Acta Cybernetica 12 (1996) 295-308.

On a Normal Form of Petri Nets*

Ferucio Laurenţiu ŢIPLEA[†] Masashi KATSURA [‡]
Masami ITO [‡]

Abstract

A Petri net is called (n, m)*-transition restricted* if its 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. Using the results from [6] it has been proved ([13]) that any λ-labelled Petri net is equivalent to a λ-labelled $(2,2)$-transition restricted Petri net, with respect to the finite transition sequence behaviour. This one may be considered as a normal form of Petri nets, called the *super-normal form* of Petri nets, and the question is whether it preserves or not the partial words and processes of Petri nets ([13]). In this paper we show that the answer to this question is positive for partial words and negative for processes. Then some infinite hierarchies of families of partial languages generated by (labelled) (n, m)-transition restricted Petri nets, are obtained.

1 Introduction and Preliminaries

A Petri net is a formalism that has been used intensively to model parallel computations ([9], [10]). Since the problems surrounding general Petri nets seem to be very difficult to analyze, it is sometimes necessary to narrow the scope of the investigation to subclasses of Petri nets. If such a subclass does not limit, in a precisely specified sense, the entire class of Petri nets then we say that the Petri nets in that subclass are in a certain *normal form*. For example, J.L. Peterson ([8]) considered the so-called *standard normal form* which preserves the sequential behaviour of Petri nets, and S. Crespi-Reghizzi and D. Mandrioli ([3]) considered the *semibounded normal form* which preserves the reachability. Another normal form, used to obtain logical characterizations of Petri nets ([5], [7]), was considered by E. Pelz in 1990 [6]. A Petri net in this normal form is called *normalized* and it is characterized by the fact that the weight function and the initial and final

*This work was carried out while the first author was visiting the Kyoto Sangyo University by a grant from the Ministry of Education (Monbusho), Japan

[†]correspondence address: Faculty of Science, Kyoto Sangyo University, 603 Kyoto, Japan, e-mail: `tiplea@ksuvx0.kyoto-su.ac.jp` and Faculty of Informatics "Al. I. Cuza" University, 6600 Iaşi, Romania e-mail:`fltiplea@infoiasi.ro`

[‡]Faculty of Science Kyoto Sangyo University 603 Kyoto, Japan e-mail: `katsura,ito@ksuvx0.kyoto-su.ac.jp`

markings take values in $\{0,1\}$. It was shown that for any labelled Petri net γ one can effectively construct a normalized labelled Petri net γ' such that γ and γ' have the same processes up to an isomorphism. As a result, the normalization preserves the finite and infinite transition sequence behaviour, step behaviour, and partial word behaviour ([6]).

In [12] 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 with respect to transitions ([13]) 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 were obtained, and in the case of λ-labelled Petri nets, the Pelz's 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 normalized and $(2,2)$-transition restricted net (with respect to the finite transition sequence behaviour).

A very natural question is whether this new normal form, called the *super-normal form*, preserves or not the partial words and processes of Petri nets. In this paper we show that the answer is positive for partial words and negative for processes. Therefore, whenever we deal with sequential or partial languages of λ-labelled Petri nets we may freely use normalized and $(2,2)$-transition restricted nets which could lead to simpler proofs and easiness of manipulation of complex structures. Then we turn our attention to labelled (n,m)-transition restricted Petri nets and we show that the families of partial languages (generated by such nets) form infinite hierarchies both with respect to n and to m.

First of all let us fix the terminology and notation that will be used in our paper (for details concerning Petri nets and processes the reader is referred to [1], [2], [7], [9], [10], [11]). 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 in the set B. The set of nonnegative integers is denoted by $\mathbf{N}$. 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 $(x,y) \notin F$. 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 ${}^\bullet x = \{y|(y,x) \in F\}$ and $x^\bullet = \{y|(x,y) \in F\}$. If t is a transition of Σ then Σ_t will denote the *subnet (of Σ) generated* by t (i.e. Σ_t contains only the transition t and all the places $s \in {}^\bullet t \cup t^\bullet$; the arcs between t and places are those from Σ, with their weights). The Petri nets that will be considered in this paper are without *isolated* places or transition, that is for any $x \in S \cup T$ the set ${}^\bullet x \cup x^\bullet$ is non-empty.

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. An *mPTN with final markings*, abbreviated *mPTNf*, is a 3-tuple $\gamma = (\Sigma, M_0, \mathcal{M})$, where the first two components form an $mPTN$ and $\mathcal{M}$, the *set of final markings* of γ, is a finite set of markings of Σ. A *labelled mPTN* (*mPTNf*, resp.), abbreviated $lmPTN$ ($lmPTNf$, resp.), is a 3-tuple (4-tuple, resp.) $\gamma = (\Sigma, M_0, l)$ ($\gamma = (\Sigma, M_0, \mathcal{M}, l)$, resp.) where the first two (three, resp.) components form an $mPTN$ ($mPTMf$, resp.) and l, the *labelling function* of γ, assigns to each transition a letter. A *λ-labelled mPTN* (*mPTNf*, resp.), abbreviated $l^\lambda mPTN$ ($l^\lambda mPTNf$, resp.), is defined as an $lmPTN$ ($lmPTMf$, resp.) with the difference that the labelling function, called now the *λ-labelling function* of γ, assigns to each transition either a letter or the empty word λ. In the sequel we shall often use the term "Petri net" or "net" whenever we refer to a structure γ defined as 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. The labelling function is denoted by placing letters into the boxes representing transitions and the final markings are explicitly listed.

Let γ be a net, M a marking and t a transition. We say that 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 notation "$[\cdot\rangle_\gamma$" will be simplified to "$[\cdot\rangle$" whenever γ is understood from the 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 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*. If N is an occurrence net we define its *final cut* as being the set $N^\circ = \{b \in B \| b^\bullet| = 0\}$; the *partially ordered set induced* by N is $(B \cup E, \prec_N)$, where $\prec_N = R^+$. In defining processes we will use *labelled occurrence nets* which are couples (N, p), where N is a occurrence net and p is a total function from $B \cup E$ into an alphabet V. For a Petri net $\Sigma = (S, T, F, W)$ and a labelled occurrence net (N, p) such that p is a function from $B \cup E$ into $S \cup T$ satisfying $p(B) \subseteq S$ and $p(E) \subseteq T$, define the *marking induced by N° in Σ* as being $M_{N^\circ}(s) = |\{b \in N^\circ | p(b) = s\}|$, for all $s \in S$. In what follows we adopt the inductive definition of processes ([2]). Let $\gamma = (\Sigma, M_0)$ be an $mPTN$. Define inductively the sets $\prod_0, \prod_1, \ldots$ by:

- $\prod_0$ contains only couples (N, p), where $N = (B, \emptyset, \emptyset)$, $|B| = \sum_{s \in S} M_0(s)$, and for each $s \in S$, B contains $M_0(s)$ distinct conditions b with $p(b) = s$ (this defines p as well);

- suppose $\prod_i$, $i \geq 0$, has already been constructed. For each $\pi = (N,p) \in \prod_i$, $N = (B,E,R)$, and each $t \in T$ such that $M_{N^\circ}[t\rangle$, we define a new labelled occurrence net $\pi' = (N',p')$, $N' = (B',E',R')$, as follows:

  (a) B', E' and R' will contain B, E and respectively, R;

  (b) for each $x \in B \cup E$, $p'(x) = p(x)$;

  (c) supplimentary, we add to N a new event e with $p'(e) = t$. For each $s \in S$ with $W(s,t) > 0$ we choose (from N°) $W(s,t)$ distinct conditions b labelled by s ($p(b) = s$) and we add the arcs (b,e) to R' for each such b. Similarly, for each $s \in S$ with $W(t,s) > 0$ we add (to N) $W(t,s)$ distinct conditions b labelled by s ($p'(b) = s$); the arcs (b,e), for each such b, are also added;

  (we shall sometimes write $\pi \xrightarrow{e(t)} \pi'$). Finally, consider $\prod_{i+1}$ as being the set of all the labelled occurrence nets defined as above.

Now, a *process* of γ is any element π of $\cup_{i\geq 0} \prod_i$. It is clear that for any such π there is at least a sequence $\pi_0, \pi_1, \ldots, \pi_m = \pi$ with $\pi_i \in \prod_i$ and π_{i+1} may be constructed from π_i as described above, for all $0 \leq i \leq m-1$. This sequence is called an *inductive definition* of π. As we can see events without pre-conditions or post-conditions are allowed too. In order to obtain processes of λ-labelled nets $\gamma = (\Sigma, M_0, l)$ what we have to do is to consider each process $\pi = (N,p)$ 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 $l \circ p$. Processes of labelled nets with final markings $\gamma = (\Sigma, M_0, l, \mathcal{M})$ are defined as being those processes π of (Σ, M_0, l) such that $M_{\pi^\circ} \in \mathcal{M}$.

To compare concurrent behaviour of nets we will use the *process* and *partial word equivalence*. Two processes $\pi = (N,p)$ and $\pi' = (N',p')$ (not necessary of the same net) are *isomorphic*, abbreviated $\pi \cong \pi'$, if there is a bijection $f : B \cup E \longrightarrow B' \cup E'$ such that:

(1) $p(x) = p'(f(x))$ for all $x \in E$, and
$p(x) = p'(f(x))$ whenever $x \in B$ and $p(x) \in S \cap S'$ or $p'(f(x)) \in S \cap S'$;

(2) $x \prec_\pi y$ iff $f(x) \prec_{\pi'} f(y)$ for all $x, y \in B \cup E$.

If γ and γ' are two nets such that for any process π of γ there is a process π' of γ' such that $\pi \cong \pi'$, and conversely, we say that γ and γ' are *process equivalent.*

Partial words are obtained in a similar way to processes by recording only the events which are not labelled by λ. Let $\pi = (N,p)$ be a process of a λ-labelled Petri 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, denoted $PWL(\gamma)$, is called the *partial language* of γ. Two nets with the same partial languages are called *partial word equivalent.*

For finite (infinite) transition sequence behaviour and step behaviour of Petri nets the reader is referred to [4], [6], [9], [14], [15]; the languages generated under these behaviours are called *finite* (*infinite*) *sequential languages* and *step languages.*

2 Normalization of λ-labelled Petri Nets

The notion of a normalized Petri net plays an important role in Petri net theory ([6], [7]). A Petri net is *normalized* if the weight function and the initial and final markings take values in $\{0,1\}$. E. Pelz developed in 1990 an algorithm for normalizing labelled Petri nets (with or without final markings) by preserving their processes. That is, for any labelled Petri net γ one can effectively compute a normalized labelled Petri net γ' such that γ and γ' are process equivalent. The Pelz's algorithm for normalizing Petri nets can be applied to λ-labelled Petri nets as well (the label λ is treated like any other label). So we can formulate the Pelz's result in a more general way, that is:

Theorem 2.1 *(E. Pelz, 1990)*
Let γ be a λ-labelled Petri net (with or without final markings). Then a normalized λ-labelled Petri net γ' can be effectively constructed such that γ and γ' have the same processes up to an isomorphism. Consequently, these two nets will have the same partial languages, step languages, and (finite or infinite) sequential languages.

We recall now a few notations and results concerning transition restricted Petri nets. A Petri net γ is called *(n,m)-transition restricted*, where n and m are non-zero natural numbers, if the range of the weight function is $\{0,1\}$ and $1 \leq |{}^\bullet t| \leq n$ and $1 \leq |t^\bullet| \leq m$ for all transitions t of γ. In [13] it has been shown that any λ-labelled Petri net is equivalent to a (2,2)-transition restricted Petri net, with respect to the finite transition sequence behaviour. In what follows we show that this result can be extented to the partial word behaviour but not to process behaviour. 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. The first one is to eliminate the cases $|{}^\bullet t| = 0$ and $|t^\bullet| = 0$, and it is in fact the same as in [13].

Transformation-A: Let γ be a normalized net. The next procedure yields a normalized net γ' satisfying $|{}^\bullet t| \geq 1$ and $|t^\bullet| \geq 1$ for all transitions t. Moreover, γ and γ' will have the same partial words.

For any transition t of Σ such that $|{}^\bullet t| = 0$ or $|t^\bullet| = 0$ we replace the subnet Σ_t by the net $\overline{\Sigma}_t$ as follows: if Σ_t is the net in Figure 2.1(a) (2.1(c), resp.) then $\overline{\Sigma}_t$ is the net in Figure 2.1(b) (2.1(d), resp.). Let Σ'

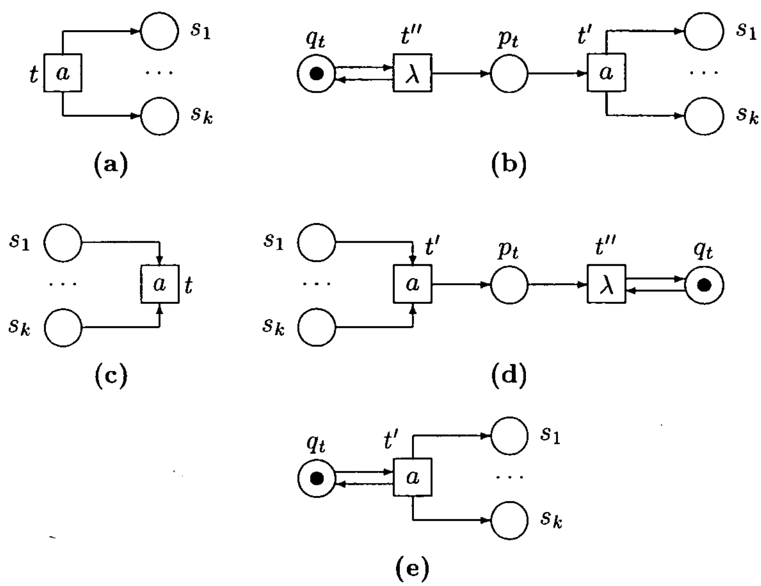


Figure 2.1

be the net thus obtained (for technical reasons we rename the transitions t of Σ' which have not been processed as above, by t'; thus, the set of transitions of Σ' is $T' = \{t' | t \in T \ \wedge \ |{}^\bullet t| \geq 1 \ \wedge \ |t^\bullet| \geq 1\} \cup \{t', t'' | t \in T \ \wedge \ (|{}^\bullet t| = 0 \ \vee \ |t^\bullet| = 0)\}$). For each $M \in \mathbf{N}^S$ we define the marking M' by:

$$M'(s) = \begin{cases} M(s), & \text{if } s \in S \\ 1, & \text{if } s = q_t \text{ for some } t \\ 0, & \text{otherwise,} \end{cases}$$

for all places s of Σ', and let l' be the labelling given by $l'(t') = l(t)$ and $l'(t'') = \lambda$ for all $t \in T$ (l being the labelling of γ). If γ has final markings, $\mathcal{M}$, then we add to (Σ', M_0', l') the set of final markings $\mathcal{M}' = \{M' | M \in \mathcal{M}\}$. Let γ' be the net obtained in this way.

End of Transformation-A.

It is clear that the net γ' yielded by Transformation-A is normalized and satisfies $|^\bullet t| \geq 1$ and $|t^\bullet| \geq 1$ for all transitions t. We show that:

Theorem 2.2 *Let γ be a normalized net and γ' the net yielded by Transformation A applied to γ. Then γ and γ' are partial word equivalent.*

Proof Let

$$(*)\quad \pi_0 \xrightarrow{e_1(t_1)} \pi_1 \xrightarrow{e_2(t_2)} \cdots \xrightarrow{e_n(t_n)} \pi_n = \pi$$

be an inductive definition of a process π of γ. We will replace each π_i by a sequence $\pi_i^1, \ldots, \pi_i^{k_i}$ $(1 \leq k_i \leq 2)$ of processes of γ' such that:

(1) $(\pi_i^{k_i})^\circ = (\pi_i)^\circ \cup Q_i$, where Q_i is a set of cardinality $|\{t \in T \,|\, |^\bullet t| = 0 \ \vee\ |t^\bullet| = 0\}|$, disjoint from $(\pi_i)^\circ$, and whose elements are one-to-one labelled from elements of the form q_t;

(2) $PW(\pi_i) = PW(\pi_i^{k_i})$.

Therefore, we will obtain a sequence

$$(**)\quad \pi_0^1, \ldots, \pi_0^{k_0}, \pi_1^1, \ldots, \pi_1^{k_1}, \ldots, \pi_m^1, \ldots, \pi_m^{k_m}$$

which is an inductive definition of a process $\pi' = \pi_m^{k_m}$ of γ' and $PW(\pi) = PW(\pi')$. This will prove the inclusion $PWL(\gamma) \subseteq PWL(\gamma')$ (if γ has final markings and M_{π° is such a marking then $M_{\pi'^\circ}$ will be a final marking of γ').

First we remark that we can define π_0^1 such that (1) and (2) be satisfied because $S \subseteq S'$. Then, we set $k_0 = 1$ and so (1) and (2) hold.

Suppose the process π_i has already been replaced by a sequence $\pi_i^1, \ldots, \pi_i^{k_i}$ $(1 \leq k_i \leq 2)$ of at most two processes of γ', satisfying (1) and (2). For π_{i+1} we have to consider three cases:

Case 1: t_{i+1} has not been processed by Transformation-A. Since (1) holds for $\pi_i^{k_i}$ it follows that we may define a process π_{i+1}^1 of γ' from $\pi_i^{k_i}$ and t'_{i+1}. Moreover, we may define π_{i+1}^1 such that $(\pi_{i+1}^1)^\circ = (\pi_{i+1})^\circ \cup Q_i$ (by setting $^\bullet(e_{i+1}^1) = {}^\bullet e_{i+1}$, $(e_{i+1}^1)^\bullet = e_{i+1}^\bullet$, and $p_{i+1}^1(x) = p_i^{k_i}(x)$ for all $x \in (e_{i+1}^1)^\bullet$, where $\pi_i^{k_i} \xrightarrow{e_{i+1}^1(t'_{i+1})} \pi_{i+1}^1$). We set now $k_{i+1} = 1$ and $Q_{i+1} = Q_i$, and it is easy to see that (1) and (2) are satisfied for $\pi_{i+1}^{k_{i+1}}$.

Case 2: $|^\bullet t_{i+1}| = 0$. Since (1) holds for $\pi_i^{k_i}$ it follows that we may define a process π_{i+1}^1 of γ' from $\pi_i^{k_i}$ and t''_{i+1}. Moreover, we will have $(\pi_{i+1}^1)^\circ = (\pi_i)^\circ \cup Q'_i \cup \{b\}$ and $PW(\pi_{i+1}^1) = PW(\pi_i^{k_i})$, where Q'_i is a set with the same properties as Q_i and b is a new condition $(b \notin (\pi_i)^\circ \cup Q'_i)$ labelled by $p_{t_{i+1}}$. Define now a new process π_{i+1}^2 from π_{i+1}^1 and t'_{i+1} such that $(\pi_{i+1}^2)^\circ = (\pi_{i+1})^\circ \cup Q'_i$ (by setting $^\bullet(e_{i+1}^2) = \{b\}$, $(e_{i+1}^2)^\bullet = e_{i+1}^\bullet$, and $p_{i+1}^2(x) = p_{i+1}(x)$ for all $x \in (e_{i+1}^2)^\bullet$, where $\pi_{i+1}^1 \xrightarrow{e_{i+1}^2(t'_{i+1})} \pi_{i+1}^2$). We set now $k_{i+1} = 2$ and $Q_{i+1} = Q'_i$, and so (1) holds true for $\pi_{i+1}^{k_{i+1}}$. In order to show that (2) also holds for $\pi_{i+1}^{k_{i+1}}$ it is sufficient to notice that any path

$$b_0, x_1, b_1, \ldots, b_k, x_{k+1}$$

such that $b_0 \in (\pi_0^1)^\circ$, $p_{i+1}^2(b_k) = q_{t_{i+1}}$ and $p_{i+1}^2(x_{k+1}) = l'(t'_{i+1})$ $(= l(t_{i+1}))$ has the properties:

(a) $p_{i+1}^2(b_0) = \cdots = p_{i+1}^2(b_{k-1}) = q_{t_{i+1}}$;

(b) $p_{i+1}^2(x_1) = \cdots = p_{i+1}^2(x_k) = \lambda$.

Using the construction of $\pi_{i+1}^{k_{i+1}}$ it is now easy to show that (2) holds true for $\pi_{i+1}^{k_{i+1}}$.

Case 3 $|t^\bullet{}_{i+1}| = 0$. This case is similar to the previous one: we have to define first π_{i+1}^1 from $\pi_i^{k_i}$ and t'_{i+1} and then π_{i+1}^2 from π_{i+1}^1 and t''_{i+1}.

Let us consider now the other inclusion. Let

$$(***)\quad \pi'_0 \xrightarrow{e'_1(x_1)} \pi'_1 \xrightarrow{e'_2(x_2)} \cdots \xrightarrow{e'_m(x_m)} \pi'_m = \pi'$$

be a process of γ'. The basic idea is to transform the sequence $(***)$ into a $(**)$-like sequence. To do that we define first the next two elementary transformations (initially, all the transitions $x_i \in T'$ which have not been processed by Transformation-A, that is no q- or p-like place is in ${}^\bullet x_i \cup x_i^\bullet$, are *marked* whereas the others are *unmarked*):

- **(T1)** determine the first i such that x_i is unmarked, is of the form t' for some $t \in T$, and $|{}^\bullet t| = 0$. We rearrange the sequence $(***)$ such that t is directly located after its unmarked matching pair $x_j = t''$, $j < i$ (it is easy to see that there is an unmarked matching pair t''; moreover, this rearrangement does not affect the process π'). Mark x_i and x_j and denote the result also by $(***)$;

- **(T2)** determine the first i such that x_i is unmarked, is of the form t' for some $t \in T$, and $|t^\bullet| = 0$. We rearrange the sequence $(***)$ such that t' is directly located before its unmarked matching pair $x_j = t''$, $j > i$ (if there does not exist a matching pair t'' for this transition t' we introduce one and apply it at π'_{i+1}). It is easy to see that we can do that and this rearrangement does not affect the process π'; mark x_i and x_j and denote the result also by $(***)$.

Now, we apply these two transformations to the sequence $(***)$ as long as they are possible and we obtain a new inductive definition of π'. Then we remove all the unmarked elemets of the form t'' and we recompute all the processes. Finally, we get a $(**)$-like sequence defining a process π'' with the property that $PW(\pi'') = PW(\pi')$. Now we "pack" this sequence into a $(*)$-like sequence (this operation is in fact the reverse of that which permitted us the passing from $(*)$ to $(**)$). The sequence obtained in this way defines a process π of γ such that $PW(\pi'') = PW(\pi)$ (the case of final markings is discussed as in the first part of the proof).

The theorem is completely proved. □

We want to point out that the structure in Figure 2.1(b) cannot be simplified to that in Figure 2.1(e) because in the first one the transition a can occur concurrently with itself but not in the second one.

Transformation-B: Let γ be a normalized net satisfying $|{}^\bullet t| \geq 1$ and $|t^\bullet| \geq 1$ for all transitions t. The next procedure yields a normalized and (2,2)-transition restricted net with the same partial language as γ.

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 2.2, 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 2.2 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 ${}^\bullet t \cap t^\bullet \neq \emptyset$ can be easily imagined). Let Σ' be the net such obtained. For each

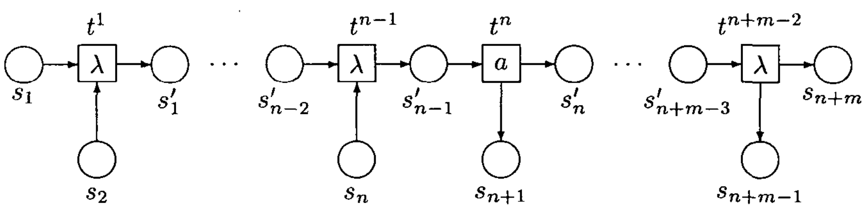


Figure 2.2

marking M of Σ we consider the marking M' given by:

$$M'(s) = \begin{cases} M(s), & \text{if } s \in S \\ 0, & \text{otherwise} \end{cases}$$

for all places s of Σ', and let l' be the labelling mentioned in each subnet (l' agrees with the labelling l of γ on all the transitions t which were not modified by the transformation). If γ has final markings, $\mathcal{M}$, we add to (Σ', M'_0, l') the set of final markings $\mathcal{M}' = \{M' | M \in \mathcal{M}\}$ and let γ' be the net such obtained.

End of Transformation-B.

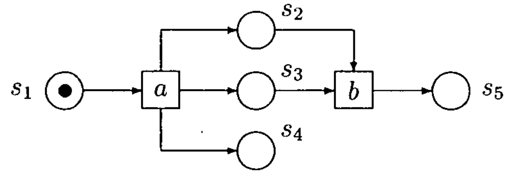
It is clear that the net γ' yielded by Transformation-B is normalized and (2,2)-transition restricted. We have:

Theorem 2.3 *Let γ be a normalized net satisfying $|{}^\bullet t| \geq 1$ and $|t^\bullet| \geq 1$ for all transitions t, and γ' the net yielded by Transformation-B applied to γ. Then γ and γ' are partial word equivalent.*

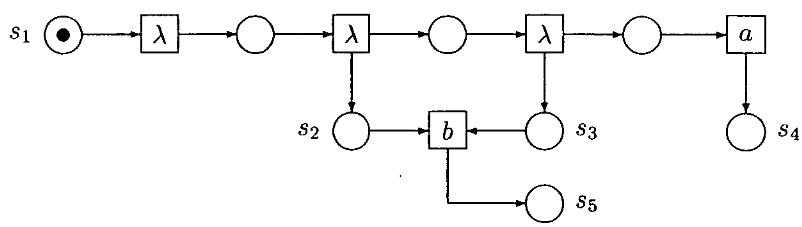
Proof The proof of this theorem is similar to the previous one. Therefore we will only sketch the differences (we use the same notations as in the proof of Theorem 2.2 and Transformation-A). Consider first the inclusion $PWL(\gamma) \subseteq PWL(\gamma')$. In the proof of Theorem 2.2 each transition t with $|{}^\bullet t| = 0$ or $|t^\bullet| = 0$ has been simulated by the "sequence" $t''t'$ or $t't''$ respectively. As a result any process π_i in the sequence $(*)$ has been either kept unchanged or replaced by at most two processes (obtained by applying t'' and t' or conversely). In the case of this theorem each transition t with $|{}^\bullet t| \geq 3$ or $|t^\bullet| \geq 3$ is simulated by a "sequence" $t^1 \cdots t^n \cdots t^{n+m}$ and so any process π_i in the sequence $(*)$ will be replaced by at most $max\{|{}^\bullet t| + |t^\bullet| | t \in T\}$ processes (in the same manner as in the proof of Theorem 2.2).

The inclusion $PWL(\gamma') \subseteq PWL(\gamma)$ is based on the fact that any transition t^n may occur only after $t^1, \ldots, t^{n-1}$. Therefore, $t^1, \ldots, t^n$ can be rearranged (in a $(***)$-like sequence) one after the other. The transitions $t^{n+1}, \ldots, t^{n+m-2}$ can also be applied after t^n (the missing transitions are supplimentary added). □

The position of the label a in Figure 2.2 cannot be changed, as the net in Figure 2.3 shows us. That is, α_1 is a partial word of the net in Figure 2.3(a) but not α_2, whereas the net in Figure 2.3(b), obtained from the net in Figure 2.3(a) by applying Transformation-B and changing the position of the label a, has α_2 as a partial word but not α_1.



(a)



(b)

$\alpha_1 : \; a \longrightarrow b$ $\qquad\qquad$ $\alpha_2 : \; \begin{matrix} b \\ a \end{matrix}$

Figure 2.3

Transformation-B and Theorem 2.3 show us that each Petri net is partial word equivalent with a (2,2)-transition restricted Petri net. This result cannot be extended to processes and to see that it is enough to consider the net in Figure 2.3(a) and to apply to it the Transformation-B. But, what we have to say is that our transformations preserve the finite sequential language and all types of infinitary languages (see [6]). Abbreviate "(n,m)-transition restricted Petri net" by $PTN(t_{n,m})$ and denote then

$$\begin{aligned}
\mathbf{PPW^{\lambda}(t_{n,m})} &= \{PWL(\gamma) | \gamma \text{ is an } l^{\lambda}mPTN(t_{n,m})\}, \\
\mathbf{LPW^{\lambda}(t_{n,m})} &= \{PWL(\gamma) | \gamma \text{ is an } l^{\lambda}mPTN(t_{n,m})f\},
\end{aligned}$$

for all $n, m \geq 1$.

Corollary 2.1 *For all $X \in \{P, L\}$ we have:*

1. $\mathbf{XPW^{\lambda}(t_{n,m})} = \mathbf{XPW^{\lambda}(t_{2,2})}$, $\forall n, m \geq 2$;
2. $\mathbf{XPW^{\lambda}(t_{n,1})} = \mathbf{XPW^{\lambda}(t_{2,1})}$, $\forall n \geq 2$;
3. $\mathbf{XPW^{\lambda}(t_{1,m})} = \mathbf{XPW^{\lambda}(t_{1,2})}$, $\forall m \geq 2$.

Proof *1* directly follow from Transformation-B and Theorem 2.3. *2* and *3* follows the same line as *1* with the remark that γ' yielded by Transformation-B is $(2,1)$-transition restricted ($(1,2)$-transition restricted, resp.) whenever the input net γ is $(n,1)$-transition restricted ($(1,m)$-transition restricted, resp.). □

Theorem 2.4 *The diagram in Figure 2.4 holds ("$\longrightarrow$" indicates a proper inclusion; the unrelated families are incomparable).*

$$\begin{tikzcd}
& \mathbf{XPW^{\lambda}(t_{1,1})} \arrow[dl] \arrow[dr] & \\
\mathbf{XPW^{\lambda}(t_{n,1})} = \mathbf{XPW^{\lambda}(t_{2,1})} \arrow[dr] & & \mathbf{XPW^{\lambda}(t_{1,2})} = \mathbf{XPW^{\lambda}(t_{1,m})} \arrow[dl] \\
& \mathbf{XPW^{\lambda}(t_{2,2})} = \mathbf{XPW^{\lambda}(t_{n,m})} &
\end{tikzcd}$$

Figure 2.4

Proof The inclusions follows from definitions, and the equalities from Corollary 2.1. To prove that the inclusions are proper it is enough to show that the families $\mathbf{XPW^{\lambda}(t_{1,2})}$ and $\mathbf{XPW^{\lambda}(t_{2,1})}$ are incomparable. Consider the partial language $L = \{\alpha\}$, where α is the partial word in Figure 2.5(a). It is clear that $L \in \mathbf{XPW^{\lambda}(t_{1,2})}$. Suppose for the sake of contradiction that there is a $(2,1)$-transition restricted net γ generating this partial word. Then, there is a process π of γ such

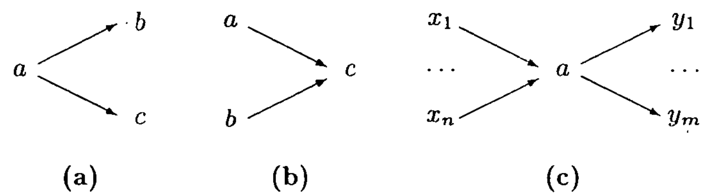


Figure 2.5

that $PW(\pi) = \alpha$. The process π will contain three events e_1, e_2, e_3 labelled by a, b, c. There are paths from e_1 to e_2 and from e_1 to e_3; let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two such paths. It is clear that there is no path from e_2 to e_3 or conversely. $\mathcal{D}_1$ and $\mathcal{D}_2$ have in common e_1 and therefore it must exist a common branching element, x. This x cannot be an event because the net γ is $(2,1)$-transition restricted, and it cannot be a condition because the conditions in occurrence nets are not branching elements; a contradiction. Hence, $L \notin \mathbf{XPW}^\lambda(\mathbf{t_{2,1}})$.

Similarly one can prove that $L' = \{\beta\}$, where β is the partial word in Figure 2.5(b), is a member of $\mathbf{XPW}^\lambda(\mathbf{t_{2,1}})$ but not of $\mathbf{XPW}^\lambda(\mathbf{t_{1,2}})$. Therefore, $\mathbf{XPW}^\lambda(\mathbf{t_{1,2}})$ and $\mathbf{XPW}^\lambda(\mathbf{t_{2,1}})$ are incomparable. □

It is interesting to compare the results in Theorem 2.4 with that in [13] where it was shown that $\mathbf{P}^\lambda(\mathbf{t_{n,1}}) = \mathbf{P}^\lambda(\mathbf{t_{1,1}})$ and $\mathbf{L}^\lambda(\mathbf{t_{n,1}}) = \mathbf{L}^\lambda(\mathbf{t_{1,1}}) = \mathbf{L}^\lambda(\mathbf{t_{1,m}})$, for all $n, m \geq 2$. To have full comparisons with [13] let us denote

$$\begin{array}{lcl}
\mathbf{PPW^f(t_{n,m})} & = & \{PWL(\gamma) | \gamma \text{ is an } mPTN(t_{n,m})\}, \\
\mathbf{PPW(t_{n,m})} & = & \{PWL(\gamma) | \gamma \text{ is an } lmPTN(t_{n,m})\}, \\
\mathbf{LPW^f(t_{n,m})} & = & \{PWL(\gamma) | \gamma \text{ is an } mPTN(t_{n,m})f\}, \\
\mathbf{LPW(t_{n,m})} & = & \{PWL(\gamma) | \gamma \text{ is an } lmPTN(t_{n,m})f\},
\end{array}$$

for all $n, m \geq 1$.

A partial word as that in Figure 2.5(c) is called an *(n,m)-star*, $n, m \geq 0$. Let π_1 and π_2 be two labelled partially ordered sets. We say that *π_1 is embedable in π_2* if π_1 is isomorphic with a part (subset) π_2' of π_2. If the condition "$x(\prec_{\pi_1} - \prec_{\pi_1}^2)y$ iff $x(\prec_{\pi_2'} - \prec_{\pi_2'}^2)y$ for all x, y" is suplimentary added then we say that *π_1 is strictly embedable in π_2*. Any (n,m)-transition restricted net has (n,m)-stars as partial words, but one can easily get (as in the proof of Theorem 2.4) that no (n,m)-star is strictly embedable in any process of an (n',m')-transition restricted net without λ-labels if $n' < n$ or $m' < m$. As a result we have:

Theorem 2.5 *The diagram in Figure 2.6 holds, for all $Y \in \{PPW^f, PPW, LPW^f, LPW\}$.*

$$
\begin{tikzcd}
 & & Y(t_{1,1}) \arrow[dl] \arrow[dr] & & \\
 & Y(t_{2,1}) \arrow[dl] \arrow[dr] & & Y(t_{1,2}) \arrow[dl] \arrow[dr] & \\
Y(t_{3,1}) & & Y(t_{2,2}) & & Y(t_{1,3}) \\
\dots & & \dots & & \dots
\end{tikzcd}
$$

Figure 2.6

References

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

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

[3] S. Crespi-Reghizzi, D. Mandrioli: *Petri Nets and Szilard Languages*, Information and Control 33, 1977, 177–192.

[4] M. Jantzen: *Language Theory of Petri Nets*, LNCS 254, Springer-Verlag, 1986, 379–412.

[5] M. Parigot, E. Pelz: *A Logical Approach of Petri Net Languages*, Theoretical Computer Science 39, 1985, 155–169.

[6] 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).

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

[8] J.L. Peterson. *Computation Sequence Sets*, Journal of Computer and System Sciences, vol. 13, 1976, 1–24.

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

[10] W. Reisig: *Petri Nets. An Introduction*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1985.

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

[12] 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, 151–189.

[13] F.L. Tiplea, C. Ene: *Hierarchies of Petri Net Languages and a Super-Normal Form*, in Proc. of the 2nd Intern. Conf. "Developments in Languages Theory", 1995, Magdeburg (Germany).

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

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

Received March, 1996