On Replacement of Petri Nets and Some Applications[1]

Ferucio Laurenţiu ŢIPLEA [2]
Faculty of Informatics
"Al. I. Cuza" University, 6600 Iaşi, Romania
e-mail: `fltiplea@infoiasi.ro`

Masashi KATSURA, Masami ITO
Faculty of Science
Kyoto Sangyo University, 603 Kyoto, Japan
e-mail: `katsura,ito@ksuvx0.kyoto-su.ac.jp`

Abstract

Refinement and *abstraction* are complementary techniques in system design and analysis; both of them are generally referred to as *replacements*. In this paper we consider a general enough technique of replacement of Petri nets. It may be simply described by "replace the subnet γ_1 of γ by the net γ_2; denote the result by $\gamma[\gamma_1 \leftarrow \gamma_2]$". The subnet γ_1 will always be generated by a subset of transitions of γ, and this is the only restriction imposed in our replacement operation. Moreover, this operation is expressed in terms of *difference* and *catenation* of nets. A commutative monoid of nets with respect to catenation and two congruences, $\approx_{mP}$ and $\approx_{mPW}$, on this monoid are given. It is shown that from $\gamma_1 \approx_{mP} \gamma_2$ ($\gamma_1 \approx_{mPW} \gamma_2$, resp.) one can infer $\gamma \approx_P \gamma[\gamma_1 \leftarrow \gamma_2]$ ($\gamma \approx_{PW} \gamma[\gamma_1 \leftarrow \gamma_2]$, resp.), where $\approx_P$ and $\approx_{PW}$ are the *process* and respectively *partial word equivalence* on nets. A particular case when the equivalences $\approx_{mP}$ and $\approx_{mPW}$ may be easier decided is highlighted. Some applications on normalization of Petri nets are then presented. They consist in establishing the correctness of four transformations ([10], [19]) in a much simpler way.

1 Introduction and Preliminaries

Many transformations of Petri nets may be simply described by "replace the subnet Σ_1 of Σ by the net Σ_2"; that means the subnet Σ_1 will be removed from Σ and the net Σ_2 is inserted in its place. Clearly, some fitting conditions must be satisfied and if one wants to preserve some properties of the original net, some more conditions are needed. If Σ_2 is more "detailed" than Σ_1 this operation is called *refinement*, otherwise it is an *abstraction*; both of them are particular cases of *replacement*. In literature mostly refinement was studied and various techniques was proposed ([1], [2], [5], [6], [8], [9], [16], [20], [23], [24]). A large number of behaviour and equivalence notions preserving refinements have also been considered.

[1]a complete final version of this paper will be published elsewhere

[2]the present correspondence address: Faculty of Science, Kyoto Sangyo University, 603 Kyoto, Japan, e-mail: `tiplea@ksuvx0.kyoto-su.ac.jp`

In this paper we propose a technique of replacement preserving the processes and partial words of the original net (Section 2). Using this technique we give shorter and elegant proofs to the correctness of some transformations of Petri nets (Section 3). In the remainder of this section we recall the basic definitions and notations in Petri net theory (for further details the reader is referred to [3], [4], [12], [13], [14]). The empty set is denoted by $\emptyset$, and $|A|$ denotes the cardinality of the finite set A. $A \subseteq B$ denotes the inclusion of the set A into the set B. The set of nonnegative integers is denoted by $\mathbf{N}$. If $f_i : A_i \longrightarrow \mathbf{N}$ are functions, $i = 1, 2$, $f_1 + f_2$ is the function from $A_1 \cup A_2$ into $\mathbf{N}$ given by $(f_1 + f_2)(a) = f_1(a)$ for all $a \in A_1 - A_2$, $(f_1 + f_2)(a) = f_2(a)$ for all $a \in A_2 - A_1$, and $(f_1 + f_2)(a) = f_1(a) + f_2(a)$ for all $a \in A_1 \cap A_2$. The restriction of a function $f : A \longrightarrow B$ to the set $C \subseteq A$ is denoted by $f|_C$; f^{-1} is the function from B into the powerset of A given by $f^{-1}(b) = \{a \in A | f(a) = b\}$ for all $b \in B$.

A (finite) *P/T-net*, abbreviated *PTN*, is a 4-tuple $\Sigma = (S, T, F, W)$ where S and T are two finite non-empty sets (of *places* and *transitions*, resp.), $S \cap T = \emptyset$, $F \subseteq (S \times T) \cup (T \times S)$ is the *flow relation* and $W : (S \times T) \cup (T \times S) \longrightarrow \mathbf{N}$ is the *weight function* of Σ verifying $W(x, y) = 0$ iff $(x, y) \notin F$ (we suppose that all our nets do not have isolated transitions). A *marking* of Σ is any function $M : S \longrightarrow \mathbf{N}$; it will sometimes be identified with a vector $M \in \mathbf{N}^{|S|}$. The operations and relations on vectors are componentwise defined. For $x \in S \cup T$ we set ${}^\bullet x = \{y | (y, x) \in F\}$, $x^\bullet = \{y | (x, y) \in F\}$, ${}^\bullet x^\bullet = {}^\bullet x \cup x^\bullet$, and usually extend these notations to subsets $X \subseteq S \cup T$.

A *marked PTN*, abbreviated *mPTN*, is a pair $\gamma = (\Sigma, M_0)$, where Σ is a *PTN* and M_0, the *initial marking* of γ, is a marking of Σ. A *λ-labelled mPTN*, abbreviated *l^λmPTN*, is a 3-tuple $\gamma = (\Sigma, M_0, l)$, where the first two components form an *mPTN* and l, the *λ-labelling function* of γ, assigns to each transition either a letter or the empty word λ. When λ is not in the range of l we will refer to such triples as *labelled mPTN* and to l as *labelling function*; the notation will be modified correspondingly to *lmPTN*.

In the sequel we shall often use the term "Petri net" or "net" whenever we refer to a structure γ as defined above. In all the above definitions Σ is called the *underlying net* of γ. A marking (place, transition, arc, weight, resp.) of a net γ is any marking (place, transition, arc, weight, resp.) of the underlying net of γ. Pictorially, a net γ is represented by a graph. Then the places are denoted by circles and transitions by boxes; the flow relation is represented by arcs. The arc $f \in F$ is labelled by $W(f)$ whenever $W(f) > 1$. The initial marking M_0 is presented by putting $M_0(s)$ tokens into the circle representing the place s and the labelling function is denoted by placing letters into the boxes representing transitions.

Let γ be a net and M a marking of it. The *transition rule* states that a transition t is *enabled* at M, denoted $M[t\rangle_\gamma$, if $M(s) \geq W(s, t)$ for all $s \in S$. If t is enabled at M then t may *occur* yielding a new marking M', abbreviated $M[t\rangle_\gamma M'$, given by $M'(s) = M(s) - W(s, t) + W(t, s)$ for all $s \in S$. The transition rule is extended to sequences of transition $w \in T^*$ in the usual way. If $M_0[w\rangle_\gamma M$ then M is called *reachable*; $[M_0\rangle_\gamma$ denotes the set of all reachable markings. The notation "$[\cdot\rangle_\gamma$" will be simplified to "$[\cdot\rangle$" whenever γ is understood from context.

The concurrent behaviour of Petri nets is well-expressed by the notion of a *process*. Generally speaking, processes of Petri nets are obtained by running the nets and solving conflicts in an arbitrary fashion as and when they arise. A process of a net is also a net; these nets are called *occurrence nets* and they are classical nets $N = (B, E, R)$ (R is the flow relation and the weights of the arcs are 1) satisfying:

(i) $|^\bullet b| \leq 1$ and $|b^\bullet| \leq 1$, for all $b \in B$;

(ii) R^+ is acyclic, i.e. for all $x, y \in B \cup E$ if $(x, y) \in R^+$ then $(y, x) \notin R^+$.

Usually the elements of B are called *conditions* whereas the elements of E are called *events.* The *partially ordered set induced* by N is $(B \cup E, \prec_N)$, where $\prec_N = R^+$. A *B-cut* of N is any maximal subset $C \subseteq B$ of incomparable elements according with the relation $\prec_N$. As we will only use B-cuts we shortly call them *cuts* (see [4] for more details). The *initial* (*final*, resp.) *cut* of N is $^\circ N = \{b \in B \| ^\bullet b| = 0\}$ ($N^\circ = \{b \in B\||b^\bullet| = 0\}$, resp.). In defining processes we will use *labelled occurrence nets* which are couples $\pi = (N, p)$, where N is a occurrence net and p is a total function from $B \cup E$ into an alphabet V. The above definitions (partial order, cut, initial and final cut) are transferred to labelled occurrence nets π by means of N; the notations are obtained by changing "N" into "π" (e.g. $\prec_\pi$, $^\circ\pi$, π°). Let $\Sigma = (S, T, F, W)$ be a Petri net, $\pi = (N, p)$ a labelled occurrence net such that p is a function from $B \cup E$ into $S \cup T$ satisfying $p(B) \subseteq S$ and $p(E) \subseteq T$, and C a subset of conditions of π. Define the *marking induced by C in* Σ as being $M_C(s) = |p^{-1}(s) \cap C|$, for all $s \in S$. There are two alternative definitions of a process, axiomatic and inductive, and it is well-known that for Petri nets of finite synchronization they yields exactly the same objects ([4]). We adopt here the axiomatic definition. A *process* of $\gamma = (\Sigma, M_0)$ is any couple $\pi = (N, \varphi)$, where N is an occurrence net and $\varphi : B \cup E \longrightarrow S \cup T$ is a mapping satisfying:

(i) $\varphi(B) \subseteq S$, $\varphi(E) \subseteq T$;

(ii) $M_0(s) = |\varphi^{-1}(s) \cap {}^\circ N|$ for all $s \in S$;

(iii) $W(s, \varphi(e)) = |\varphi^{-1}(s) \cap {}^\bullet e|$ and $W(\varphi(e), s) = |\varphi^{-1}(s) \cap e^\bullet|$ for all $e \in E$ and $s \in S$

(as we can see we allow events without pre-conditions or post-conditions). In order to obtain processes of λ-labelled nets $\gamma = (\Sigma, M_0, l)$ what we have to do is to consider each proces $\pi = (N, p)$ of (Σ, M_0) and to replace the function p by p', where $p'(x) = p(x)$ for all the conditions x, and $p'(x) = (l \circ p)(x)$ for all the events x; that is, the events will be labelled by $l \circ p$; we will sometimes use $l \circ p$ instead of p' with the above meaning. Finally, to compare processes we consider the notions of *isomorphism of processes* and *partial word.* Let $\pi_1 = (N_1, p_1)$ and $\pi_2 = (N_2, p_2)$ be processes (not necessary of the same net). π_1 and π_2 are said *isomorphic*, abbreviated $\pi_1 \cong \pi_2$, if there is a bijection $\varphi : B_1 \cup E_1 \longrightarrow B_2 \cup E_2$ such that

(1) $p_1(x) = p_2(\varphi(x))$ for all $x \in E_1$, and
$p_1(x) = p_2(\varphi(x))$ whenever $x \in B_1$ and $p_1(x) \in S_1 \cap S_2$ or $p_2(\varphi(x)) \in S_1 \cap S_2$;

(2) $x \prec_{\pi_1} y$ iff $\varphi(x) \prec_{\pi_2} \varphi(y)$ for all $x, y \in B_1 \cup E_1$.

(1) says that φ is a label-preserving bijection between $E_1 \cup p_1^{-1}(S_1 \cap S_2)$ and $E_2 \cup p_2^{-1}(S_1 \cap S_2)$, while (2) says that φ is order preserving in both directions.

In order to define the partial word associated to a process we have to derive from processes another structure by recording only the events which are not labelled by λ. Let $\pi = (N, p)$ be a process of a net γ. An *abstraction* of π is any labelled partially ordered set (E', A, p'), where:

- $E' = \{e \in E | p(e) \neq \lambda\}$ and $p' = p|_{E'}$;

– $(e, e') \in A$ iff there is a path in π leading from e to e'.

The equivalence class with respect to isomorphism induced by (E', A, p'), denoted $PW(\pi)$, is called the *partial word* associated to π. The set of all partial words of γ is denoted by $PW(\gamma)$.

Let γ_1 and γ_2 be two nets. We say that γ_1 and γ_2 are *process equivalent*, abbreviated $\gamma_1 \approx_P \gamma_2$, if for any process π_1 of γ_1 there is a process π_2 of γ_2 such that $\pi_1 \cong \pi_2$, and vice versa. We say that γ_1 and γ_2 are *partial word equivalent*, abbreviated $\gamma_1 \approx_{PW} \gamma_2$, if $PW(\gamma_1) = PW(\gamma_2)$. Thus we have obtained two equivalence relations on nets: $\approx_P$ (process equivalence) and $\approx_{PW}$ (partial word equivalence).

2 Replacements of Petri Nets

The operation that we will consider in this section has its roots in various transformations of Petri nets. These transformations may be simply described by "replace the subnet Σ_1 of Σ by the net Σ_2", that means the subnet Σ_1 will be removed from Σ and the net Σ_2 is inserted in its place. Clearly, some fitting conditions must be satisfied and if one wants to preserve the behaviour of the original net, some more conditions are needed, and this is what we want to discuss in the next two subsections.

2.1 Subnets, Connecting Places, and Replacements

In our paper the subnets that are considered will always be generated by subsets of transitions. That is, if $\Sigma = (S, T, F, W)$ is a net and $T_1 \subseteq T$ then the *subnet generated by* T_1 is $\Sigma_1 = (S_1, T_1, F_1, W_1)$, where $S_1 = {}^\bullet T_1^\bullet$ and F_1 and W_1 are the restrictions of F and W to S_1 and T_1. Let Σ_1 be such a subnet of Σ. The set of places S_1 can be partitioned in two disjoint subsets (not necessary both of them non-empty) $S_1 = S_1^c \cup S_1^i$, where

$$S_1^c = \{s \in S | \exists t \in T - T_1,\ \exists t_1 \in T_1 :\ s \in {}^\bullet t^\bullet \cap {}^\bullet t_1^\bullet\}.$$

We will call the places in S_1^c (S_1^i, resp.) *connecting* (*internal*, resp.) *places* of Σ_1. The terminology is clear, the connecting places assure the connection of Σ_1 to Σ and the internal places are specific only for Σ_1, that is no transition in $T - T_1$ is connected to any place in S_1^i.

Definition 2.1.1 Let Σ be a net and Σ_1 a subnet of Σ. We say that Σ' is the *difference of* Σ *and* Σ_1, abbreviated $\Sigma' = \Sigma - \Sigma_1$, if $S' = S - S_1^i$, $T' = T - T_1$ and the flow relation and the weight function of Σ' are the corresponding restrictions of F and W to S' and T'.

Specifying Σ_1 the set S_1^c is implicitly specified. Our intention is to replace Σ_1 by a new net $\Sigma_2 = (S_2, T_2, F_2, W_2)$ (suppose $(S \cup T) \cap (S_2 \cup T_2) = \emptyset$). To do that we have first to specify a set of connecting places of Σ_2, S_2^c, and an one-to-one correspondence between S_1^c and S_2^c. Therefore we consider an one-to-one mapping h from S_1^c into S_2 and denote $S_2^c = h(S_1^c)$ and $S_2^i = S_2 - S_2^c$. This one-to-one mapping says us how to connect Σ_2 to $\Sigma - \Sigma_1$. Now the replacement of Σ_1 by Σ_2 can be described as follows: connect the nets $\Sigma - \Sigma_1$ and Σ_2 by identifying the places $s \in S_1^c$ and $h(s) \in S_2$; the resulting net is denoted $\Sigma[\Sigma_1 \overset{h}{\leftarrow} \Sigma_2]$.

Definition 2.1.2 Let Σ_1 and Σ_2 be nets, $S_1^c \subseteq S_1$, and h an one-to-one mapping from S_1^c into S_2 (suppose $(S_1 \cup T_1) \cap (S_2 \cup T_2) = \emptyset$). We say that the net Σ' is obtained by *h-catenation of* Σ_1 *and* Σ_2, abbreviated $\Sigma' = \Sigma_1 \circ_h \Sigma_2$, if the following properties hold:

- $S' = S_1 \cup (S_2 - h(S_1^c))$;
- $T' = T_1 \cup T_2$;
- $(x,y) \in F'$ iff $((x,y) \in F_1 \ \wedge \ W'(x,y) = W_1(x,y))$ or $((h(x),y) \in F_2 \ \wedge \ W'(x,y) = W_2(h(x),y))$ or $((x,h(y)) \in F_2 \ \wedge \ W'(x,y) = W_2(x,h(y)))$ or $((x,y) \in F_2 \ \wedge \ W'(x,y) = W_2(x,y))$
  (this defines both F' and W').

We allow the case $S_1^c = \emptyset$ as well: the net Σ' consists of two disjoint subnets, Σ_1 and Σ_2.

Definition 2.1.3 Let Σ and Σ_2 be nets, Σ_1 a subnet of Σ and h an one-to-one mapping from S_1^c into S_2 (suppose $(S \cup T) \cap (S_2 \cup T_2) = \emptyset$). We say that the net Σ' is obtained by *h-replacement of* Σ_1 *by* Σ_2 *in* Σ, abbreviated $\Sigma' = \Sigma[\Sigma_1 \overset{h}{\leftarrow} \Sigma_2]$, if $\Sigma' = (\Sigma - \Sigma_1) \circ_h \Sigma_2$.

The above definitions are naturally extended to nets $\gamma = (\Sigma, M_0, l)$. For instance, the difference $\gamma - \gamma_1$ is simply defined by restricting the components of γ to $\Sigma - \Sigma_1$ (γ_1 is a subnet of γ). If γ_1 and γ_2 are nets such that $M_0^1(s) = M_0^2(h(s))$ for all $s \in S_1^c$ ($S_1^c \subseteq S_1$ and h is an one-to-one mapping from S_1^c into S_2) then the h-catenation of γ_1 and γ_2 is the net γ' given by:

- $\Sigma' = \Sigma_1 \circ_h \Sigma_2$;
- $M_0' = M_0^1|_{S_1^i} + M_0^1|_{S_1^c} + M_0^2|_{S_2^i}$;
- $l' = l_1 \cup l_2$.

The h-catenation of processes (labelled occurrence nets) needs a supplementary condition: the function h should preserve the labels of places, that is $p_1(s) = p_2(h(s))$ for all $s \in S_1^c$ (p_1 and p_2 are the corresponding labelling functions). Now, the extension of the h-replacement to γ_1 by γ_2 follows from the above extensions of the difference and h-catenation.

Example 2.1.1 In Figure 2.1.1(a) is picturially represented a net γ and a subnet γ_1 of it; the net $\gamma - \gamma_1$ is given in Figure 2.1.1(b). Consider the net γ_2 in Figure 2.1.1(c) and the mapping $h(s_i) = s_i'$, $i = 2,3$. The net $\gamma[\gamma_1 \overset{h}{\leftarrow} \gamma_2]$ is given in Figure 2.1.1(d).

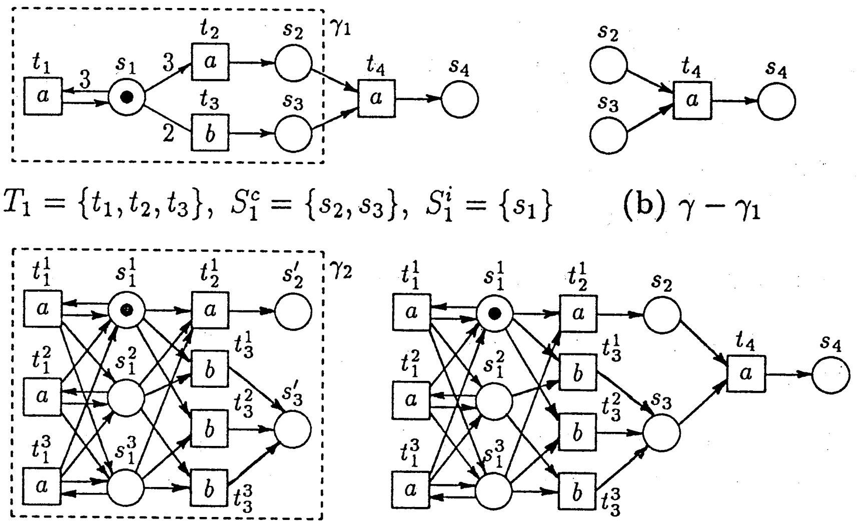
As we have seen the replacement operation on $\gamma = (\Sigma, M_0, l)$ requires the specification of a subnet Σ_1 of Σ and an one-to-one mapping h. The subnet Σ_1 will be the subnet generated by a non-empty set of transitions $T_1 \subset T$, and therefore to specify Σ_1 is to specify T_1. The mapping h will be specified considering $S_1^c = S_2^c$ (h is the identity, $S \cap S_2 = S_1^c$, and $T \cap T_2 = \emptyset$). Then we will simply write $\gamma[\gamma_1 \leftarrow \gamma_2]$ ($\circ$, resp.) instead of $\gamma[\gamma_1 \overset{h}{\leftarrow} \gamma_2]$ ($\circ_h$, resp.). Let $PN(S^c)$ denote the class of Petri nets having in common only the set S^c of places and

$$PN(S^c, M_0^c) = \{\gamma \in PN(S^c) | M_0|_{S^c} = M_0^c\}$$

($M_0^c \in \mathbf{N}^{S^c}$). Then, we have:

Proposition 2.1 *$(PN(S^c, M_0^c), \circ)$ is a commutative monoid.*

Proof The associativity and commutativity directly follow from definitions; the net $((S^c, \emptyset, \emptyset, \emptyset), M_0^c, \emptyset)$ is the unity of this monoid. □



(a) $T_1 = \{t_1, t_2, t_3\}$, $S_1^c = \{s_2, s_3\}$, $S_1^i = \{s_1\}$ (b) $\gamma - \gamma_1$

(c) $S_2^c = \{s_2', s_3'\}$ (d) $\gamma[\gamma_1 \overset{h}{\leftarrow} \gamma_2]$

Figure 2.1.1

2.2 Concurrent Behaviour and Replacements

Our interest is to find some equivalence relations on nets, $\approx$, such that from $\gamma_1 \approx \gamma_2$ one can infer $\gamma \approx_P \gamma[\gamma_1 \leftarrow \gamma_2]$ or $\gamma \approx_{PW} \gamma[\gamma_1 \leftarrow \gamma_2]$. In what follows we will give two such equivalence relations; as we shall see they are very close to the relations $\approx_P$ and $\approx_{PW}$, the difference consisting in the interaction with the environment.

Let γ be a net, $S^c \subseteq S$, and π a process of γ. An *S^c-abstraction* of π is any labelled partially ordered set $(B' \cup E', A, p')$, where:

- $B' = \{b \in B | p(b) \in S^c \ \wedge \ (|{}^\bullet b| = 0 \ \vee \ |b^\bullet| = 0)\}$, $E' = \{e \in E | p(e) \neq \lambda\}$, and $p' = p|_{B' \cup E'}$;
- $(x, y) \in A$ iff there is a path in π leading from x to y.

The equivalence class with respect to isomorphism induced by $(B' \cup E', A, p')$, denoted S^c-$PW(\pi)$, is called the *S^c-partial word* associated to π. The class of all S^c-partial words of γ is denoted by S^c-$PW(\gamma)$.

Let $\gamma \in PN(S^c, M_0^c)$ and $M \in \mathbf{N}^{S^c}$. By $(\gamma + M)$ we denote the net $(\Sigma, M_0 + M, l) \in PN(S^c, M_0^c + M)$. For any $\gamma_1, \gamma_2 \in PN(S^c, M_0^c)$ and $M \in \mathbf{N}^{S^c}$ we have $(\gamma_1 \circ \gamma_2 + M) = (\gamma_1 + M) \circ (\gamma_2 + M)$.

Definition 2.2.1 Let $\gamma_1, \gamma_2 \in PN(S^c, M_0^c)$. We say that γ_1 and γ_2 are *m-process equivalent* (*m-partial word equivalent*, resp.), abbreviated $\gamma_1 \approx_{mP} \gamma_2$ ($\gamma_1 \approx_{mPW} \gamma_2$, resp.), if for any $M \in \mathbf{N}^{S^c}$ we have $(\gamma_1 + M) \approx_P (\gamma_2 + M)$ (S^c-$PW(\gamma_1 + M) = S^c$-$PW(\gamma_2 + M)$, resp.).

This definition says that no matter how the initial marking on S^c is increased that two nets have the same processes (up to an isomorphism) or the same S^c-partial words. It is easy to see that $\approx_{mP}$ and $\approx_{mPW}$ are equivalence relations on nets (in order to simplify the writing we will use, in what follows, the next notations and conventions: $\approx_m$ denotes one of the relations $\approx_{mP}$ or $\approx_{mPW}$, while $\approx$ denotes $\approx_P$ or $\approx_{PW}$. Moreover, when we use both $\approx_m$ and $\approx$ and $\approx_m$ denotes $\approx_{mP}$ then $\approx$ will denote $\approx_P$, and similar for the other case). Let $\overset{M}{\approx}_m$ be the binary relation

$$\gamma_1 \overset{M}{\approx}_m \gamma_2 \quad \text{iff} \quad (\gamma_1 + M) \approx_m (\gamma_2 + M),$$

where $M \in \mathbf{N}^{S^c}$. That is, $\gamma_1 \overset{M}{\approx}_m \gamma_2$ iff $(\gamma_1 + M') \approx (\gamma_2 + M')$ for all $M' \geq M$ ($M' \in \mathbf{N}^{S^c}$). We have:

Proposition 2.2.1 *(1)* $\approx_m \subseteq \approx$;
(2) $\approx_m \subseteq \overset{M}{\approx}_m \subseteq \overset{M'}{\approx}_m$ *for all* $M, M' \in \mathbf{N}^{S^c}$ *with* $M \leq M'$.

The next theorem represents the main result of this section.

Theorem 2.2.1 *Let* $\gamma_1, \gamma_2 \in PN(S^c, M_0^c)$. *If* $\gamma_1 \approx_m \gamma_2$ *then* $\gamma_0 \circ \gamma_1 \approx \gamma_0 \circ \gamma_2$ *for all* $\gamma_0 \in PN(S^c, M_0^c)$.

Corollary 2.2.1 *Let* $\gamma_1, \gamma_2 \in PN(S^c, M_0^c)$. *If* $\gamma_1 \approx_m \gamma_2$ *then* $\gamma_0 \circ \gamma_1 \approx_m \gamma_0 \circ \gamma_1$ *for all* $\gamma_0 \in PN(S^c, M_0^c)$. *Therefore,* $\approx_{mP}$ *and* $\approx_{mPW}$ *are congruences on* $(PN(S^c, M_0^c), \circ)$.

Corollary 2.2.2 *Let* γ_1 *be a subnet of* γ *generated by a subset* T_1 *of transitions, and* $\gamma_2 \in PN(S^c, M_0^c)$, *where* $M_0^c = M_0|_{S_1^c}$. *If* $\gamma_1 \approx_m \gamma_2$ *then* $\gamma \approx \gamma[\gamma_1 \leftarrow \gamma_2]$.

From practical point of view we are interested in Corollary 2.2.2. The difficulty in using this corollary consists in the fact that we have to decide whether or not $\gamma_1 \approx_m \gamma_2$, that is we have to decide whether or not $(\gamma_1 + M) \approx (\gamma_2 + M)$ for all $M \in \mathbf{N}^{S^c}$. A favourable particular case would be when there is a marking $M \in \mathbf{N}^{S^c}$ such that the processes of nets $(\gamma + M')$, where $M' \in \mathbf{N}^{S^c}$ and $\neg(M' \leq M)$, can be reduced to processes of $(\gamma + M)$. Let us introduce first a few notations. Let $\pi = (N, p)$ be a labelled occurrence net and A a subset of the range of p. If $C \subseteq p^{-1}(A) \cap {}^\circ\pi^\circ$, where ${}^\circ\pi^\circ = {}^\circ\pi \cap \pi^\circ$, then we will denote by $(\pi - C)$ the labelled occurrence net obtained from π by removing all the conditions in C.

Definition 2.2.2 Let $\gamma \in PN(S^c, M_0^c)$, $M_1, M_2 \in \mathbf{N}^{S^c}$, $\pi_1 \in \Pi(\gamma + M_1)$, and $\pi_2 \in \Pi(\gamma + M_2)$. We say that π_1 and π_2 are *almost isomorphic*, abbreviated $\pi_1 \cong_a \pi_2$, if there are $C_1 \subseteq p_1^{-1}(S^c) \cap {}^\circ\pi_1^\circ$ and $C_2 \subseteq p_2^{-1}(S^c) \cap {}^\circ\pi_2^\circ$ such that $(\pi_1 - C_1) \cong (\pi_2 - C_2)$.

This definition wants to say that if we remove from π_1 and π_2 some conditions without predecessors and succesors and labelled by places in S^c we get two isomorphic labelled occurrence nets.

Lemma 2.2.1 *Let* $\gamma \in PN(S^c, M_0^c)$, $M_1, M_2 \in \mathbf{N}^{S^c}$, $\pi_1 \in \Pi(\gamma + M_1)$, *and* $\pi_2 \in \Pi(\gamma + M_2)$. *Then,* $\pi_1 \cong_a \pi_2$ *iff* $(\pi_1 - (p_1^{-1}(S^c) \cap {}^\circ\pi_1^\circ)) \cong (\pi_2 - (p_2^{-1}(S^c) \cap {}^\circ\pi_2^\circ))$.

Definition 2.2.3 Let $\gamma \in PN(S^c, M_0^c)$ and $M \in \mathbf{N}^{S^c}$. We say that γ is *process stable w.r.t.* M if for any marking $M' \in \mathbf{N}^{S^c}$ with $\neg(M' \leq M)$ we have:

- for any process π' of $(\gamma + M')$ there is a process π of $(\gamma + M)$ such that $\pi \cong_a \pi'$;
- vice versa.

As an example, the nets in Figure 3.2(a)(b) are process stable w.r.t. $M = (0, \ldots, 0)$. We have:

Proposition 2.2.2 *Let* $\gamma_1, \gamma_2 \in PN(S^c, M_0^c)$ *be two process stable nets w.r.t.* M. *If* $(\gamma_1 + M) \approx (\gamma_2 + M)$ *then* $\gamma_1 \approx_m \gamma_2$.

3 Applications

The Corollary and Proposition 2.2.2 may be used to prove the correctness of some transformations of Petri nets. Their efficiency directly depends on the easiness of deciding the equivalences $\approx_{mP}$ and $\approx_{mPW}$. Intuitively, the simpler are γ_1 and γ_2 the easier we can test $\gamma_1 \approx_{mP} \gamma_2$ and $\gamma_1 \approx_{mPW} \gamma_2$ and therefore the more efficient we can apply Corollary 2.2.2 and Proposition 2.2.2. In what follows will exemplify our discussion by some applications. First we recal the inductive definition of processes in terms of catenation of nets. Let $\gamma = (\Sigma, M_0)$ be a net and t a transition of it. An *elementary occurrence net* associated to t is any labelled occurrence net $\pi = (N, p)$ with the properties: π contains only an event e labelled by t, $W(s,t)$ preconditions and $W(t,s)$ postconditions of e labelled by s, for all $s \in S$ (π does not contain other elements than those already specified). An *initial occurrence net* of γ is any occurrence net (N,p) which does not contain any event and for each $s \in S$ it contains exactly $M_0(s)$ conditions labelled by s (it does not contain other elements than those already specified). Define $\Pi(\gamma)$ as being the smallest set with the properties:

(1) $\Pi(\gamma)$ contains all the initial occurrence nets associated to γ;
(2) if $\pi_1 \in \Pi(\gamma)$ and π_2 is an elementary occurrence net such that π_1 and π_2 are disjoint, and $h : B_1' \longrightarrow {}^\circ\pi_2$ is a bijection ($B_1' \subseteq B_1$) such that $p_1(b) = p_2(h(b))$ for all $b \in B_1'$, then $\pi_1 \circ_h \pi_2 \in \Pi(\gamma)$.

Processes of λ-labelled nets may be defined as in the first section.

According to [10], a labelled marked Petri net is called *normalized* if the weight function and the initial marking take values into $\{0,1\}$. E. Pelz ([10]) showed that any Petri net is process equivalent with a normalized one. Moreover an algorithm to transform a Petri net into an equivalent normalized one was given. The Pelz's algorithm works in two main steps, denoted **Transformation-A** and **Transformation-B**. In the first one the weight function, and in the second the initial marking, is processed. Also the initial marking is needed to be processed in the first step. The Pelz's solution for processing the initial marking was to add new places and transitions in order to "simulate" it. This fact led to an increasing almost double of the complexity of construction. In [18] another solution was proposed. It consists of a distribution of the initial marking in the old places. No place and transition is needed more. The complexity of **Transformation-A** is to the half reduced. Therefore, we will describe the Pelz's algorithm as in [18]. First, a few notations and definitions are nedeed.

Let $\gamma = (\Sigma, M_0, l)$ be a net, $S_1 \subseteq S$, and M a marking of γ. We say that M is *uniformly distributed over* S_1 if $|M(s_1) - M(s_2)| \leq 1$ for all $s_1, s_2 \in S_1$. Let $s \in S$ and γ' be a net obtained from γ replacing the place s by a set of new places, S', and connecting the places of S' to transitions of γ in an arbitrary but fixed way. Let M and M' be markings of γ and respectively γ'. We say that M and M' are *s-compatible* if $M'(x) = M(x)$ for all $x \in S - \{s\}$, and $M(s) = \sum_{s' \in S'} M'(s')$.

Now, using the replacement operation we can describe the Pelz's algorithm as follows.

Transformation-A: Let $\gamma = (\Sigma, M_0, l)$ be a net and $n_s = max\{W(s,t), W(t,s) | t \in T\}$ for all $s \in S$. For each s with $n_s > 1$ replace the subnet γ_s generated by $T_s = \{t \in T | t \in {}^\bullet s^\bullet\}$, by the net γ_s' defined as follows:

- $\gamma'_s = (\Sigma'_s, M'_s, l'_s)$, $\Sigma'_s = (S'_s, T'_s, F'_s, W'_s)$;
- $S'_s = (S_s - \{s\}) \cup C(s)$, where S_s is the set of places of γ_s and $C(s) = \{s^1, \ldots, s^{n_s}\}$ is a set of n_s new places (copies of s);
- $T'_s := \cup_{t \in {}^\bullet s^\bullet} C(t)$, where for each $t \in {}^\bullet s^\bullet$, $C(t) = \{t_{A,B} | A, B \subseteq C(s) \wedge |A| = W(s,t) \wedge |B| = W(t,s)\}$ is a set of new transitions (copies of t);
- $F' := F_1 \cup F_2$, where:

$$\begin{aligned} F_1 &= \{(s', t_{A,B}) | t_{A,B} \in C({}^\bullet s^\bullet), (s', t) \in F\} \cup \\ &\quad \{(t_{A,B}, s') | t_{A,B} \in C({}^\bullet s^\bullet), (t, s') \in F\}, \\ F_2 &= \{(s', t_{A,B}) | t_{A,B} \in C({}^\bullet s^\bullet), s' \in A\} \cup \\ &\quad \{(t_{A,B}, s') | t_{A,B} \in C({}^\bullet s^\bullet), s' \in B\}, \end{aligned}$$

  and $C({}^\bullet s^\bullet)$ is the union-extension of $C(\cdot)$ to the set ${}^\bullet s^\bullet$;
- W'_s is given by:

$$\begin{aligned} W'_s(s', t_{A,B}) &= W(s', t) \text{ for all } (s', t_{A,B}) \in F_1, \\ W'_s(t_{A,B}, s') &= W(t, s') \text{ for all } (t_{A,B}, s') \in F_1, \\ W'_s(f) &= 1 \text{ for all } f \in F_2; \end{aligned}$$

- $M'_s|_{C(s)}$ is an arbitrary but fixed uniformly distributed marking over $C(s)$ and s-compatible with $M_0|_s$; $M'_s(s') := M_0(s')$ for all $s' \in S_s - \{s\}$;
- $l'_s(t_{A,B}) = l(t)$ for all $t_{A,B} \in T'_s$.

This transformation is exemplified in Example 2.1.1. It is clear that the arcs of the net yielded by it have the weight 0 or 1.

Theorem 3.1 ([10], [18])
The net γ' yielded by Transformation-A on the input γ satisfies $\gamma \approx_P \gamma'$.

Proof In the view of Corollary 2.2.2 we have to prove that $\gamma_s \approx_{mP} \gamma'_s$, for all s with $n_s > 1$. To prove this it is enough to show that:

- any initial occurrence net of $(\gamma_s + M)$ is isomorphic with any initial occurrence net of $(\gamma'_s + M)$,
- any elementary occurrence net of $(\gamma_s + M)$ associated to a transition t is isomorphic with any elementary occurrence net of $(\gamma'_s + M)$ associated to any copy of t,

where M is an arbitrary marking on the connecting places. But these facts follows directly from the definition of γ'_s. □

Transformation-B: Let $\gamma = (\Sigma, M_0, l)$ be a net such that $W(f) = 1$ for all $f \in F$. Let $m_s = M_0(s)$ for all $s \in S$. For each s with $m_s > 1$ replace the subnet γ_s of γ generated by $T_s = \{t \in T | {}^\bullet s^\bullet\}$, by the net γ'_s defined as follows:

- $\gamma'_s = (\Sigma'_s, M'_s, l'_s)$, $\Sigma'_s = (S'_s, T'_s, F'_s, W'_s)$;
- $S'_s = (S_s - \{s\}) \cup C(s)$, where S_s is the set of places of γ_s and $C(s) = \{s^1, \ldots, s^{m_s}\}$ is a set of new places (copies of s);
- $T'_s := \cup_{t \in T_s} C(t)$, where for each $t \in T_s$, $C(t) = \{t^1, \ldots, t^{m_s}\}$ is a set of new transitions (copies of t);
- $F'_s := F_1 \cup F_2$, where:

$$\begin{aligned} F_1 &= \{(s', t^i) | t^i \in T'_s, s' \in S_s - \{s\}, (s', t) \in F\} \cup \\ &\quad \{(t^i, s') | t^i \in T'_s, s' \in S_s - \{s\}, (t, s') \in F\}, \\ F_2 &= \{(s^i, t^j) | t^j \in T'_s, s^i \in C(s), (s, t) \in F\} \cup \\ &\quad \{(t^j, s^i) | t^j \in T'_s, s^i \in C(s), (t, s) \in F\}; \end{aligned}$$

- $W'_s(f) = 1$ for all $f \in F'_s$;
- $M'_s(s^i) = 1$ for all $1 \leq i \leq m_s$; $M'_s(s') := M_0(s')$ for all $s' \in S_s - \{s\}$;
- $l'_s(t^i) = l(t)$ for all $t^i \in T'_s$.

This transformation is exemplified in Example 3.1 for the case of the place s_1. Clearly, the net yielded by Transformation-B is normalized.

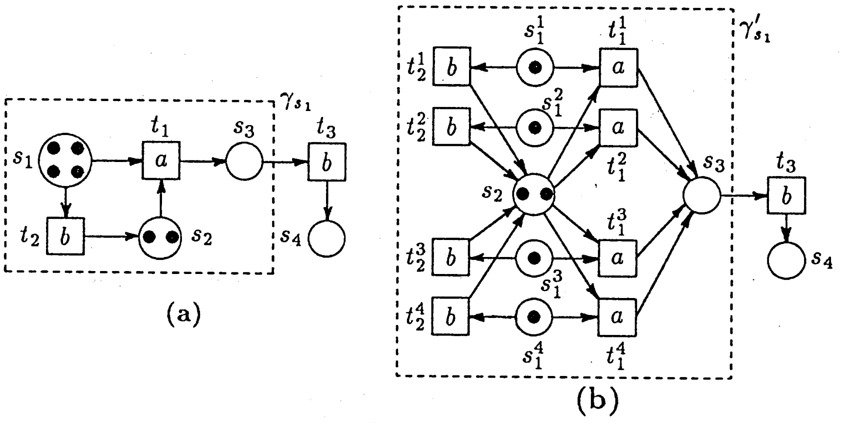


Figure 3.1

Theorem 3.2 ([10])
The net γ' yielded by Transformation-B on the input γ satisfies $\gamma \approx_P \gamma'$.

Proof A similar argument as in the proof of the above theorem works in this case too: any elementary occurrence net of γ_s whose event is labelled by a is isomorphic to any elementary occurrence net of γ'_s whose event is labelled by a too. □

The above two theorems assure the correctness of the Pelz's algorithm.

In [15] a systematic investigation of graph theoretic properties of Petri nets within the framework of language theory was initiated. In other words, various subclasses of Petri nets were introduced by imposing various restrictions on the in- and out- degree of nodes in the graph of the underlying net structure. Further these restrictions were refined in [17] by considering (n, m)*-transition restricted Petri nets* as being Petri nets for which the weight function takes values in $\{0, 1\}$ and $1 \leq |{}^\bullet t| \leq n$ and $1 \leq |t^\bullet| \leq m$ for all transitions t. Thus, interesting hierarchies of Petri net languages was obtained, and in the case of λ-labelled Petri nets, the above normal form was improved with respect to the finite transition sequence behaviour. More precisely, it was shown that every λ-labelled Petri net is equivalent to a $(2, 2)$-transition restricted net (with respect to the finite transition sequence behaviour). This result was extended in [19] by showing that this new normal form, called the *super-normal form*, preserves the partial words but not the processes of Petri nets. We will give here short proofs of these results. Let us recall first the basic transformations.

Let γ be a λ-labelled net. In the view of the Pelz's theorem we may assume that γ is normalized. Now we have to do two basic transformations on γ.

Transformation-C: Let γ be a normalized net. For each transition t of γ such that $|{}^\bullet t| = 0$ or $|t^\bullet| = 0$ replace the subnet γ_t (generated by t) by the net $\gamma'_t = (\Sigma'_t, M'_t, l'_t)$ as follows:

- if Σ_t is the net in Figure 3.2(a) then Σ'_t is the net in Figure 3.2(b);
- if Σ_t is the net in Figure 3.2(c) then Σ'_t is the net in Figure 3.2(d);
- the initial marking of γ'_t on the places $s_1, \ldots, s_k$ is the same as the initial marking of γ on these places;
- the labeling is that specified in diagrams.

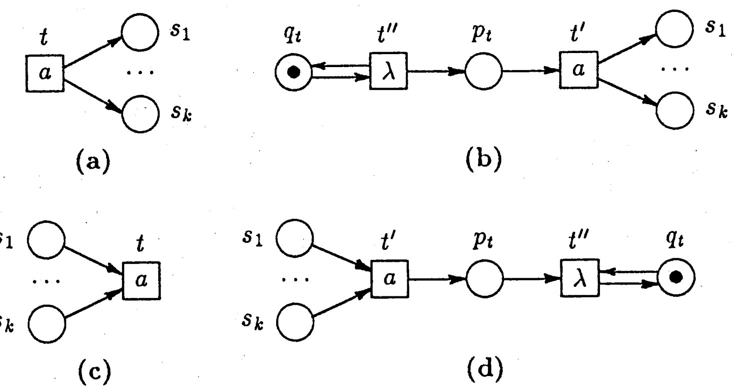


Figure 3.2

It is clear that the net γ' yielded by Transformation-C is normalized and satisfies $|{}^\bullet t| \geq 1$ and $|t^\bullet| \geq 1$ for all transitions t.

Theorem 3.3 ([19])
The net γ' yielded by Transformation-C on the input γ satisfies $\gamma \approx_{PW} \gamma'$.

Proof In the view of Corollary 2.2.2 we have to prove that $\gamma_t \approx_{mPW} \gamma'_t$ for all t with the property $|{}^\bullet t| = 0$ or $|t^\bullet| = 0$. But this job is as simple as minute it is, and therefore it is omitted (for the nets in Figure 3.2(a)(b) we may use Proposition 2.2.2). □

Transformation-D: Let γ be a normalized net γ satisfying $|{}^\bullet t| \geq 1$ and $|t^\bullet| \geq 1$ for all $t \in T$. For each transition $t \in T$ such that $|{}^\bullet t| \geq 3$ or $|t^\bullet| \geq 3$ replace the subnet Σ_t by the net Σ'_t as given in Figure 3.3, but with the next remarks:

- in the case $n = 1$ or $n = 2$ the places s_1 or s_1 and s_2 respectively are directly connected to t^n;
- in the case $m = 1$ or $m = 2$ the only successors of t^n are s_{n+1} or s_{n+1} and s_{n+2} respectively

(in Figure 3.3 it was assumed that ${}^\bullet t = \{s_1, \ldots, s_n\}$, $t^\bullet = \{s_{n+1}, \ldots, s_{n+m}\}$, $s'_1, \ldots, s'_{n+m-3}$ are new places, and $t^1, \ldots, t^{n+m-2}$ are new transitions. Moreover, it was assumed that ${}^\bullet t \cap t^\bullet = \emptyset$; the case ${}^\bullet t \cap t^\bullet \neq \emptyset$ case can be easily imagened).

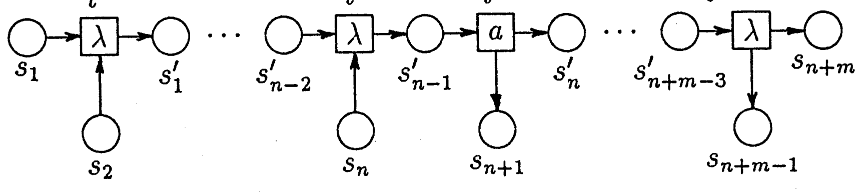


Figure 3.3

It is clear that the net γ' yielded by Transformation-D is normalized and $(2,2)$-transition restricted, and the proof of the next theorem can be easily completed.

Theorem 3.4 ([19])
The net γ' yielded by Transformation-D on the input γ satisfies $\gamma \approx_{PW} \gamma'$.

Conclusions

We have proposed a replacement technique of Petri nets and some congruences preserving the process and partial word equivalence between the original net and the net yielded by replacement. The "power" of these congruences have been proved by reporting them to different transformations of Petri nets known from literature.

References

[1] W. Brauer, R. Gold, W. Vogler: *A Survey of Behaviour and Equivalence Preserving Refinements of Petri Nets*, in: Advances of Petri Nets 1990, LNCS 483, 1990, 1–46.

[2] E. Best, R. Devillers, A. Kiehn, L. Pomelo: *Concurrent Bisimulations in Petri Nets*, Acta Informatica 28, 1991, 231-264.

[3] E. Best, C. Fernandez: *Notations and Terminology on Petri Net Theory*, Arbeitspapiere der GMD 195, 1986.

[4] E. Best, C. Fernandez: *Nonsequential Processes. A Petri Net Point of View*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1988.

[5] G. Chehaibar: *Replacement of Open Interface Subnets and Stable State Transformation Equivalence*, in: Advances in Petri Nets 1993, LNCS 674, 1993, 1–25.

[6] R.J.v. Glabbeck, U. Golz: *Equivalence Notions for Concurrent Systems and Action Refinement*, in: Mathematical Foundations of Computer Science 1989, LNCS 379, 1989, 237–248.

[7] M. Jantzen: *Language Theory of Petri Nets*, LNCS 254, Springer-Verlag, 1986.

[8] A. Kiehn: *Petri Net Systems and their Closure Properties*, in: Advances in Petri Nets 1989, LNCS 424, 1990, 306–328.

[9] K. Müller: *Constructable Petri Nets*, Journal of Information Processing and Cybernetics EIK 21, 1985, 171–199.

[10] E. Pelz: *Normalization of Place/Transition-Systems Preserves Net Behaviour*, in: Reseaux et logique. L'etude des semantique de la concurrence dans le cadre des reseaux de Petri (These d'etat), Universite de Paris-Sud, 1990 (also in: RAIRO Informatique Theorique 26, no.1, 1992, 19–44).

[11] E. Pelz: *Place/Transition-Systems. Concurrent Behaviour and Logic*, Report LRI 571, 1990.

[12] J.L. Peterson: *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, 1981.

[13] W. Reisig: *Petri Nets. An Introduction*, EATCS Monographs on Theoret. Comput. Sci., Springer-Verlag, 1985.

[14] W. Reisig: *Place Transition Systems*, LNCS 254, Springer-Verlag, Berlin, Heidelberg, 1986.

[15] G. Rozenberg, R. Verraedt: *Restricting the In-Out Structure of Graphs of Petri Nets. A Language Theoretic Point of View*, Fundamenta Informaticae VII.2, 1984.

[16] I. Suzuki, T. Murata: *A Method for Stepwise Refinement and Abstraction of Petri Nets*, J. Comput. Syst. Sci. 27, 1983, 51–76.

[17] F.L. Ţiplea, C. Ene: *Hierarchies of Petri Net Languages and a Super-Normal Form*, in: Developments in Languages Theory 1995, Magdeburg, Germany, 1995.

[18] F.L. Ţiplea: *On Normalization of Petri Nets*, submitted.

[19] F.L. Ţiplea, M. Katsura, M. Ito: *On a Normal Form of Petri Nets*, submitted.

[20] R. Valette: *Analysis of Petri Nets by Stepwise Refinement*, J. Comput. Syst. Sci. 18, 1979, 35–46.

[21] R. Valk: *Infinite Behaviour of Petri Nets*, Theoretical Computer Science 25, 1983, 311–341.

[22] R. Valk: *Infinite Behaviour and Fairness*, in: Advances in Petri Nets 1986, LNCS 254, Springer-Verlag, 1986, 377–396.

[23] W. Vogler: *Behaviour Preserving Refinements of Petri Nets*, in: Graph-Theoretic Concepts in Computer Science, LNCS 246, 1897, 82–93.

[24] W. Vogler: *Failure Semantics of Petri Nets and the Refinement of Places and Transitions*, Technical Report TUM-I9003, Institut für Informatik, Technische Universität zu München, Germany, 1990.