On the expressive power of Petri nets with transfer arcs vs. Petri nets with reset arcs

Gilles GEERAERTS*

Université Libre de Bruxelles – Département d'informatique
Campus Plaine, Boulevard du Triomphe, CPI 212, B1050
Bruxelles, Belgium

Abstract

In this paper, we revisit the conjecture of [1], stating that ε-free Petri nets with transfer arcs are strictly more expressive than ε-free Petri nets with reset arcs. In [1], an ε-free Petri net with transfer arcs is provided, whose language is conjectured not to be recognizable by any ε-free Petri net with reset arcs. We show that this latter conjecture is flawed, by exhibiting an ε-free Petri net with reset arcs that accepts the same language. We also provide a new ε-free Petri net with transfer arcs, for which we conjecture too that it recognizes a language that is not definable by ε-free Petri nets with reset arcs.

1 Introduction

Since their introduction by Petri in [2], *Petri nets* are regarded as a most suitable formalism to represent *distributed systems*, i.e., systems in which the computing payload is distributed among several entities (henceforth called *processes*) that communicate.

A common way to model a distributed system by a Petri net consists in applying the idea of the *counting abstraction*, as shown in [3]. This principle consists in mapping each process to a token, and representing each state of each type of process by a place. In this case, the presence of a token in a given place p indicates that, in the current global state of the system, there is a process which is in its local state that corresponds to p. The transitions of the Petri net then consume and produce tokens to move the associated processes from one state to another. This formalization has the drawback to abstract away the actual identities of the processes. Still, some interesting properties (such as certain kind of *safety properties*) can be verified at this level of abstraction.

Unsurprisingly, Petri nets have thus broadly attracted the attention of researchers interested in the field of *computed-aided verification* for nearly twenty

*gilles.geeraerts@ulb.ac.be

years. In particular, it has been shown that certain *communication procedures* which are common when programming distributed systems, can easily be modeled by Petri nets. For instance, the classical *rendez-vous* is easily encoded [3]. Unfortunately, other communication procedures such as *broadcast* are not easily captured by plain Petri nets. For that reason, several extensions of the model have been proposed. We address two of them is this paper:

- Petri nets with *transfer* arcs, in which a transition can also consume *all* the tokens present in one place (regardless of the actual number of tokens) and move them to another place.
- Petri net with *reset* arcs, in which a transition can *delete all* the tokens present in one place.

The meaningfulness of these extensions in terms of modeling power has been demonstrated, for instance, in [4] (modeling and verification of cache-coherency protocols) or [5] (modeling and verification of Java multithread programs) to cite only a few.

It is also rather useful to consider *labeled Petri nets*, in which each transition bears a *label*, i.e., either a character taken from some finite alphabet or the silent character ε. In that case, and provided that we have an acceptance condition at our disposal (in the present case, a finite set of accepting markings), one can define *the language*[1] accepted by a given Petri net, in the usual way.

A natural question about these different extensions is then to compare their *respective expressive power*. Roughly speaking, if their exists a Petri net with transfer arcs that accepts some language L, can we build a Petri net with reset arcs that accepts L, and vice-versa ? Moreover, it is desirable that the constructions at work (if they exist) respect the 'ε-freeness' of the nets, since it is well-known that the introduction of ε-transition can sometimes strictly augment the expressive power of certain models of computation [6]. That is, if we provide a Petri net with transfer arcs *without ε-labeled transitions*, it would be interesting to be able to build a Petri net with reset arcs *without ε-labeled transitions* too.

One of the seminal papers addressing these questions is [1]. In this paper, it is shown that any Petri net with reset arcs can be turned into an equivalent Petri net with transfer arcs. However, the reverse direction is not completely covered: the construction to convert a Petri net with transfer arcs into an equivalent Petri net with reset arcs involves the introduction of ε-labeled transitions. Moreover, it is conjectured these ε-transitions are, in general compulsory, i.e., that there exists at least one language L that can be recognized by an ε-free Petri net with transfer arcs, but not by any ε-free Petri net with reset arcs. As a support to this conjecture, a Petri net with transfer arcs (henceforth called $\mathcal{N}_1$, see Fig. 1(a)) is given, whose language is conjectured not to be recognizable by a ε-free Petri net with reset arcs.

In this paper, we prove that the language of $\mathcal{N}_1$ is recognizable by a ε-free Petri net with reset arcs. This, however, does not show that Petri nets with

[1] In this case, a finite words languages.

transfer arcs are not strictly more expressive than Petri nets with reset arcs. We still leave the question open, but provide a new net, which is simpler than that of [1], and whose language we believe cannot be accepted by a ε-free Petri net with reset arcs.

2 Preliminaries

In this section, we recall the basic definition that we exploit in this paper. For the sake of brevity, we have chosen to introduce the extensions of Petri nets we are interested in along the lines of [7]. The original presentation of [1] is more general. The reader will easily convince itself that the present definitions are equivalent to those of [1].

Definition 1 *A (labeled) extended Petri net (*EPN *for short) is a tuple* $\langle P, T, \Sigma\rangle$ *where* P *is a finite set of* places*,* T *is a finite set of* transitions *and* Σ *is a finite* alphabet*.* T *is such that* $P \cap T = \emptyset$*, and every transition* t *is of the form* $\langle I, O, s, d, \lambda\rangle$*, where:*

- I *is a multiset*[2] *of* input places*;*
- O *is a multiset of* output places*;*
- $s \in P \cup \{\bot\}$ *is the* source place*;*
- $d \in P \cup \{\bot\}$ *is the* destination place*;*
- $\lambda \in \Sigma \cup \{\varepsilon\}$ *is the* label *of the transition.*

Moreover, each transition $t = \langle I, O, s, d, \lambda\rangle$*, respects* $d \neq \bot \Rightarrow s \neq \bot$*.*

In the case where, for any $t = \langle I, O, s, d, \lambda\rangle \in T$: $\lambda \neq \varepsilon$, we say that $\mathcal{N}$ is *ε-free*.

Given an EPN $\mathcal{N} = \langle P, T, \Sigma\rangle$, we define a *marking* $\mathbf{m}$ of $\mathcal{N}$ as a function $\mathbf{m} : P \mapsto \mathbb{N}$, where $\mathbb{N}$ denotes the set $\{0, 1, 2, \ldots\}$ of positive integers. We assume that the places of $\mathcal{N}$ are totally ordered, i.e., $P = \{p_1, p_2, \ldots, p_{|P|}\}$. In that case, we often see a marking $\mathbf{m}$ as vector whose i-th coordinate is $\mathbf{m}(p_i)$, and which is denoted as $\big\langle \mathbf{m}(p_1), \mathbf{m}(p_2), \ldots, \mathbf{m}(p_{|P|})\big\rangle$.

It is convenient to associate to any EPN $\mathcal{N} = \langle P, T, \Sigma\rangle$ an *initial marking* $\mathbf{m}_0$:

Definition 2 *An initialized* EPN *is a tuple* $\mathcal{N} = \langle P, T, \Sigma, \mathbf{m}_0\rangle$*, where* $\mathcal{N}' = \langle P, T, \Sigma\rangle$ *is an* EPN*, and* $\mathbf{m}_0$ *is marking of* $\mathcal{N}'$*.*

In the sequel, we will consider initialized EPN only, which we will denote simply as EPN.

Let $\mathcal{N} = \langle P, T, \Sigma, \mathbf{m}_0\rangle$ be an EPN. Depending on the form of the transitions of $\mathcal{N}$, we might classify $\mathcal{N}$ into one of the three following classes:

[2] In this paper, we often regard multisets of places as functions $P \mapsto \mathbb{N}$, where $\mathbb{N}$ is the set of positive integers. In this case, if M is a multiset of places and p is a place, then $M(p)$ is the number of occurrences of p in M.

1. If for any $\langle I, O, s, d, \lambda\rangle \in T$: $s = \bot = d$, then $\mathcal{N}$ is a (plain) Petri net (PN for short).

2. If for any $\langle I, O, s, d, \lambda\rangle \in T$: either $s = \bot = d$, or $s \neq \bot$ and $d = \bot$, then $\mathcal{N}$ is a Petri net with reset arcs (PN+R for short).

3. If for any $\langle I, O, s, d, \lambda\rangle \in T$: either $s = \bot = d$ or $s \neq \bot \neq d$, then $\mathcal{N}$ is a Petri net with transfer arcs (PN+T for short).

Remark that, according to Definition 1, there are EPN which are neither PN, nor PN+R, nor PN+T. We do not consider such EPN in the sequel.

We use the classical convention to graphically represent EPN and markings. Places are depicted by circles and transitions by filled rectangles. For any place p and transition $t = \langle I, O, s, d, \lambda\rangle$ s.t. $p \in I$ (resp. $p \in O$), we draw an arrow from p to t (t to p) labeled by $I(p)$ ($O(p)$). We omit the label when it is 1. When $s \neq \bot$ and $d = \bot$, we draw a line bearing a cross between t and s. When $s \neq \bot \neq d$, we draw a thick gray arrow from s to t and from t to d. A marking $\mathbf{m}$ is depicted by drawing $\mathbf{m}(p)$ black tokens in each place p.

Let us now explain how to associate a (finite words) language to an EPN $\mathcal{N} = \langle P, T, \Sigma, \mathbf{m}_0\rangle$. For that purpose, we need the following notions. Let $\mathbf{m}$ be a marking, and let t be a transition. Then t is said to be *enabled* in $\mathbf{m}$ iff for any $p \in P$: $\mathbf{m}(p) \geq I(p)$. In that case, t can *fire* and transform $\mathbf{m}$ into another marking $\mathbf{m}'$ computed as follows:

1. First compute $\mathbf{m}_1$ as follows. For any $p \in P$: $\mathbf{m}_1(p) = \mathbf{m}_1(p) - I(p)$.

2. Second, compute $\mathbf{m}_2$ as follows. For any place $p \in P \setminus \{s, d\}$, $\mathbf{m}_2(p) = \mathbf{m}_1(p)$. The values of $\mathbf{m}_2(s)$ and $\mathbf{m}_2(d)$ depend upon the values of s and d. If $s = \bot$, then $\mathbf{m}_2(s) = \mathbf{m}_1(s)$ and $\mathbf{m}_2(d) = \mathbf{m}_1(d)$. If $s \neq \bot$ and $d = \bot$, then $\mathbf{m}_2(s) = 0$ and $\mathbf{m}_2(d) = \mathbf{m}_1(d)$. If $s \neq \bot$ and $d \neq \bot$, then $\mathbf{m}_2(s) = 0$ and $\mathbf{m}_2(d) = \mathbf{m}_1(s)$.

3. Finally, $\mathbf{m}'$ is computed from $\mathbf{m}_2$ as follows. For any $p \in P$: $\mathbf{m}'(p) = \mathbf{m}_2(p) + O(p)$.

By careful inspection of these definitions, it is not difficult to see that the effect of a transition of the form $t = \langle I, O, p, \bot, \lambda\rangle$ (for some place p), is actually to *reset* p, i.e., delete all the tokens from p, after $I(p)$ tokens have been removed from p. Similarly, the effect of a transition of the form $t = \langle I, O, p, p', \lambda\rangle$ (for places p and p') is to *transfer* all the tokens from p to p' (after the removing of $I(p)$ tokens).

In the case where $t = \langle I, O, s, d, \lambda\rangle$ is enabled in $\mathbf{m}$ and transforms it into $\mathbf{m}'$, we might $\mathbf{m} \xrightarrow{t} \mathbf{m}'$, when we want to stress the name of the transition, or $\mathbf{m} \xrightarrow{\lambda} \mathbf{m}'$, when the label of the transition matters, or simply $\mathbf{m} \rightarrow \mathbf{m}'$, when nor the label, nor the name of the transition is relevant.

Let $\sigma = t_1 t_2 \cdots t_k$ be a (possibly empty) sequence of transitions. Let us assume that λ_i denotes the label of t_i in σ. We associate, to any such sequence

σ, its *label*, denoted by $\Lambda(\sigma)$ and defined as $\Lambda(\sigma) = \lambda_1 \cdot \lambda_2 \cdots \lambda_k$, where $\cdot$ denotes word concatenation. As usual, the empty word is denoted by ε, and, for every word w, $w \cdot \varepsilon = w = \varepsilon \cdot w$. Remark that in the case where σ is the empty sequence, we have $\Lambda(\sigma) = \varepsilon$.

Let $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{k+1}$ be $k+1$ markings such that $\mathbf{m}_1 \xrightarrow{t_1} \mathbf{m}_2 \xrightarrow{t_2} \cdot \xrightarrow{t_k} \mathbf{m}_{k+1}$. In that case, we may write $\mathbf{m}_1 \xrightarrow{\sigma} \mathbf{m}_{k+1}$, or $\mathbf{m}_1 \xrightarrow{\Lambda(\sigma)} \mathbf{m}_{k+1}$.

Finally, by associating to an EPN $\mathcal{N} = \langle P, T, \Sigma, \mathbf{m}_0 \rangle$ a finite set A of *accepting markings*, we are able to define the *language accepted by* $\mathcal{N}$. Let σ be a sequence of transitions of $\mathcal{N}$. We say that σ is *accepting* iff $\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$ with $\mathbf{m} \in A$. Then:

Definition 3 *Let* $\mathcal{N} = \langle P, T, \Sigma, \mathbf{m}_0 \rangle$ *be an* EPN *and let* A *be a finite set of markings of* $\mathcal{N}$*. Then, the* language accepted by $\mathcal{N}$ *(for accepting set* A*) is:*

$$L(\mathcal{N}, A) = \{\Lambda(\sigma) \mid \sigma \textit{ is accepting.}\}$$

Following these definitions, it is not difficult to see that the class of the languages accepted by ε-free $\mathsf{PN{+}T}$ (resp. ε-free $\mathsf{PN{+}R}$, $\mathsf{PN{+}T}$, $\mathsf{PN{+}R}$)corresponds to the class L_t^n (L_r^n, L_t^λ, L_r^λ) of [1].

Given these definitions, one natural question is to compare the respective expressive powers of the classes $\mathsf{PN{+}T}$ and $\mathsf{PN{+}R}$. The generic question is thus: is it possible to build, for any $\mathsf{PN{+}T}$ $\mathcal{N}$ and accepting set A, a $\mathsf{PN{+}R}$ $\mathcal{N}'$ and an accepting set A' s.t. $L(\mathcal{N}, A) = L(\mathcal{N}, A')$, and vice-versa ?

More particularly, we will distinguish, as is usual in language theory, between the languages that request ε-transitions in the EPN accepting it, and those that can be accepted by an ε-free EPN. Some of these questions have been answered in [1]:

Theorem 1 ([1]) *For any* $\mathsf{PN{+}R}$ $\mathcal{N}$ *and accepting set* A *of* $\mathcal{N}$*, it is possible to build a* $\mathsf{PN{+}T}$ $\mathcal{N}'$ *and an accepting set* A' *s.t.* $L(\mathcal{N}, A) = L(\mathcal{N}', A')$*. Moreover, if* $\mathcal{N}$ *is* ε*-free, then so is* $\mathcal{N}'$*.*

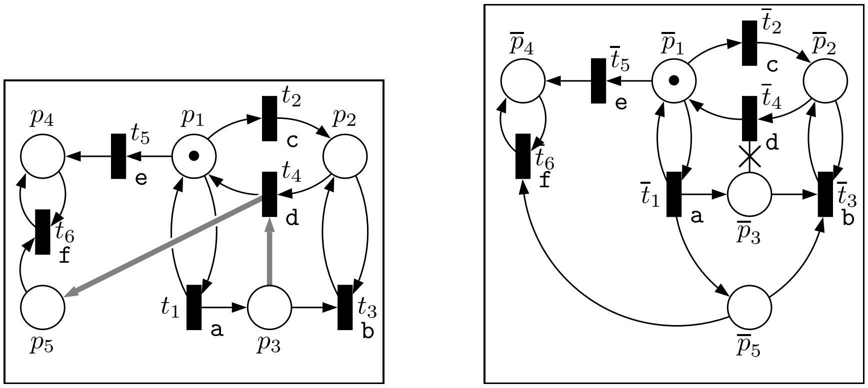
Thus, $\mathsf{PN{+}R}$ are no more expressive than $\mathsf{PN{+}T}$, even when considering ε-free nets only. On the other hand:

Theorem 2 ([1]) *For any* $\mathsf{PN{+}T}$ $\mathcal{N}$ *and any accepting set* A *of* $\mathcal{N}$*, it is possible to build a* $\mathsf{PN{+}R}$ $\mathcal{N}'$ with ε-transitions *and an accepting set* A' *s.t.* $L(\mathcal{N}, A) = L(\mathcal{N}', A')$

It is important to remark that the construction presented in [1] to prove this theorem involves the use of ε-transitions. The remaining open question is thus:

> Does there exist a ε-free $\mathsf{PN{+}T}$ $\mathcal{N}$ and any accepting set A of $\mathcal{N}$ such that there are no ε-free $\mathsf{PN{+}R}$ $\mathcal{N}'$ and no accepting set A' with $L(\mathcal{N}, A) = L(\mathcal{N}', A')$?

A positive answer would prove that $\mathsf{PN{+}T}$ are strictly more expressive than $\mathsf{PN{+}R}$ when ε-transitions are disallowed. As stated before, no answer is provided to this question in [1], but it is conjectured that the answer is 'yes', as we recall now.



(a) The PN+T $\mathcal{N}_1$ (b) The PN+R $\mathcal{N}_2$

Figure 1: The PN+T $\mathcal{N}_1$ from [1] and the PN+R $\mathcal{N}_2$ that accepts the same language.

3 A PN+T to strictly separate the languages of PN+T and PN+R?

As a support to the conjecture that PN+T are strictly more expressive than ε-free PN+R, the PN+T $\mathcal{N}_1$ (see Fig. 1(a)), together with its accepting marking $\mathbf{m}_a = \langle 1,0,0,0,0\rangle$, is introduced in [1]. $L(\mathcal{N}_1, \mathbf{m}_a)$ is thus supposed to be an example of language that cannot be accepted by an ε-free PN+R:

Conjecture 1 *[1] There is no ε-free* PN+R $\mathcal{N}$ *without and no finite set of markings S s.t. $L(\mathcal{N}, S) = L(\mathcal{N}_1, \{\mathbf{m}_a\})$.*

In this section, we discuss the language $L(\mathcal{N}_1, \{\mathbf{m}_a\})$ and provide an ε PN+R that accepts it (see Fig. 1(b)). Conjecture 1 is thus unfortunately flawed. However, in the next section, we provide a new PN+T which appears to us a good candidate to prove that ε-free PN+T are strictly more expressive than ε-free PN+R.

Let us consider $\mathcal{N}_1$ more carefully. Remark that the language $L(\mathcal{N}_1, \{\mathbf{m}_a\})$ is actually [1]:

$$\left\{ \begin{array}{c} \mathtt{a}^{m_1}\mathtt{cb}^{n_1}\mathtt{da}^{m_2}\mathtt{cb}^{n_2}\mathtt{d}\cdots\mathtt{a}^{m_k}\mathtt{cb}^{n_k}\mathtt{def}^{\ell} \\ \text{s.t.} \\ k \in \mathbb{N} \wedge \forall 1 \leq i \leq k : m_i \geq n_i \wedge \ell = \sum_{i=1}^{k}(m_i - n_i) \end{array} \right\}$$

Indeed, any accepting sequence of transitions of $\mathcal{N}_1$ can be split into two parts. In a *first part*, a token is initially present in p_1, p_3 is empty, and subsequences of the form $\sigma_i = t_1^{m_i} t_2 t_3^{n_i} t_4$ (labeled by $\mathtt{a}^{m_i}\mathtt{cb}^{n_i}\mathtt{d}$) may be fired successively (with $m_i \geq n_i$). Each time t_1 fires, a token is added to p_3. Hence, p_3 counts the number of $\mathtt{a}$'s in the current σ_i. That is, when t_2 fires, the markings of p_3 is exactly m_i. The firing of t_2 moves the token to p_2 and t_3 (labeled by $\mathtt{b}$) can fire at most m_i times, which removes one token at a time from p_3. Thus, when t_4 fires, p_3 contains $m_i - n_i$. That amount of tokens is *transferred* to p_5. At that point, the token is back in p_1, p_3 is empty, and the next subsequence σ_{i+1} can be fired. After the firing of $\sigma_1 \cdot \sigma_2 \cdots \sigma_k$, the token is in p_1, places p_2, p_3 and p_4 are empty, and p_5 contains exactly $\sum_{i=1}^{k}(m_i - n_i)$ tokens.

The net can then enter the *second part* of its execution by firing t_5 (labeled by $\mathtt{e}$) which moves the token to p_4. At that point, the only possible execution to reach $\mathbf{m}_a$ is to fire t_6 (labeled by $\mathtt{f}$) exactly $\ell = \sum_{i=1}^{k}(m_i - n_i)$ times, because ℓ is the amount of tokens in p_5, and t_6 removes exactly one token at a time from p_5.

Let us now introduce $\mathcal{N}_2$, an ε-free $\mathsf{PN{+}R}$ that accepts exactly $L(\mathcal{N}_1, \{\mathbf{m}_a\})$. It can be found in Fig. 1(b).

Let us show that $\mathcal{N}_2$ indeed accepts the same language as $\mathcal{N}_1$:

Proposition 1 $L(\mathcal{N}_2, \{\mathbf{m}_a\}) = L(\mathcal{N}_1, \{\mathbf{m}_a\})$.

Proof. First, remark that $\mathcal{N}_1$ and $\mathcal{N}_2$ have a very similar structure. Let p_i be a place of $\mathcal{N}_1$. We say that the place $\overline{p}_i$ is its *corresponding* place in $\mathcal{N}_2$ (and vice-versa: there is thus a one-to-one correspondence between the places of $\mathcal{N}_1$ and $\mathcal{N}_2$). Similarly, to each transition t_i of $\mathcal{N}_1$, corresponds one and only one transition $\overline{t}_i$ of $\mathcal{N}_2$. Thus, to any sequence of transitions σ of $\mathcal{N}_1$ corresponds one and only one sequence of transitions $\overline{\sigma}$ of $\mathcal{N}_2$, with $\Lambda(\sigma) = \Lambda(\overline{\sigma})$. We prove that $L(\mathcal{N}_2, \{\mathbf{m}_a\}) \supseteq L(\mathcal{N}_1, \{\mathbf{m}_a\})$ and $L(\mathcal{N}_2, \{\mathbf{m}_a\}) \subseteq L(\mathcal{N}_1, \{\mathbf{m}_a\})$ independently:
$L(\mathcal{N}_2, \{\mathbf{m}_a\}) \supseteq L(\mathcal{N}_1, \{\mathbf{m}_a\})$: Let us consider the generic accepting sequence of transitions $\sigma = t_1^{m_1} t_2 t_3^{n_1} t_4 t_1^{m_2} t_2 t_3^{n_2} t_4 \cdots t_1^{m_k} t_2 t_3^{n_k} t_4 t_5 t_6^{\ell}$ of $\mathcal{N}_1$, with $\ell = \sum_{i=1}^{k}(m_i - n_i)$, and let $\overline{\sigma}$ be its corresponding sequence in $\mathcal{N}_2$. Let us show that $\overline{\sigma}$ is firable and leads to $\mathbf{m}_a$. For any $1 \le i \le k$, let $\sigma_i = t_1^{m_i} t_2 t_3^{n_i} t_4$, and let $\overline{\sigma}_i$ be its corresponding sequence. Moreover, for any $1 \le i \le k$, let $\mathbf{m}_i$ be s.t. $\mathbf{m}_0 \xrightarrow{\sigma_1 \cdots \sigma_i} \mathbf{m}_i$ (where $\mathbf{m}_0$ is the initial marking of $\mathcal{N}_1$), and let $\overline{\mathbf{m}}_i$ be s.t. $\overline{\mathbf{m}}_0 \xrightarrow{\overline{\sigma}_1 \cdots \overline{\sigma}_i} \overline{\mathbf{m}}_i$ (where $\overline{\mathbf{m}}_i$ is the initial marking of $\mathcal{N}_2$). According to [1], we know that for any $1 \le i \le k$: $\mathbf{m}_i\big(\{p_2, p_3, p_4\}\big) = 0$, $\mathbf{m}_i(p_1) = 1$ and $\mathbf{m}_i(p_5) = \sum_{j=1}^{i}(m_j - n_j)$. It is easy to show by induction on i that for any $1 \le i \le k$: $\overline{\mathbf{m}}_i = \mathbf{m}_i$. Hence $\overline{\sigma}_1 \cdots \overline{\sigma}_k$ is firable in $\mathcal{N}_2$ and leads to $\overline{\mathbf{m}}_k = \mathbf{m}_k = \langle 1, 0, 0, 0, \ell \rangle$ with $\ell = \sum_{i=1}^{k}(m_i - n_i)$. From that point, it is easy to see that the sequence $t_5 t_6^{\ell}$ is firable and leads to $\mathbf{m}_a$. Hence, for any sequence of σ of $\mathcal{N}_1$ that lead to $\mathbf{m}_a$, the sequence $\overline{\sigma}$ of $\mathcal{N}_2$ leads to $\mathbf{m}_a$ with $\Lambda(\sigma) = \Lambda(\overline{\sigma})$. Thus, $L(\mathcal{N}_2, \{\mathbf{m}_a\}) \supseteq L(\mathcal{N}_1, \{\mathbf{m}_a\})$.
$L(\mathcal{N}_2, \{\mathbf{m}_a\}) \subseteq L(\mathcal{N}_1, \{\mathbf{m}_a\})$: The other direction can be proved by similar arguments and is thus omitted.

□

4 A new conjecture

Thus, $\mathcal{N}_1$ is not suitable to strictly separate the expressive powers of ε-free $\mathsf{PN{+}R}$ and ε-free trans. Nevertheless, we suggest $\mathcal{N}_3$ (see Figure 2) with accepting set

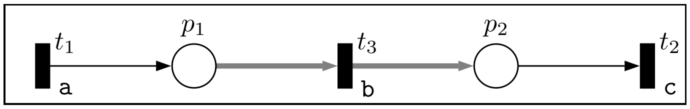


Figure 2: The PN+T $\mathcal{N}_3$: a new conjecture to separate the languages of PN+T and PN+R.

of markings $\{\langle 0,0\rangle\}$ as a new candidate of PN+T whose language cannot be accepted by a ε-free PN+R:

Conjecture 2 *There is no ε-free* PN+R $\mathcal{N}$ *and no finite set of markings S s.t.* $L(\mathcal{N}, S) = L(\mathcal{N}_3, \{\langle 0,0\rangle\})$.

5 Conclusion

In this paper, we have discussed the conjecture of [1], stating no ε-free PN+R can accept the language of $\mathcal{N}_1$ (Fig. 1(a)), for accepting marking $\langle 1,0,0,0,0\rangle$. We have shown that this conjecture is flawed, by exhibiting $\mathcal{N}_2$ (Fig. 1(b)), a ε-free PN+R that accepts exactly this language. We have proposed a new ε-free PN+T whose language we conjecture cannot be accepted by an ε-free PN+R.

We have purposely restrained our discussion to the case where the accepting set is a *finite set* of accepting markings (the so-called L-type languages in [1, 8]). In [7], the authors consider accepting sets which are *upward-closed*, i.e., that respect the condition that if $\mathbf{m}$ is accepting, then any marking $\mathbf{m}'$ s.t. $\mathbf{m}(p) \leq \mathbf{m}'(p)$ for any p is also accepting. In that case too, no formal proof that PN+T are more expressive than ε-free PN+R exists (see [9]). We are however confident that the PN+T $\mathcal{N}_3$ we have introduced could be useful to devise such a proof.

References

[1] G. Ciardo, Petri nets with marking-dependent ar cardinality: Properties and analysis., in: R. Valette (Ed.), Application and Theory of Petri Nets, Vol. 815 of Lecture Notes in Computer Science, Springer, 1994, pp. 179–198.

[2] C. A. Petri, Kommunikation mit automaten, Ph.D. thesis, Technical University Darmstadt (1962).

[3] S. M. German, A. P. Sistla, Reasoning about Systems with Many Processes, Journal of ACM 39 (3) (1992) 675–735.

[4] G. Delzanno, Automatic verification of parameterized cache coherence protocols., in: E. A. Emerson, A. P. Sistla (Eds.), CAV, Vol. 1855 of Lecture Notes in Computer Science, Springer, 2000, pp. 53–68.

[5] G. Delzanno, J.-F. Raskin, L. Van Begin, Towards the automated verification of multithreaded java programs., in: J.-P. Katoen, P. Stevens (Eds.), TACAS, Vol. 2280 of Lecture Notes in Computer Science, Springer, 2002, pp. 173–187.

[6] M. Jantzen, Language theory of petri nets., in: W. Brauer, W. Reisig, G. Rozenberg (Eds.), Advances in Petri Nets, Vol. 254 of Lecture Notes in Computer Science, Springer, 1986, pp. 397–412.

[7] G. Geeraerts, J.-F. Raskin, L. Van Begin, Well-Structured Languages, Acta Informatica 44 (3-4) (2007) 249–288.

[8] J. L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice Hall, 1981.

[9] G. Geeraerts, Coverability and expressiveness properties of WSTS, Ph.D. thesis, Université Libre de Bruxelles (2007).