\mu) = \mu + D_{\bullet,j}(\mu)$, where $D = D^+ - D^-$ is the incidence matrix. The reachability set is $\mathcal{R}(N) = \{\mu \in \mathbb{N}^{|P|} : \exists \sigma \in T^* \wedge \mu = \mathcal{M}(\sigma, \mu^{[0]})\}$, where the first argument of function $\mathcal{M}$ is extended to sequences of transitions.

2.1 Classes of P/T-Nets with Marking-Dependent Arc Cardinality

The class of P/T-nets just defined is Turing-equivalent, since an input arc from p_i to t_j with cardinality $2\mu_i$ is exactly equivalent to an inhibitor arc from p_i to t_j (see Fig. 1), and P/T-nets with inhibitor arcs can model Turing machines [15]: if $\mu_i = 0$, the arc has no effect on the enabling of t_j and no token is removed from p_i if t_j fires; if $\mu_i > 0$, the cardinality of the arc is greater than the number of tokens in p_i, hence t_j is disabled.

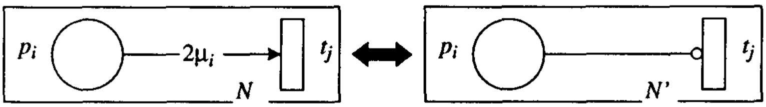

Fig. 1. Equivalence of inhibitor arcs and marking-dependent cardinality arcs.

Certain marking-dependent behaviors, though, such as removing all tokens from a place or moving all tokens from a place to another are often useful to model many systems of practical interest, hence we define the following subclasses of P/T-nets with restricted marking-dependent cardinality arcs (assume $\alpha^-, \alpha^+ \in \mathbb{N}^{|P|\times|T|}$ and $\beta^-, \beta^+ \in \mathbb{N}^{|P|\times|T|\times|P|}$):

$\mathcal{N}_o$: ordinary P/T-nets, with constant arc cardinalities,

$$D^-_{i,j} = \alpha^-_{i,j} \quad \wedge \quad D^+_{i,j} = \alpha^+_{i,j} \ .$$

$\mathcal{N}_r$: reset P/T-nets [3], with the addition of reset, or flushing arcs,

$$D^-_{i,j} = \alpha^-_{i,j} + \beta^-_{i,j,i}\mu_i, \ \beta^-_{i,j,i} \leq 1, \beta^-_{i,j,i} = 1 \Rightarrow \alpha^-_{i,j} = 0 \quad \wedge \quad D^+_{i,j} = \alpha^+_{i,j} \ .$$

$\mathcal{N}_p$: post self modifying P/T-nets [24], also called set nets in [25], where the cardinality of output arcs can be any nonhomogeneous linear combination of the marking (hence they allow duplication of the number of tokens in a place),

$$D^-_{i,j} = \alpha^-_{i,j} \quad \wedge \quad D^+_{i,j} = \alpha^+_{i,j} + \sum_{1 \le l \le |P|} \beta^+_{i,j,l}\mu_l \ .$$

$\mathcal{N}_t$: transfer P/T-nets, a new class, where the firing of a transition can move all the tokens from one place to another (but not duplicate them),

$$D^-_{i,j} = \alpha^-_{i,j} + \beta^-_{i,j,i}\mu_i, \ \beta^-_{i,j,i} \le 1, \beta^-_{i,j,i} = 1 \Rightarrow \alpha^-_{i,j} = 0$$

$$\wedge \ D^+_{i,j} = \alpha^+_{i,j} + \sum_{1 \le l \le |P|} \beta^+_{i,j,l}\mu_l \ ,$$

subject to $\beta^-_{i,j,i} = \sum_{1 \le l \le |P|} \beta^+_{l,j,i}$.

$\mathcal{N}_l$: linear transfer P/T-nets, also called reset-set nets in [25], which allow both reset and duplication, hence transfer, behavior,

$$D^-_{i,j} = \alpha^-_{i,j} + \beta^-_{i,j,i}\mu_i, \ \beta^-_{i,j,i} \le 1, \beta^-_{i,j,i} = 1 \Rightarrow \alpha^-_{i,j} = 0$$

$$\wedge \ D^+_{i,j} = \alpha^+_{i,j} + \sum_{1 \le l \le |P|} \beta^+_{i,j,l}\mu_l \ .$$

$\mathcal{N}_s$: self modifying P/T-nets [24], where the cardinality of both input and output arcs can be any nonhomogeneous linear combination of the marking,

$$D^-_{i,j} = \alpha^-_{i,j} + \sum_{1 \le l \le |P|} \beta^-_{i,j,l}\mu_l \quad \wedge \quad D^+_{i,j} = \alpha^+_{i,j} + \sum_{1 \le l \le |P|} \beta^+_{i,j,l}\mu_l \ .$$

The possible patterns of marking-dependent arc cardinalities are summarized in Fig. 2.

2.2 Observations

Often, a transition t_1 having p_1 as its only input place, with cardinality μ_1, should not be considered enabled when $\mu_1 = 0$. In other words, the flushing of p_1 should occur only when μ_1 reaches a minimum threshold k, often one (see N in Fig. 3). This behavior is represented in $N' \in \mathcal{N}_r$ (hence $\mathcal{N}_t$, $\mathcal{N}_l$, and $\mathcal{N}_s$) by adding places p_{run} and p_{stop} and transition $t_{(1,p_1)}$, with p_{run} being input and output with cardinality one to any other transition in the net (this is a standard technique which ensures that $t_{(1,p_1)}$ is the only enabled transition when the token is removed from p_{run}). Hence, given $\delta^- \in \mathbb{N}^{|P| \times |T|}$, we can allow $D^-_{i,j}$ to be of the form $\max\{\delta^-_{i,j}, \alpha^-_{i,j} + \beta^-_{i,j,i}\mu_i\}$ for nets is $\mathcal{N}_r$, $\mathcal{N}_t$, and $\mathcal{N}_l$, or $\max\{\delta^-_{i,j}, \alpha^-_{i,j} + \sum_{1 \le l \le |P|} \beta^-_{i,j,l}\mu_l\}$ for nets in $\mathcal{N}_s$, with the understanding that this is just a shorthand, not a real extension.

In the original definition [3], reset arcs were drawn from the transition to the place to be reset, with a circle drawn on them. Hence, the threshold behavior

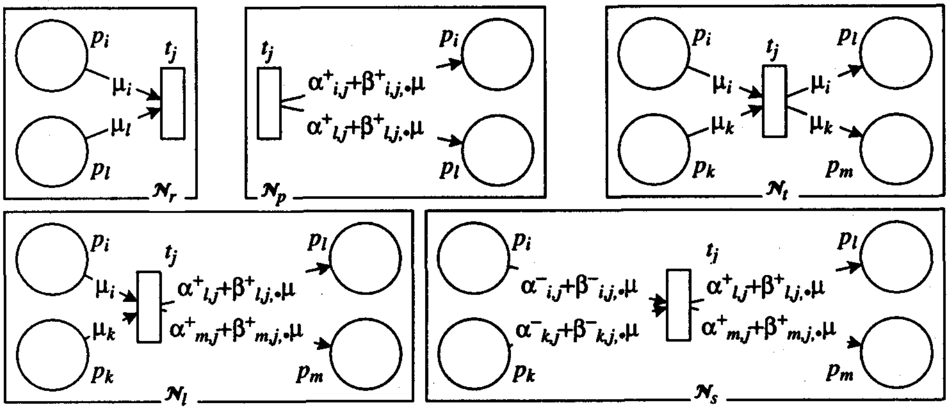

Fig. 2. Patterns of marking-dependent cardinality arcs for $\mathcal{N}_r$, $\mathcal{N}_p$, $\mathcal{N}_t$, $\mathcal{N}_l$, and $\mathcal{N}_s$.

could be represented naturally with a normal input arc (see net N'' in Fig. 3). An additional requirement could be that m tokens be deposited in p_1 after this flushing, so that the entire semantic of N is that t_1 fires only if $\mu_1 \geq k$, and, after the firing, $\mu_1 = m$. This can be accomplished with an arc from t_1, for N and N'', or from $t_{(1,p_1)}$, for N', back to p_1, with cardinality m.

It is also easy to show that a reset behavior can be modeled by $\mathcal{N}_t$, $\mathcal{N}_l$, and $\mathcal{N}_s$, in addition to $\mathcal{N}_r$. In the case of $\mathcal{N}_t$, it is necessary to add a dummy place where the tokens removed from the place to be reset can be deposited (or we could explicitly allow a reset behavior in addition to the transfer behavior in $\mathcal{N}_t$, by requiring $\beta^-_{i,j,i} \geq \sum_{1\leq l\leq|P|} \beta^+_{l,j,i}$, instead of a strict equality).

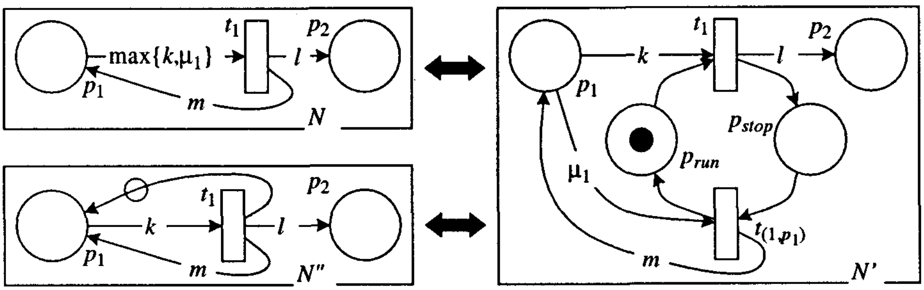

Fig. 3. Flushing p_1 only after a threshold k.

3 Semiflow Computation

Invariants are an excellent aid in understanding the behavior of an ordinary P/T-net. In this section, we generalize them to P/T-nets with marking-dependent arc cardinality.

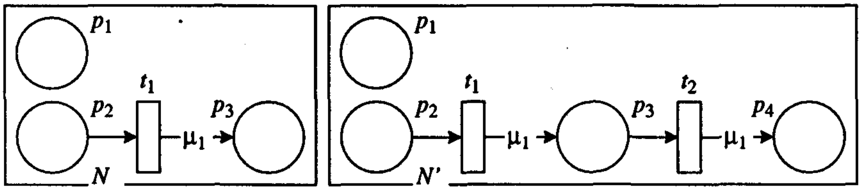

Fig. 4. P/T-nets with non-linear invariants.

3.1 P-Semiflows

In ordinary P/T-nets, a structural linear p-invariant could state that $\forall\mu \in \mathcal{R}(N), x \cdot \mu = x \cdot \mu^{[0]}$, where $x \in \mathbb{Q}^{|P|}$ is a p-flow, and satisfies $xD = 0$, and the operator "$\cdot$" indicates inner (scalar) product between two vectors. Specifically, we are interested in the minimal non-zero p-flows $x \in \mathbb{N}^{|P|}$, or minimal p-semiflows [14]. If D contains arbitrary marking-dependent expressions, non-linear structural p-invariants might exist as well. For example, N in Fig. 4 satisfies the non-linear invariant $\forall\mu \in \mathcal{R}(N), \mu_1\mu_2 + \mu_3 = \mu_1^{[0]}\mu_2^{[0]} + \mu_3^{[0]}$.

Following [25], this behavior can be described by a "bilinear invariant" $X \in \mathbb{Z}^{(1+|P|)\times(1+|P|)}$:

$$\forall\mu \in \mathcal{R}(N), [1,\mu]X \cdot [1,\mu] = [1,\mu^{[0]}]X \cdot [1,\mu^{[0]}] \quad \text{where} \quad X = \begin{bmatrix} 0&0&0&1 \\ 0&0&1&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix} .$$

Unfortunately, though, bilinear invariants can only express equalities of the form $p(\mu) = p(\mu^{[0]})$, where p is a second-degree polynomial in the $|P|$ variables $\{\mu_1, \ldots, \mu_{|P|}\}$. If we consider N' in Fig. 4, we see that it satisfies an invariant corresponding to a third-degree polynomial,

$$\forall\mu \in \mathcal{R}(N), \mu_1^2\mu_2 + \mu_1\mu_3 + \mu_4 = (\mu_1^{[0]})^2\mu_2^{[0]} + \mu_1^{[0]}\mu_3^{[0]} + \mu_4^{[0]} .$$

In this paper, we restrict ourselves to (linear) p-semiflows for the class $\mathcal{N}_s$ (hence $\mathcal{N}_r$, $\mathcal{N}_p$, $\mathcal{N}_t$, and $\mathcal{N}_l$), that is, solutions $x \in \mathbb{N}^{|P|}, x \neq 0$ to the equation $xD = 0$.

Theorem 1. The set of p-semiflows of $N = (P, T, D^-, D^+, \mu^{[0]}) \in \mathcal{N}_s$ equals the set of p-semiflows of $N' = (P, T', D'^-, D'^+, \mu^{[0]}) \in \mathcal{N}_o$, where $T' = T \cup \{t_{(j,l)} : 1 \le j \le |T|, 1 \le l \le |P|\}$ and $D'^-_{i,j} = \alpha^-_{i,j}$, $D'^+_{i,j} = \alpha^+_{i,j}$, $D'^-_{i,(j,l)} = \beta^-_{i,j,l}$, $D'^+_{i,(j,l)} = \beta^+_{i,j,l}$.

Proof: the entries of the incidence matrix D of N are

$$D_{i,j}(\mu) = \alpha^+_{i,j} - \alpha^-_{i,j} + \sum_{1 \le l \le |P|} (\beta^+_{i,j,l} - \beta^-_{i,j,l})\mu_l = \alpha_{i,j} + \sum_{1 \le l \le |P|} \beta_{i,j,l}\mu_l \ ,$$

where $\alpha = \alpha^+ - \alpha^-$, and $\beta = \beta^+ - \beta^-$. A p-semiflow for N, $x \in \mathbb{N}^{|P|}$, $x \neq 0$, satisfies $xD = 0$, resulting in the $|T|$ equations

$$\forall j, 1 \le j \le |T|, \ \sum_{1 \le i \le |P|} x_i \left(\alpha_{i,j} + \sum_{1 \le l \le |P|} \beta_{i,j,l}\mu_l \right) = 0 \ , \tag{1}$$

which can be rearranged into

$$\forall j, 1 \le j \le |T|, \ \sum_{1 \le i \le |P|} x_i \alpha_{i,j} + \sum_{1 \le l \le |P|} \left(\sum_{1 \le i \le |P|} x_i \beta_{i,j,l} \right) \mu_l = 0 \ .$$

The above equations must be satisfied by x independent of the marking μ, hence, in particular, they must be satisfied by the empty marking, resulting in

$$\forall j, 1 \le j \le |T|, \ \sum_{1 \le i \le |P|} x_i \alpha_{i,j} = 0 \ , \tag{2}$$

and by the $|P|$ markings having one token in place $l, 1 \le l \le |P|$, and no tokens elsewhere, resulting in

$$\forall j, 1 \le j \le |T|, \forall l, 1 \le l \le |P|, \ \sum_{1 \le i \le |P|} x_i \alpha_{i,j} + \sum_{1 \le i \le |P|} x_i \beta_{i,j,l} = 0 \ ,$$

which, given (2), imply

$$\forall j, 1 \le j \le |T|, \forall l, 1 \le l \le |P|, \ \sum_{1 \le i \le |P|} x_i \beta_{i,j,l} = 0 \ . \tag{3}$$

But these are exactly the constraints satisfied by the p-semiflows of N': (2) are imposed by the set of transitions $T \subseteq T'$ and (3) are imposed by the set of transitions $T' \setminus T$, hence $xD' = 0$. On the other hand, any p-semiflow x for N', solution of $xD' = 0$, satisfies (1) as well, and is therefore a p-semiflow for N. □

This result has an intuitively appealing interpretation. Consider, for example, Fig. 5, containing both $N \in \mathcal{N}_s$ and its corresponding $N' \in \mathcal{N}_o$ (when $\forall i, 1 \le i \le |P|, \beta^-_{i,j,l} = \beta^+_{i,j,l} = 0$, transition $t_{(j,l)}$ has empty input and output bags and can be removed from N'). Clearly, the effect of firing t_1 in N when the marking

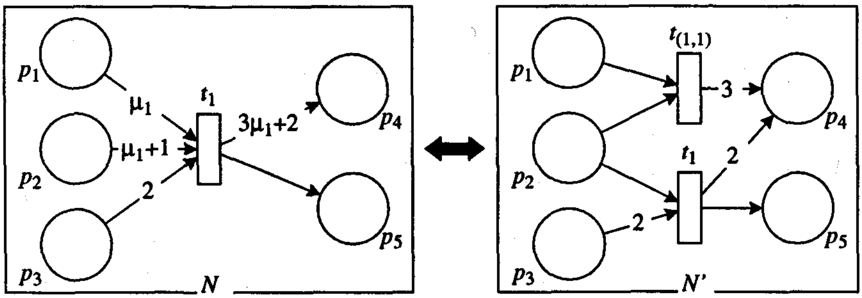

Fig. 5. Computing p-semiflows for a net $N \in \mathcal{N}_s$.

is μ can be simulated by firing t_1 once and $t_{(1,1)}$ μ_1 times in N'. The constraints imposed on x by this example are:

$$\text{from (2)}: \; 0x_1 + (-1)x_2 + (-2)x_3 + 2x_4 + 1x_5 = 0$$
$$\text{from (3)}: \; (-1)x_1 + (-1)x_2 + 0x_3 + 3x_4 + 0x_5 = 0$$

resulting in the four minimal p-semiflows shown in Fig. 6.

x_1	x_2	x_3	x_4	x_5
1	2	0	1	0
3	0	1	1	0
0	3	0	1	1
0	0	1	0	2

Fig. 6. The minimal p-semiflows of N in Fig. 5.

In ordinary P/T-nets, if, for a given t_j, $\alpha_{\bullet,j} \leq 0$ (or $\alpha_{\bullet,j} \geq 0$), no place p_i for which $\alpha_{i,j} < 0$ (or $\alpha_{i,j} > 0$) can be covered by a linear structural p-invariant. In linear transfer and self modifying P/T nets, in addition, p_i cannot be covered if, for some l, $\beta_{\bullet,j,l} \leq 0$ (or $\beta_{\bullet,j,l} \geq 0$) and $\beta_{i,j,l} < 0$ (or $\beta_{i,j,l} > 0$). This implies that a necessary condition for the net to be completely covered by linear structural p-invariants is that, for each $t_j \in T$, the set of places appearing in the marking dependent expressions of the input and output arc cardinalities must coincide. In particular, then, a place p_i such that $\beta^-_{i,j} = 1$ in a reset P/T-net (or such that $\beta^+_{i,j,l} > 0$ for some l, in a post self modifying P/T-net) cannot be covered by a linear structural p-invariant.

The approach just used for the computation of minimal p-semiflows can still be applied even if arbitrary non-negative integer functions are used in the specification of the marking-dependent arc cardinalities, instead of just linear expressions. Consider, for example, the net N in Fig. 7. Informally, the same approach still applies if we consider $\mu_3\mu_4$ and μ_1^2 as "atomic terms", hence

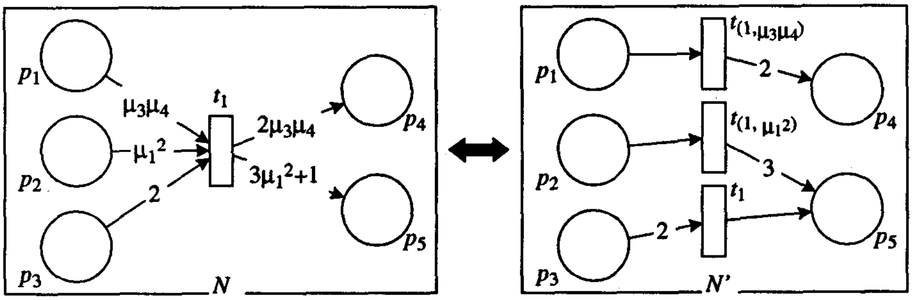

Fig. 7. P-semiflows in a net with nonlinear arc cardinalities.

the net N' on the right in Fig. 7 has the same p-semiflows as N. A different transition can be defined for each atomic term, such as $t_{(1,\mu_3\mu_4)}$ and $t_{(1,\mu_1^2)}$ in N'. However, elaborate symbolic manipulations might be required to establish nonlinear integer equalities among atomic terms, (e.g., $\lfloor e^{\mu_1} \rfloor + 1 = \lceil e^{\mu_1} \rceil$, or $\mu_1^2 + \mu_2^2 + 2\mu_1\mu_2 = (\mu_1 + \mu_2)^2$). P/T-nets with such nonlinear behavior, though, are unlikely to be found in practical modeling applications, so we do not consider them further.

3.2 T-Semiflows

The existence of t-semiflows, solutions $y \in \mathbb{N}^{|T|}$, $y \neq 0$, for the equation $Dy = 0$ (where y is a column vector), is also an important property of P/T-nets. If y is interpreted as the firing vector of a sequence $\sigma \in T^*$, that is, y_j is the number of times t_j appears in σ, then $\forall \mu \in \mathbb{N}^{|P|}, \mu = \mathcal{M}(\sigma, \mu)$, provided σ can be fired starting in μ.

Unfortunately, with marking-dependent arc cardinalities, the definition of t-semiflows, let alone their existence, becomes a problem. If we naively carry on the product Dy, we obtain:

$$\forall i, 1 \leq i \leq |P|, \sum_{1 \leq j \leq |T|} \left(\alpha_{i,j} + \sum_{1 \leq l \leq |P|} \beta_{i,j,l}\mu_l \right) y_j \ ,$$

but these $|P|$ equations are semantically incorrect, since the usage of μ_l is ambiguous. When multiplied by $\beta_{i,j_1,l}$, μ_l refers to the number of tokens in p_l when t_{j_1} fires, but, when multiplied by $\beta_{i,j_2,l}$, it refers to the number of tokens in p_l when t_{j_2} fires, which can be a different quantity (this quantity can vary even between two firings of the same transition). The following example typifies the problem. Consider N in Fig. 8 where the equation $Dy = 0$ results in

$$\begin{bmatrix} -\mu_1 & \mu_1 \\ \mu_1 & -\mu_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 0 \Rightarrow \ -\mu_1 y_1 + \mu_1 y_2 = 0 \Rightarrow \ y_1 = y_2 \ ,$$

suggesting the minimal t-semiflow $y = [1,1]$. This is clearly incorrect, since $[0, \mu_1^{[0]} + \mu_2^{[0]}] = \mathcal{M}((t_1, t_2), [\mu_1^{[0]}, \mu_2^{[0]}])$. Indeed, N has an absorbing marking $[0, \mu_1^{[0]} + \mu_2^{[0]}]$, which is reached as soon as t_1 fires (t_1 and t_2 are still enabled, but their firing does not change the marking).

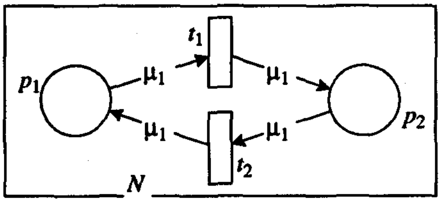

Fig. 8. A net with no t-invariants.

This shows how the existence of t-semiflows is irreparably compromised by marking-dependent arc cardinalities, unless the marking-dependent arc cardinalities themselves happen to be irrelevant, as in the case of Fig. 9, where the only difference between N and N' is that the recurrent markings are $[\mu_1^{[0]} + \mu_2^{[0]}, 0]$ and $[0, \mu_1^{[0]} + \mu_2^{[0]}]$ in the former, and $[1,0]$ and $[0,1]$ in the latter. Both nets have the minimal t-semiflow $y = [1,1]$.

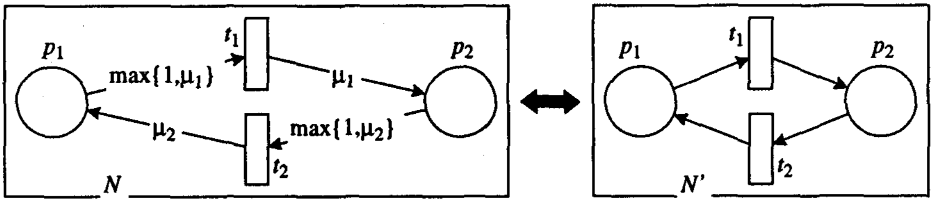

Fig. 9. A net with marking-dependent arc cardinalities and an equivalent one without.

Some useful information about firing sequences can still be obtained, though, even in the presence of marking-dependent arc cardinalities. An important reason to study t-semiflows when modeling a system is to establish relationships among the throughputs (firing rates) of various transitions. For example, assume that we are interested in performing a steady-state simulation of N in Fig. 10, which has minimal t-semiflows $y^{(1)} = [1,1,1,0,1]$ and $y^{(2)} = [0,1,0,1,0]$. We can say that $y^{(1)}$ "fires" at the unknown rate $\phi^{(1)}$, and that $y^{(2)}$ "fires" at the unknown rate $\phi^{(2)}$. If ϕ_j is the throughput of transition t_j, we can then conclude that $\phi_1 = \phi_3 = \phi_5 = \phi^{(1)}$, $\phi_2 = \phi^{(1)} + \phi^{(2)}$, and $\phi_4 = \phi^{(2)}$, hence we could collect statistics only for t_2 and t_4, and still obtain the throughput of all five transitions ($\phi_1 = \phi_3 = \phi_5 = \phi_2 - \phi_4$).

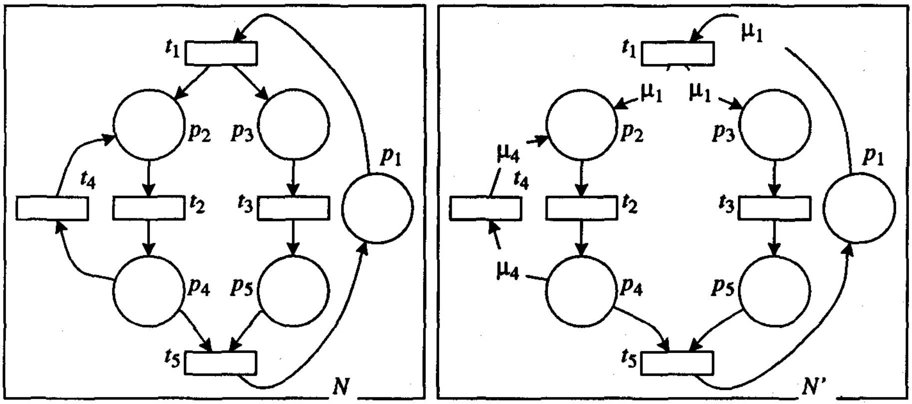

Fig. 10. An example.

Consider now the case of $N' \in \mathcal{N}_l$ in Fig. 10, where

$$D = \begin{bmatrix} -\mu_1 & & & & 1 \\ \mu_1 & -1 & & \mu_4 & \\ \mu_1 & & -1 & & \\ & 1 & & -\mu_4 & -1 \\ & & 1 & & -1 \end{bmatrix} \quad \text{and} \quad Dy = 0 \;\Rightarrow\; \begin{cases} -\mu_1 y_1 + y_5 = 0 \\ \mu_1 y_1 - y_2 + \mu_4 y_4 = 0 \\ \mu_1 y_1 - y_3 = 0 \\ y_2 - \mu_4 y_4 - y_5 = 0 \\ y_3 - y_5 = 0 \end{cases} .$$

Defining $y'_1 = \mu_1 y_1$ and $y'_4 = \mu_4 y_4$, a set of equations in $\{y'_1, y_2, y_3, y'_4, y_5\}$ results, with the same coefficients as in the constant case. Hence, we can compute the following "pseudo t-semiflows": $y^{(1)} = [1/\mu_1, 1, 1, 0, 1]$ and $y^{(2)} = [0, 1, 0, 1/\mu_4, 0]$. These can then be used to compute bounds on the ratio between throughputs, provided we know bounds on the minimum and maximum values that μ_1 and μ_4 can assume when t_1 and t_4 fire, respectively (these, of course, depend on $\mu^{[0]}$).

Given the initial marking and the fact that N' is covered by p-semiflows, we can compute the maximum number of tokens in p_1 and p_4 as $M_1 = \mu_1^{[0]} + \min\{\mu_2^{[0]} + \mu_4^{[0]}, \mu_3^{[0]} + \mu_5^{[0]}\}$, and $M_4 = \mu_1^{[0]} + \mu_2^{[0]} + \mu_4^{[0]}$, respectively. Furthermore, the system might impose a minimum threshold $m_1 = \delta^-_{1,1}$ and $m_4 = \delta^-_{4,4}$ on the number of tokens required in p_1 and p_4 before t_1 and t_4 can fire, respectively. Then, the pseudo t-semiflows still fire at an unknown rate $\phi^{(1)}$ and $\phi^{(2)}$. We can then conclude that $\phi_1 \in [\phi^{(1)}/M_1, \phi^{(1)}/m_1]$, $\phi_2 = \phi^{(1)} + \phi^{(2)}$, $\phi_3 = \phi_5 = \phi^{(1)}$, and $\phi_4 \in [\phi^{(2)}/M_4, \phi^{(2)}/m_4]$.

If p_1 is unbounded ($M_1 = \infty$), we only obtain the upper bound $\phi_1 < \phi^{(1)}/m_1$. If t_1 can fire when p_1 is empty ($m_1 = 0$), we only obtain the lower bound $\phi_1 > \phi^{(1)}/M_1$ and, in the particular case of N in Fig. 10, the throughput of t_1 is then simply equal to the inverse of its average firing time, since t_1 is always enabled.

Realistically, though, it is more likely that, in such a system, we are interested in computing the throughput of t_1 counting only the firings that change the marking, which is equivalent to saying $m_1 = 1$. A meaningful upper bound on ϕ_1 as a function of $\phi^{(1)}$ can then be obtained.

4 Expressive Power

When modeling a system with a P/T-net, it is important to know whether the net is bounded or live, and whether certain markings are reachable. State-space-based analytical approaches, for example, require to generate and store the entire reachability graph, which must then be finite. Some partial answers might be obtainable from invariant analysis (a net is bounded if it is completely covered by p-semiflows), but more complex issues of decidability exist. Traditionally, these have been studied by addressing P/T-nets as language generators.

Given a P/T-net $N = (P, T, D^-, D^+, \mu^{[0]})$, a set of final markings $\mathcal{F} \subseteq \mathcal{R}(N)$, and a labelling function $\sigma : T \to \Sigma \cup \lambda$, where $\lambda \notin \Sigma$ denotes the empty string, we can define the language $\mathcal{L}(N, \mathcal{F}, \sigma) = \{\sigma(s) : s \in T^*, \mathcal{M}(s, \mu^{[0]}) \in \mathcal{F}\}$. Petri net languages can then be classified according to three parameters:

1. The type of P/T-net: $\mathcal{N}_o$, $\mathcal{N}_r$, $\mathcal{N}_p$, $\mathcal{N}_t$, $\mathcal{N}_l$, and $\mathcal{N}_s$.
2. The set of final states $\mathcal{F}$ [21]:
   $\mathcal{F} = \mathcal{R}(N)$: *P*-type, any reachable marking is a final state.
   $\mathcal{F} = \{\mu_1^f, \mu_2^f, \ldots, \mu_k^f\}$: *L*-type, a finite set of reachable markings.
   $\mathcal{F} = \{\mu \in \mathcal{R}(N) : \exists \mu^f \in \{\mu_1^f, \mu_2^f, \ldots, \mu_k^f\}, \mu \geq \mu^f, \}$: *G*-type, any reachable marking covering one in a finite set of markings.
   $\mathcal{F} = \{\mu \in \mathcal{R}(N) : \mathcal{E}(\mu) = \emptyset\}$: *T*-type, any terminal (absorbing, dead) marking is a final state.
3. The labelling function σ [17]:
   $\forall t \in T, \sigma(t) \in \Sigma \wedge \forall t_1, t_2 \in T, \sigma(t_1) = \sigma(t_2) \Rightarrow t_1 = t_2$: *Free*, each transition has a different label in Σ.
   $\forall \mu \in \mathcal{R}(N), \forall t_1, t_2 \in \mathcal{E}(\mu), t_1 \neq t_2 \Rightarrow \sigma(t_1) \in \Sigma \wedge \sigma(t_2) \in \Sigma \wedge \sigma(t_1) \neq \sigma(t_2)$: *Deterministic*, each enabled transition has a different label, and, if its label is λ, it is the only one enabled.
   $\forall t \in T, \sigma(t) \in \Sigma$: *Non-$\lambda$*, each transition has a non-empty label.
   $\forall t \in T, \sigma(t) \in \Sigma \cup \{\lambda\}$: λ, no restrictions.

Using a notation similar to that of Peterson [21], X_z^y indicates the class of languages which can be generated by a PN of type z (o, r, p, t, l, and s) using a final state definition of type X (P, L, G, or T), and a labelling function of type y (f, d, n, λ).

From a modeling point of view, λ-transitions are in some way analogous to immediate transitions, which fire in zero time [1], and non-λ-transitions are analogous to timed transitions. One major difference, though, is that immediate transitions have an implicit priority over timed transitions, since they will fire before any timed transition with probability one. Since P/T-nets with priorities can model Turing machines, the introduction of immediate transitions could destroy our ability to analyze the net. Two possible solutions are:

- Introduce a new type of labelling, "exclusive-λ" that does not require an implicit priority: $\forall \mu \in \mathcal{R}(N), \forall t_1, t_2 \in \mathcal{E}(\mu), \sigma(t_1) = \lambda \Rightarrow \sigma(t_2) = \lambda$.
  This is more general than a non-λ labelling, but less general than a λ-labelling. Unfortunately, to the best of our knowledge, this type of labelling has not been considered by researchers.
- Restrict ourselves to k-prompt nets [15], where no more than k λ-transitions can fire consecutively. Such a net can be transformed into a non-λ net, λ-transitions are just a shorthand. This is in particular true of a bounded net, where the only infinite sequences of λ-transitions must return infinitely often to the same marking. This fact was used in [7] to show that immediate transitions can be eliminated from GSPN families if the net is bounded. Unfortunately, there are practical systems which do not fall in this category. Consider for example the GSPN in Fig. 11, modeling a queue with batch arrivals of arbitrary size. The two immediate transitions t_2 and t_3, drawn with a thin line, cannot be eliminated, since the net is unbounded and not k-prompt. The "reduced reachability graph" shown on the right is the results of the elimination of the vanishing markings, that is, markings that enable only immediate transitions. Note that, in this net, immediate transitions are not required to have implicit priority over timed transitions, since place p_{run} enforces mutual exclusion between timed and immediate transitions. This is an example of exclusive-λ labelling.

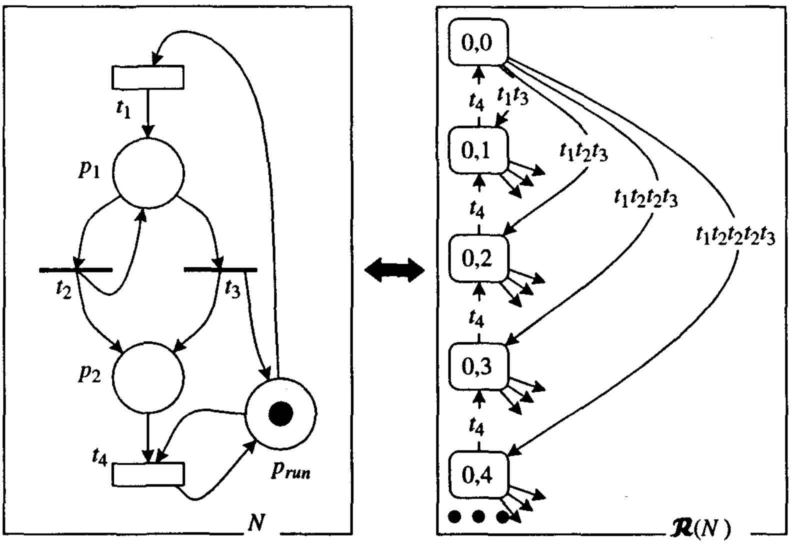

Fig. 11. A case where immediate transitions cannot be eliminated.

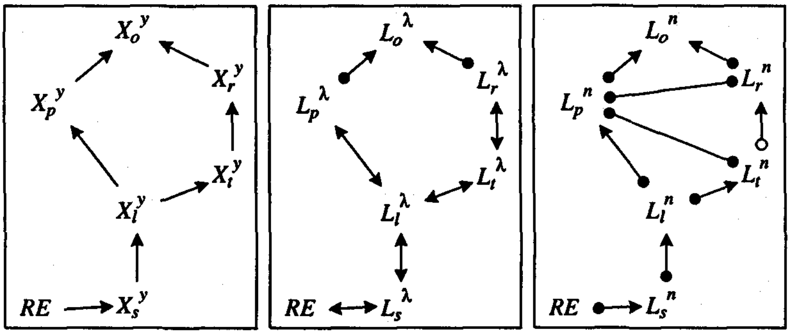

Fig. 12. Relations among some Petri net languages.

Given the definition of $\mathcal{N}_o$, $\mathcal{N}_r$, $\mathcal{N}_p$, $\mathcal{N}_t$, $\mathcal{N}_l$, and $\mathcal{N}_s$, we can immediately derive the relations shown on the left portion of Fig. 12, for any X and y (on an edge (a,b), an arrowhead on b signifies that any language in b is also a language in a, a full, or empty, circle on b signifies that there are, or we conjecture the existence of, languages in b which are not in a, respectively).

For $L_\bullet^\lambda$ languages, it is known that L_o^λ cannot generate some context-free language [17], while L_r^λ or L_p^λ can model a counter automaton, hence they allow to generate any recursive enumerable language (RE) [3, 24], resulting in the center portion of Fig. 12. On the other hand, it is also known that, for non-λ, hence k-prompt, languages, there is a strict hierarchy as shown in the right portion of Fig. 12 [25]. The only relations shown in Fig. 12 and not discussed before in the referenced literature are those regarding transfer nets. Since, as pointed out in Sect. 2.2, transfer behavior includes reset behavior, the only relations that need to be proven are:

L_p^n •–L_t^n: consider the language

$$\mathcal{L}_1 = \{wcd^{f(w)} : w \in \{a,b\}^*\} \ ,$$

where f is the value of w in base 3 when $a = 1$ and $b = 2$: $f(\lambda) = 0$, $f(wa) = 3f(w)+1$, $f(wb) = 3f(w)+2$. Using an argument similar to that of Peterson's [21], we can show that $\mathcal{L}_1$ cannot be in L_t^n (hence in L_r^n or L_o^n). Assume that $\mathcal{L}_1 \in L_t^n$, that is, $\mathcal{L}_1 = \mathcal{L}(N, \mathcal{F}, \sigma)$, where $N = (P, T, D^-, D^+, \mu^{[0]})$ and $\forall t \in T, \sigma(t) \in \{a,b,c,d\}$. Then after $k = |w|$ firings in N, at most

$$n(k) = \mu^{[0]} + k \max_{1 \le j \le |T|} \left\{ \sum_{1 \le i \le |P|} \alpha_{i,j} \right\}$$

tokens can be in the net, hence at most

$$\binom{n(k) + |P| - 1}{n(k)} = \frac{(n(k) + |P| - 1) \cdots (n(k) + 1)}{(|P| - 1)!} < (n(k) + |P|)^{|P|}$$

different markings can be reached in k firings. But $n(k)$ is linear in k, hence $(n(k)+|P|)^{|P|} < 2^k$ for a sufficiently large k, and, after k firings, N cannot distinguish between the 2^k possible $w \in \{a,b\}^k$, that is, $\exists w_1, w_2 \in \{a,b\}^k$, $w_1 \neq w_2$, such that $\mathcal{M}(w_1,\mu^{[0]}) = \mathcal{M}(w_2,\mu^{[0]})$. Then, $\mathcal{M}(w_1 cd^{f(w_1)},\mu^{[0]}) = \mathcal{M}(w_2 cd^{f(w_1)},\mu^{[0]})$, while we should have $w_1 cd^{f(w_1)} \in \mathcal{F}$ and $w_2 cd^{f(w_1)} \notin \mathcal{F}$, a contradiction.

The net $N' \in L^n_p$ in Fig. 13, instead, generates $\mathcal{L}_1$ with a non-λ labelling and $\mathcal{F} = \{[0,0,1]\}$.

$L^n_p \mathrel{-\!\!\bullet} L^n_t$: follows from $L^n_p \mathrel{-\!\!\bullet} L^n_r$ [25] and $L^n_t \to L^n_r$. We conjecture that

$$\mathcal{L}_2 = \{a^{m_1} cb^{n_1} d \cdots a^{m_k} cb^{n_k} d : k \in \mathbb{N}, \forall i, 1 \le i \le k, m_i \ge n_i\}$$

is not in L^n_p, but it is generated by $N \in \mathcal{N}_r$ in Fig. 14, with $\mathcal{F} = \{[1,0,0]\}$.

$L^n_l \mathrel{\bullet\!\!\to} L^n_t$: follows from $L^n_p \mathrel{\bullet\!\!-} L^n_t$ and $L^n_l \to L^n_p$.

$L^n_t \mathrel{\bullet\!\!-} L^n_r$: we conjecture that

$$\mathcal{L}_3 = \{a^{m_1} cb^{n_1} d \cdots a^{m_k} cb^{n_k} def^l : k \in \mathbb{N}, \forall i, 1 \le i \le k, m_i \ge n_i, l = \sum_{1 \le i \le k} (m_i - n_i)\}$$

is not in L^n_r, but it is generated by $N' \in \mathcal{N}_t$ in Fig. 14, with $\mathcal{F} = \{[0,0,0,1,0]\}$.

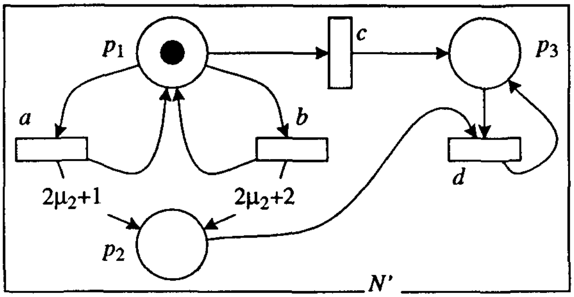

Fig. 13. A language in L^n_p but not in L^n_t.

5 Practical Applications of Variable Cardinality Arcs

From the previous sections, it should be apparent that P/T-nets with marking-dependent arc cardinalities have the potential to express certain system behaviors more naturally than ordinary P/T-nets. This is of particular interest when solving large SPN models.

Many SPN models have exponentially distributed or zero firing times, hence an underlying continuous-time Markov chain, as in GSPNs [1, 10], or exponentially distributed or constant, including zero, firing times, hence an underlying

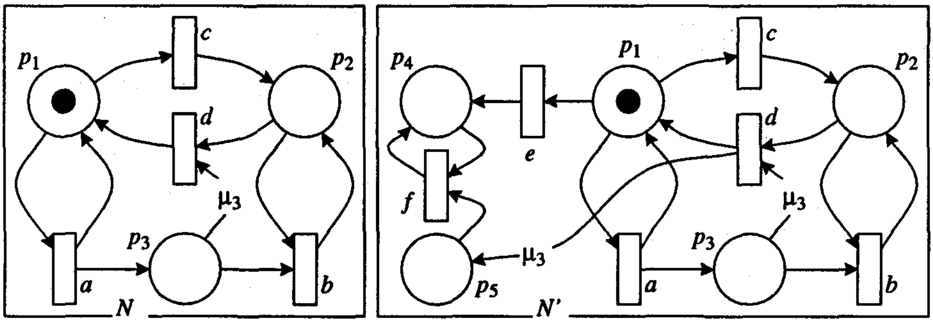

Fig. 14. Conjectures: a language in L_r^n but not in L_p^n and one in L_t^n but not in L_r^n.

semi-regenerative process, as in DSPNs [2, 10]. With finite reachability sets, the marking-dependent arc cardinalities are not needed in theory, since the behavior of the system is described by a large finite automaton, but, in practice, they improve the description, resulting in a more concise model, and allow a more efficient generation of the state space.

In all cases, the behavior to be modeled is the transfer of all tokens from one or more places to one or more other places, often as a result of a some kind of reset action. A common pattern, then, is that of N in Fig. 15. If ordinary P/T-nets are used, a subnet similar to that of N' in Fig. 15 must be used, where, in the usual notation of SPNs, immediate transitions, having a zero firing time, are drawn with a thin bar. Alternative approaches could use transitions priorities or guards or, when the net is bounded, complementary places, but the amount of clutter would be analogous. Clearly, N and N' have the same stochastic behavior, but the readability of N' is seriously hindered.

Furthermore, computer-based packages used for the numerical solution of these models [6, 13, 18, 19] generate the entire reachability graph. Structural simplifications aimed at removing immediate transitions before the generation of the reachability graph are sometimes possible [7]. These technique apply to GSPN families, that is, they are independent of the initial marking $\mu^{[0]}$, and substitute a timed transition t_i, plus one or more immediate transitions, with a set of timed transitions. They are not applicable to our case, though, since the number of timed transitions to be introduced is dependent on the maximum number of tokens to be removed, hence is not obtainable from a structural analysis of the net. If we are willing to perform simplifications that take into account the initial marking, then an equivalent SPN can always be obtained from a GSPN, provided the reachability set is finite, but the efficiency of this approach is questionable.

With N' as input, then, a computer tool has little choice but generating long sequences of vanishing markings, which are then eliminated either during or after the generation of the reachability graph.

The resulting reduced reachability graph is the same as that generated by N, but the computational effort is larger. The total number of additional reachability graph nodes and arcs corresponding to the vanishing markings due to $t_{(1,0)}$, $t_{(1,1)}$, and $t_{(1,2)}$ could be as large as the number of "tangible" (enabling only timed transitions) markings. In fact, for each tangible marking μ with a token in p_3 there is a corresponding vanishing marking μ' with a token in $p_{(1,0)}$ with $\mu'_1 = \mu_1$, $\mu'_2 = \mu_2$, and $\mu'_3 = \mu_3$.

The number of additional reachability graph arcs is even larger if the inhibitor arc from p_2 to $t_{(1,1)}$ is omitted, since, in this case, all the possible interleavings of $t_{(1,1)}$ and $t_{(1,2)}$ are generated. The resulting stochastic process is still the same as when the firings of $t_{(1,2)}$ are considered before those of $t_{(1,1)}$, since both transitions have zero firing times, but more arcs will be generated. Fortunately, this situation, "compatible" immediate transitions, is recognizable at the structural level in this case [4].

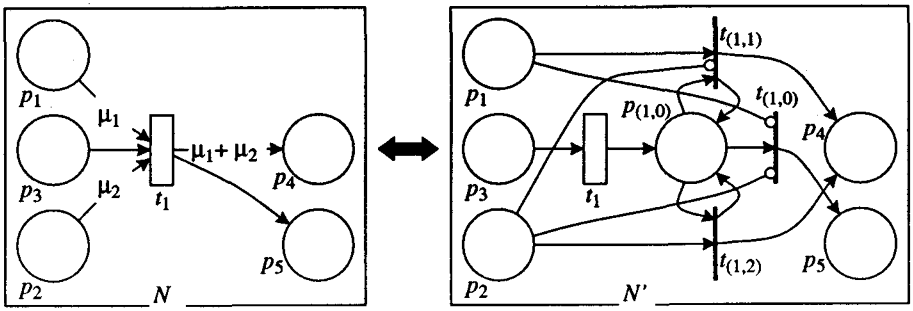

Fig. 15. A common pattern when modeling closed systems.

In the following, we give a few references to recent practical modeling applications where marking-dependent arc cardinalities are used, or could have been used, advantageously. In all cases, the stochastic behavior is that of a closed system and the net is covered by p-semiflows.

[5] models the availability of a replicated file system using CTMC-based SPNs. Various hosts can fail and be repaired, hence a read or write request for a file can be satisfied by any working host, as long as there is a quorum to certify that the host has an up-to-date copy. Both the quorum-gathering process and the mechanism by which hosts not included into the quorum for a write request become out-of-date require to move a marking-dependent number of tokens. As an ordinary P/T-net was used, the resulting model is quite complex.

[12] models the productivity of a flexible manufacturing system with CTMC-based stochastic reward nets. When finished parts are ready to leave the factory, they are gathered in a place, from where they are removed periodically, in bulk. This behavior closely reflects reality, where the means used

for transportation do not usually carry items individually. A finished part of type $P12$ is obtained by assembling two different parts, $P1$ and $P2$. To maintain a closed environment, k raw parts $P1$ and $P2$ enter the system whenever k finished parts $P12$ leaves. The resulting model is a linear transfer P/T-net.

[11] studies the performance of a simple protocol with DSPNs. In this protocol, if no acknowledgment is received within a given constant time, a timeout occurs and the current message is retransmitted. This can happen either because the message was not received properly at the other end or because of delays. In the resulting P/T-net, these two cases correspond to having a token in one of two places. When the timeout occurs, though, the token must be removed, independently of its position. This is easily accomplished using a linear transfer P/T-net with a single transition which resets both places. In this case, though, the same behavior could be represented by an ordinary P/T-net with two transitions, one to remove the token if it is in the first place, the other if it is in the second place.

[16] analyzes the availability of a VAXcluster system. When a node fails, the cluster attempts to reconfigure itself. Failed nodes with uncovered faults remain in a waiting place until either an automatic reboot isolates them, or until the number of working nodes fails below a threshold. At this point, all failed nodes are transferred to a place where they are repaired. This movement is easily accomplished by a transfer P/T-net.

[20] studies self-stability measures for a fault-tolerant clock synchronization system with DSPNs. At each cycle, the operable clocks "choose" independently with a Bernoulli probability whether they will be still operable at the next cycle, or they will have failed. After this decision is made, the tokens corresponding to the clocks which did not fail must be moved back to the place containing the operable clocks, in zero time. Since an ordinary P/T-net is used, these tokens are moved back one at a time with an immediate transition, with a negative effect on the readability of the model and on the execution time. A transfer P/T-net would have allowed to move all these tokens with the firing of a single transition.

6 Concluding Remarks

We have discussed several behaviors which, in the unbounded case, cannot be modeled by ordinary P/T-nets, yet are often needed when modeling real systems.

It is well known that *zero-testing*, the ability of firing a transition t only when a given place p is empty, for example by using inhibitor arcs, makes the formalism Turing-equivalent.

There are at least two other extensions, less known than zero-testing and complementary to each other, which are of practical interest: *zero-enforcing* ($\mathcal{N}_r$), the ability to ensure that, *after* the firing of a transition t, a given place p becomes empty, and *duplication* ($\mathcal{N}_p$), that is, the ability to add to a place p_1, after the firing of a transition t, as many tokens as there are in a place p_2.

In addition, the ability to transfer all the tokens from a place to another ($\mathcal{N}_t$), a generalization of zero-enforcing, is conjectured to be more general than strict zero-enforcing, although it cannot model duplication. On one hand this could seem obvious, since the restriction imposed on $\mathcal{N}_t$ is not dissimilar to requiring, in ordinary P/T-nets, that the number of tokens generated by a firing is not greater than the number of tokens consumed. On the other hand, in $\mathcal{N}_t$, this rule only applies to the marking-dependent portion of the arc cardinalities, not to the constant portion, so a firing can still increase the number of tokens in the net.

We conclude by observing that, from a modeling point of view, the most important feature of each subclass considered is the type of reachability graphs that it can generate. Translated into the terminology of Petri net languages, this corresponds to P-type languages, for which, unfortunately, few results are know.

One important property shared by $\mathcal{N}_o$, $\mathcal{N}_r$, $\mathcal{N}_p$, $\mathcal{N}_t$, and $\mathcal{N}_l$, but not $\mathcal{N}_s$ or inhibitor nets, is *monotonicity* [25] (for both the enabling and the firing of a transition): $t \in \mathcal{E}(\mu) \wedge \mu \leq \mu' \Rightarrow t \in \mathcal{E}(\mu') \wedge \mathcal{M}(t,\mu) \leq \mathcal{M}(t,\mu')$. Note, though, that, strict inequality implies strict inequality in the above for $\mathcal{N}_o$ and $\mathcal{N}_p$, but not for $\mathcal{N}_r$, $\mathcal{N}_t$, and $\mathcal{N}_l$. This important property strongly characterizes the structure of the reachability graph.

References

1. M. Ajmone Marsan, G. Balbo, and G. Conte. A class of Generalized Stochastic Petri Nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comp. Syst.*, 2(2):93–122, May 1984.
2. M. Ajmone Marsan and G. Chiola. On Petri Nets with deterministic and exponentially distributed firing times. In G. Rozenberg, editor, *Adv. in Petri Nets 1987, Lecture Notes in Computer Science 266*, pages 132–145. Springer-Verlag, 1987.
3. T. Araki and T. Kasami. Some decision problems related to the reachability problem for Petri nets. *Theoretical Computer Science*, 3:85–104, 1977.
4. G. Balbo, G. Chiola, G. Franceschinis, and G. Molinari Roet. On the efficient construction of the tangible reachability graph of generalized stochastic Petri nets. In *Proc. of the Int. Workshop on Petri Nets and Performance Models*, Madison, Wisconsin, Aug. 1987.
5. J. Bechta Dugan and G. Ciardo. Stochastic Petri net analysis of a replicated file system. *IEEE Trans. Softw. Eng.*, 15(4):394–401, Apr. 1989.
6. G. Chiola. A Graphical Petri Net Tool for Performance Analysis. In *Proc. 3rd Int. Conf. on Modeling Techniques and Tools for Performance Analysis, Paris*, pages 323–333, 1987.
7. G. Chiola, S. Donatelli, and G. Franceschinis. GSPNs versus SPNs: what is the actual role of immediate transitions? In *Proc. of the Fourth Int. Workshop on Petri Nets and Performance Models (PNPM91)*, Melbourne, Australia, Dec. 1991.
8. G. Ciardo. *Analysis of large stochastic Petri net models.* PhD thesis, Duke University, Durham, North Carolina, 1989.
9. G. Ciardo, A. Blakemore, P. F. J. Chimento, J. K. Muppala, and K. S. Trivedi. Automated generation and analysis of Markov reward models using Stochastic Reward Nets. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov*

Chains, and Queueing Models, volume 48 of *IMA Volumes in Mathematics and its Applications*, pages 145–191. Springer-Verlag, 1993.

10. G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. In *Proc. of the Fifth Int. Workshop on Petri Nets and Performance Models (PNPM93)*, Toulouse, France, Oct. 1993.
11. G. Ciardo and C. Lindemann. Analysis of deterministic and stochastic Petri nets. In *Proc. of the Fifth Int. Workshop on Petri Nets and Performance Models (PNPM93)*, Toulouse, France, Oct. 1993.
12. G. Ciardo and K. S. Trivedi. A decomposition approach for stochastic reward net models. *Perf. Eval.*, 18:37–59, 1993.
13. G. Ciardo, K. S. Trivedi, and J. Muppala. SPNP: stochastic Petri net package. In *Proc. of the Third Int. Workshop on Petri Nets and Performance Models (PNPM89)*, pages 142–151, Kyoto, Japan, Dec. 1989. IEEE Computer Society Press.
14. J. M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study of algorithms for the computation of minimal p-semiflows. In *10th International Conference on Application and Theory of Petri Nets*, Bonn, Germany, 1989.
15. M. Hack. Petri net languages. Technical Report 159, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachussetts, Mar. 1976.
16. O. C. Ibe and K. S. Trivedi. Stochastic Petri net modeling of VAXCluster system availability. In *Proc. of the Third Int. Workshop on Petri Nets and Performance Models (PNPM89)*, Kyoto, Japan, Dec. 1989.
17. M. Jantzen. Language theory of Petri nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Adv. in Petri Nets 1986, Part 1, Lecture Notes in Computer Science 254*, pages 397–412. Springer-Verlag, 1986.
18. G. Klas and R. Lepold. TOMSPIN, a tool for modeling with stochastic Petri nets. In *CompEuro 92*, pages 618–623, The Hague, The Netherlands, May 1992.
19. C. Lindemann. DSPNexpress: A software package for the efficient solution of Deterministic and Stochastic Petri Nets. In *Proc. 6th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 15–29, Edinburgh, Great Britain, 1992.
20. M. Lu, D. Zhang, and T. Murata. Analysis of self-stabilizing clock synchronization by means of stochastic Petri nets. *IEEE Trans. Comp.*, 39:597–604, 1990.
21. J. L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, 1981.
22. C. Petri. *Kommunikation mit Automaten.* PhD thesis, University of Bonn, Bonn, West Germany, 1962.
23. W. Reisig. *Petri Nets*, volume 4 of *EATC Monographs on Theoretical Computer Science.* Springer-Verlag, 1985.
24. R. Valk. On the computational power of extended Petri nets. In *Seventh Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 64*, pages 527–535. Springer-Verlag, 1978.
25. R. Valk. Generalizations of Petri nets. In *Mathematical foundations of computer science, Lecture Notes in Computer Science 118*, pages 140–155. Springer-Verlag, 1981.