JUMPING PETRI NETS. SPECIFIC PROPERTIES

Ferucio Laurenţiu Ţiplea and Erkki Mäkinen

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TAMPERE

REPORT A-1996-8

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS A
A-1996-8, OCTOBER 1996

JUMPING PETRI NETS. SPECIFIC PROPERTIES

Ferucio Laurenţiu Ţiplea and Erkki Mäkinen

University of Tampere
Department of Computer Science
P.O.Box 607
FIN-33101 Tampere, Finland

ISBN 951-44-4050-1
ISSN 0783-6910

Jumping Petri Nets. Specific Properties

Ferucio Laurenţiu ŢIPLEA [1] **and Erkki MÄKINEN** [2]

[1] Faculty of Informatics
"Al. I. Cuza" University of Iaşi
6600 Iaşi, Romania
E-mail: fltiplea@infoiasi.ro

[2] Department of Computer Science
University of Tampere
P.O. Box 607, FIN-33101 Tampere, Finland
E-mail: em@cs.uta.fi

Abstract

A *Jumping Petri Net* ([18], [12]), $JPTN$ for short, is defined as a classical net which can spontaneously jump from a marking to another one. In [18] it has been shown that the reachability problem for $JPTN$'s is undecidable, but it is decidable for finite $JPTN$'s ($FJPTN$).

In this paper we establish some specific properties and investigate the computational power of such nets, via the interleaving semantics. Thus, we show that the non-labelled $JPTN$'s have the same computational power as the labelled or λ-labelled $JPTN$'s. When final markings are considered, the power of $JPTN$'s equals the power of Turing machines. The family of regular languages and the family of languages generated by $JPTN$'s with finite state space are shown to be equal. Languages generated by $FJPTN$'s can be represented in terms of regular languages and substitutions with classical Petri net languages. This characterization result leads to many important consequences, e.g. the recursiveness (context-sensitiveness, resp.) of languages generated by arbitrarily labelled (labelled, resp.) $FJPTN$'s. A pumping lemma for nonterminal jumping net languages is also established. Finally, some comparisons between families of languages are given, and a connection between $FJPTN$'s and a subclass of inhibitor nets is presented.

1 Introduction

It is well-known that the behaviour of some distributed systems cannot be adequately modelled by classical Petri nets. Many extensions which increase the computational and expressive power of Petri nets have been thus introduced. One direction has led to the modification of the firing rule of the nets ([2], [3], [5], [6], [7], [8], [12], [16], [17], [18], [19], [20], [21], [22]).

In this paper we consider jumping Petri nets as introduced in [18]. A jumping Petri net is a classical net Σ equipped with a (recursive) binary relation R on the markings of Σ. If $(M, M') \in R$ then the net Σ may "spontaneously jump" from M to M' (this is similar to

λ-moves in automata theory). When modelling systems using Petri nets, the extension to jumps is interesting for several reasons ([18]):

- unrelevant parts of the behaviour may be hidden,
- exception handling and recovery mechanisms can be added.

The paper is organized as follows. Section 2 presents the basic terminology, notations, and results concerning Petri nets and jumping Petri nets. In Section 3 we show that non-labelled $JPTN$'s have the same computational power as labelled or λ-labelled $JPTN$'s and, in the case of final markings, their power equals that of Turing machines. Section 4 gives some characterization results for $FJPTN$'s in terms of regular languages and substitutions with λ-free languages. Then some important consequences are derived and a pumping lemma for nonterminal jumping net languages is established. In Section 5 some comparisons between families of languages are given. The last section presents a connection between $FJPTN$'s and a subclass of inhibitor nets.

2 Preliminaries

The aim of this section is to establish the basic terminology, notations, and results concerning Petri nets in order to give the reader the necessary prerequisites for the understanding of this paper (for details the reader is referred to [1], [9], [11], [13], [14]).

The empty set is denoted by $\emptyset$; for a finite set A, $|A|$ denotes the cardinality of A and $\mathcal{P}(A)$ denotes the set of all subsets of A. Given the sets A and B, $A \subseteq B$ ($A \subset B$, resp.) denotes the inclusion (strict inclusion, resp.) of A in B. If $R \subseteq A \times B$ then $dom(R)$ and $cod(R)$ denote the sets $dom(R) = \{a \in A | \exists b \in B : \ (a,b) \in R\}$ and $cod(R) = \{b \in B | \exists a \in A : \ (a,b) \in R\}$. The set of integers (nonnegative integers, positive integers, resp.) is denoted by $\mathbf{Z}$ ($\mathbf{N}$, $\mathbf{N}^+$, resp.).

For a (finite) alphabet V, V^* denotes the free monoid generated by V (under the concatenation operation) with the empty word λ. Given a word $w \in V^*$, $|w|$ denotes the length of w, and $alph(w)$ denotes the set of all letters occurring in w. The $alph$-notation is extended by union to sets of words (languages). $\mathcal{L}_3$ ($\mathcal{L}_2$, $\mathcal{L}_1$, $\mathcal{L}_{rec}$, resp.) denotes the family of regular (context-free, context-sensitive, recursive, resp.) languages and $\mathcal{L}_{3,pref}$ denotes the family of regular prefix languages (i.e., regular languages containing all prefixes of their words).

2.1 Petri Nets

A (finite) *P/T-net* (with infinite capacities), 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 $W : (S \times T) \cup (T \times S) \to \mathbf{N}$ is the *weight function* of Σ verifying $W(x,y) = 0$ iff $(x,y) \notin F$. A *marking* of a PTN Σ is a function $M : S \to \mathbf{N}$; it will be sometimes identified with a vector $M \in \mathbf{N}^{|S|}$. The operations and relations on vectors are componentwise defined. $\mathbf{N}^S$ denotes the set of all markings of Σ.

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 (label). 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 often use the term "Petri net" (PN) or "net" whenever we refer to a PTN ($mPTN$, $mPTNf$, $lmPTN$, $lmPTNf$, $l^\lambda mPTN$, $l^\lambda mPTNf$) γ and it is not necessary to specify its type (i.e., marked, labelled, etc.); moreover, we implicitly assume that the components of γ are defined as above. The term "Petri net" or "net" will be sometimes used together with other terms. For instance, the term "labelled net" denotes a net which is at least labelled. The first component Σ of a PN γ 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.

Graphically, a net γ is represented by a graph. The places are denoted by circles and transitions by boxes; the flow relation is represented by arcs. An 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, $t \in T$ and $w \in T^*$. We define the functions t^-, t^+, and Δw from S into $\mathbf{Z}$ by $t^-(s) = W(s,t)$, $t^+(s) = W(t,s)$ and $\Delta w(s) = 0$ if $w = \lambda$ and $\Delta w(s) = \sum_{i=1}^{n}(t_i^+ - t_i^-)$ if $w = t_1 \ldots t_n, n \geq 1$, for all $s \in S$. The sequential behaviour of γ is given by so-called *transition (firing) rule*, which consists of

(i) the *enabling rule*: a transition t is *enabled* at a marking M (in γ), abbreviated $M[t\rangle_\gamma$, iff $t^- \leq M$;

(ii) the *computing rule*: if $M[t\rangle_\gamma$, then t may *occur* yielding a new marking M', abbreviated $M[t\rangle_\gamma M'$, defined by $M' = M + \Delta t$.

In fact, for any transition t of γ we have defined a binary relation on $\mathbf{N}^S$, denoted by $[t\rangle_\gamma$ and given by

$$M[t\rangle_\gamma M' \;\text{ iff }\; t^- \leq M \text{ and } M' = M + \Delta t.$$

If $t_1, \ldots, t_n, n \geq 1$, are transitions of γ, the classical product of the relations $[t_1\rangle_\gamma, \ldots, [t_n\rangle_\gamma$ will be denoted by $[t_1 \cdots t_n\rangle_\gamma$; i.e. $[t_1 \cdots t_n\rangle_\gamma = [t_1\rangle_\gamma \circ \ldots \circ [t_n\rangle_\gamma$. Moreover, we consider the relation $[\lambda\rangle_\gamma$ given by $[\lambda\rangle_\gamma = \{(M, M) \mid M \in \mathbf{N}^S\}$.

Let γ be a marked Petri net and M_0 its initial marking. The word $w \in T^*$ is called a *transition sequence* of γ if there exists a marking M of γ such that $M_0[w\rangle_\gamma M$. Moreover, the marking M is called *reachable* (from M_0) in γ. The *set of all reachable markings* of γ is denoted by $[M_0\rangle_\gamma$. The notation "$[\cdot\rangle_\gamma$" will be simplified to "$[\cdot\rangle$" whenever γ is understood from the context.

Petri nets may be considered as generators of languages. Let γ_1 be an $mPTN$, γ_2 either an $lmPTN$ or an $l^\lambda mPTN$, γ_3 an $mPTNf$, and γ_4 either an $lmPTNf$ or an $l^\lambda mPTNf$.

The languages generated by these nets are defined as follows:

$$P(\gamma_1) = \{w|w \in T^* \ \wedge \ (\exists M \in \mathbf{N}^S : \ M_0[w\rangle_{\gamma_1} M)\},$$
$$P(\gamma_2) = \{l(w)|w \in T^* \ \wedge \ (\exists M \in \mathbf{N}^S : \ M_0[w\rangle_{\gamma_2} M)\},$$
$$L(\gamma_3) = \{w|w \in T^* \ \wedge \ (\exists M \in \mathcal{M} : \ M_0[w\rangle_{\gamma_3} M)\},$$
$$L(\gamma_4) = \{l(w)|w \in T^* \ \wedge \ (\exists M \in \mathcal{M} : \ M_0[w\rangle_{\gamma_4} M)\}.$$

The languages generated by $mPTN$ ($lmPTN$, $l^\lambda mPTN$, resp.) are called *free P-type languages* (*P-type languages*, *arbitrary P-type languages*, resp.) and the family of these languages is denoted by $\mathbf{P^f}$ ($\mathbf{P}$, $\mathbf{P}^\lambda$, resp.). For Petri nets with final markings the notation and terminology is obtained by changing "P" into "L". Sometimes, "nonterminal" ("terminal", resp.) is used instead of "P-type" ("L-type", resp.). These languages are usually referred to as *Petri net languages* or *Petri languages*.

2.2 Jumping Petri Nets

A *jumping P/T-net* ([18]), abbreviated $JPTN$, is a pair $\gamma = (\Sigma, R)$, where Σ is a PTN and R, the *set of (spontaneous) jumps* of γ, is a binary relation on the set of markings of Σ. In what follows the set R of jumps of any $JPTN$ will be assumed *recursive*, that is for any couple of markings (M, M') we can effectively decide whether or not (M, M') is a member of R. Let $\gamma = (\Sigma, R)$ be a $JPTN$. The pairs $(M, M') \in R$ are referred to as *jumps* of γ. If γ has finitely many jumps then we say that γ is a *finite* $JPTN$, abbreviated $FJPTN$.

Let $Y \in \{JPTN, FJPTN\}$. An mY (mYf, lmY, $lmYf$, $l^\lambda mY$, $l^\lambda mYf$, resp.) is defined as an $mPTN$ ($mPTNf$, $lmPTN$, $lmPTNf$, $l^\lambda mPTN$, $l^\lambda mPTNf$, resp.), by changing "Σ" into "Σ, R". For instance, $\gamma = (\Sigma, R, M_0, \mathcal{M}, l)$ denotes either an $lmYf$ or an $l^\lambda mYf$. We shall use the term "jumping net" (JN) ("finite jumping net" (FJN), resp.) to denote any of the following: $JPTN$, $mJPTN$, $mJPTNf$, $lmJPTN$, $lmJPTNf$, $l^\lambda mJPTN$, $l^\lambda mJPTNf$ ($FJPTN$, $mFJPTN$, $mFJPTNf$, $lmFJPTN$, $lmFJPTNf$, $l^\lambda mFJPTN$, $l^\lambda mFJPTNf$, resp.). In fact, all remarks about Petri nets equally hold for jumping Petri nets. Graphically, a jumping net will be represented as a classical net. Moreover, the relation R will be separately listed.

The behaviour of a jumping net γ is given by the *j-transition (j-firing) rule*, which consists of

(i) the *j-enabling rule*: the transition t is *j-enabled at a marking* M (in γ), abbreviated $M[t\rangle_{\gamma,j}$, iff there exists a marking M_1 such that $MR^*M_1[t\rangle_\Sigma$ (Σ being the underlying net of γ and R^* the reflexive and transitive closure of R);

(ii) the *j-computing rule*: the marking M' is *j-produced* by occurring t at M, abbreviated $M[t\rangle_{\gamma,j}M'$, iff there exist markings M_1, M_2 such that $MR^*M_1[t\rangle_\Sigma M_2 R^* M'$.

The notions of *transition j-sequence* and *j-reachable marking* are defined similarly as for Petri nets (the relation $[\lambda\rangle_{\gamma,j}$ is defined by $[\lambda\rangle_{\gamma,j} = \{(M, M')|M \in \mathbf{N}^S, \ MR^*M'\}$). The *set of all j-reachable markings* of a marked JN γ is denoted by $[M_0\rangle_{\gamma,j}$ (M_0 being the initial marking of γ). The notation "$[\cdot\rangle_{\gamma,j}$" will be simplified to "$[\cdot\rangle_j$" whenever γ is understood from the context.

Jumping nets may be considered as generators of languages in the same way as classical nets, by changing "$[\cdot\rangle$" into "$[\cdot\rangle_j$". For example, if $\gamma = (\Sigma, R, M_0, \mathcal{M}, l)$ is an $l^\lambda mJPTNf$, then the language generated by γ is

$$L(\gamma) = \{l(w) | w \in T^* \ \wedge \ (\exists M \in \mathcal{M}: \ M_0[w\rangle_j M)\}.$$

Thus, $\mathbf{RX^f}$ ($\mathbf{RX}$, $\mathbf{RX^\lambda}$, resp.) will denote the family of *free X-type jumping Petri net languages* (*X-type jumping Petri net languages, arbitrary X-type jumping Petri net languages,* resp.), for any $X \in \{P, L\}$. For finite jumping nets, the corresponding family of languages will be denoted by $\mathbf{RX^f_{fin}}$ ($\mathbf{RX_{fin}}$, $\mathbf{RX^\lambda_{fin}}$, resp.). For any $X \in \{P, L\}$ we have:

$$\begin{array}{lllll} \mathbf{X^f} & \subseteq & \mathbf{RX^f_{fin}} & \subseteq & \mathbf{RX^f}, \\ \mathbf{X} & \subseteq & \mathbf{RX_{fin}} & \subseteq & \mathbf{RX}, \\ \mathbf{X^\lambda} & \subseteq & \mathbf{RX^\lambda_{fin}} & \subseteq & \mathbf{RX^\lambda}. \end{array}$$

Some jumps of an FJN may be never used. Thus we say that a marked finite jumping net γ is *R-reduced* if for any jump (M, M') of γ we have $M \neq M'$, $M \in [M_0\rangle_{\gamma,j}$, and there is a final marking of γ which is j-enabled at M' (provided that final markings are defined).

As the reachability problem for FJN's is decidable ([18]), for any marked FJN γ we can effectively construct (modifying only the set of jumpings of γ) a marked FJN γ' such that γ' is R-reduced and it has the same computational power as γ. All finite jumping nets in this paper will be considered R-reduced.

3 Jumps and Labellings

In this section we show that the jumps can "simulate" the labelling of nets. Then we use this result to prove that the power of $JPTN$'s with final markings equals the class of recursively enumerable languages. In the case of jumping nets with finite state space the connection with regular languages is shown.

Theorem 3.1 *For any $X \in \{P, L\}$ we have* $\mathbf{RX^f} = \mathbf{RX} = \mathbf{RX^\lambda}$.

Proof It is enough to prove that $\mathbf{RX^\lambda} \subseteq \mathbf{RX^f}$, and we will first do it for the case $X = L$.

Let $L \in \mathbf{RL^\lambda}$ and $\gamma = (\Sigma, R, M_0, \mathcal{M}, l)$ be an $l^\lambda mJPTNf$ such that $L = L(\gamma)$. Without loss of generality we may assume that $T \cap \{l(t) | t \in T\} = \emptyset$, where $\Sigma = (S, T; F, W)$. Let $T_1 = \{t \in T | (\forall t' \in T)(t \neq t' \ \Rightarrow \ l(t) \neq l(t'))\} \subseteq T$, $T_2 = \{t \in T | (\exists t' \in T)(t \neq t' \ \wedge \ l(t) = l(t'))\}$, and $T_3 = \{t \in T | l(t) = \lambda\}$. It is easy to see that $T = T_1 \cup T_2 \cup T_3$.

If $T_2 = T_3 = \emptyset$ we consider $\gamma' = (\Sigma', R, M_0, \mathcal{M})$, where Σ' is obtained from Σ by renaming each transition t by $l(t)$. γ' is an $mJPTNf$ and $L(\gamma') = L$.

If $T_2 \neq \emptyset$ or $T_3 \neq \emptyset$ we construct $\gamma' = (\Sigma, R', M_0', \mathcal{M}')$ as described below. We partition the set $T_2 = T - T_1$ into $k \geq 1$ subsets, $T_2 = T_2^1 \cup \cdots \cup T_2^k$, such that for any i, $1 \leq i \leq k$, the set T_2^i contains those transitions of Σ which have the same label; let a_i be this label. We have $a_i \neq a_j$ for any $i \neq j$. The set of transitions of Σ' will be $T' = l(T_1) \cup T_2 \cup \{a_1, \ldots, a_k\} \cup T_3$ (for a set A, $l(A)$ stands for the set $\{l(a) | a \in A\}$). The basic idea is: when a transition $t \in T_1$ occurs in γ, its effect is simulated in γ' by the transition $l(t) \in l(T_1)$; when a transition $t \in T_2^i$ occurs in γ, $1 \leq i \leq k$, its effect is simulated in γ' by the relation R' and the transition a_i. Finally, when a transition $t \in T_3$ occurs in γ, its effect is simulated in γ' by the relation R'. The transitions of $T_2 \cup T_3$ will be blocked forever in γ'. Formally, γ' is given by:

(i) $T' = T_1' \cup T_2 \cup \{a_1, \ldots, a_k\} \cup T_3$, where $T_1' = l(T_1)$. We have $T_2 \cap \{a_1, \ldots, a_k\} = \emptyset$;

(ii) $S' = S \cup \{s_0, s_0', s_1, s_1', \ldots, s_k, s_k'\}$, where $s_0, s_0', s_1, s_1', \ldots, s_k, s_k'$ are new places. All the markings of Σ' will be written in the form

$$(M, \underbrace{\alpha_0}_{s_0}, \underbrace{\alpha_0'}_{s_0'}, \underbrace{\alpha_1}_{s_1}, \underbrace{\alpha_1'}_{s_1'}, \ldots, \underbrace{\alpha_k}_{s_k}, \underbrace{\alpha_k'}_{s_k'}),$$

where $M \in \mathbf{N}^S$ and α_i, α_i' are nonnegative integers;

(iii) $F' = F_1' \cup F_1'' \cup F_2' \cup F_2'' \cup F_3$, where
$F_1' = \{(s, l(t)) | s \in S,\ t \in T_1,\ (s,t) \in F\} \cup \{(l(t), s) | s \in S,\ t \in T_1,\ (t,s) \in F\}$,
$F_1'' = \{(s_0, t) | t \in T_1'\}$,
$F_2' = \{(s,t) \in F | t \in T_2 \cup T_3\} \cup \{(t,s) \in F | t \in T_2 \cup T_3\}$,
$F_2'' = \{(s_0', t) | t \in T_2 \cup T_3\}$,
$F_3 = \{(s_i, a_i) | 1 \le i \le k\} \cup \{(a_i, s_i') | 1 \le i \le k\}$;

(iv) $$W'(x,y) = \begin{cases} W(x,y), & \text{if } (x,y) \in F_2' \\ 1, & \text{if } (x,y) \in F_1'' \cup F_2'' \cup F_3 \\ W(x,t), & \text{if } y = l(t),\ t \in T_1,\ (x,y) \in F_1' \\ W(t,y), & \text{if } x = l(t),\ t \in T_1,\ (x,y) \in F_1'; \end{cases}$$

(v) $R' = R_0 \cup R_1 \cup R_2' \cup R_2'' \cup R_3$, where
$R_0 = \{((M_1, 0, 0, 0, 0, \ldots, 0, 0), (M_2, 0, 0, 0, 0, \ldots, 0, 0)) | (M_1, M_2) \in R\}$,
$R_1 = \{((M, 0, 0, 0, 0, \ldots, 0, 0), (M, 1, 0, 0, 0, \ldots, 0, 0)) |$
$M \in \mathbf{N}^S,\ (\exists t \in T_1 :\ M[t\rangle_\Sigma)\}$,
$R_2' = \{((M, 0, 0, 0, 0, \ldots, 0, 0), (M, 0, 0, 0, 0, \ldots, \underbrace{1}_{s_i}, \underbrace{0}_{s_i'}, \ldots, 0, 0)) |$
$M \in \mathbf{N}^S,\ 1 \le i \le k,\ (\exists t \in T_2^i :\ M[t\rangle_\Sigma)\}$,
$R_2'' = \{((M, 0, 0, 0, 0, \ldots, \underbrace{0}_{s_i}, \underbrace{1}_{s_i'}, \ldots, 0, 0), (M_1, 0, 0, 0, 0, \ldots, 0, 0)) |$
$M \in \mathbf{N}^S,\ 1 \le i \le k,\ (\exists t \in T_2^i :\ M[t\rangle_\Sigma M_1)\}$,
$R_3 = \{((M_1, 0, 0, 0, 0, \ldots, 0, 0), (M_2, 0, 0, 0, 0, \ldots, 0, 0)) |$
$M \in \mathbf{N}^S,\ (\exists t \in T_3 :\ M_1[t\rangle_\Sigma M_2)\}$.
The relation R' such defined is recursive;

(vi) $M_0' = (M_0, 0, 0, 0, 0, \ldots, 0, 0)$;

(vii) $\mathcal{M}' = \{(M, 0, 0, 0, 0, \ldots, 0, 0) | M \in \mathcal{M}\}$.

Graphically, the net Σ' is represented in Figure 3.1. We show that $L = L(\gamma')$. Let us first consider the inclusion $L \subseteq L(\gamma')$. It is enough to prove that if $M_0[w\rangle_{\gamma,j} M$, $w \in T^*$, then

$$(M_0, 0, 0, 0, 0, \ldots, 0, 0)[l(w)\rangle_{\gamma',j}(M, 0, 0, 0, 0, \ldots, 0, 0).$$

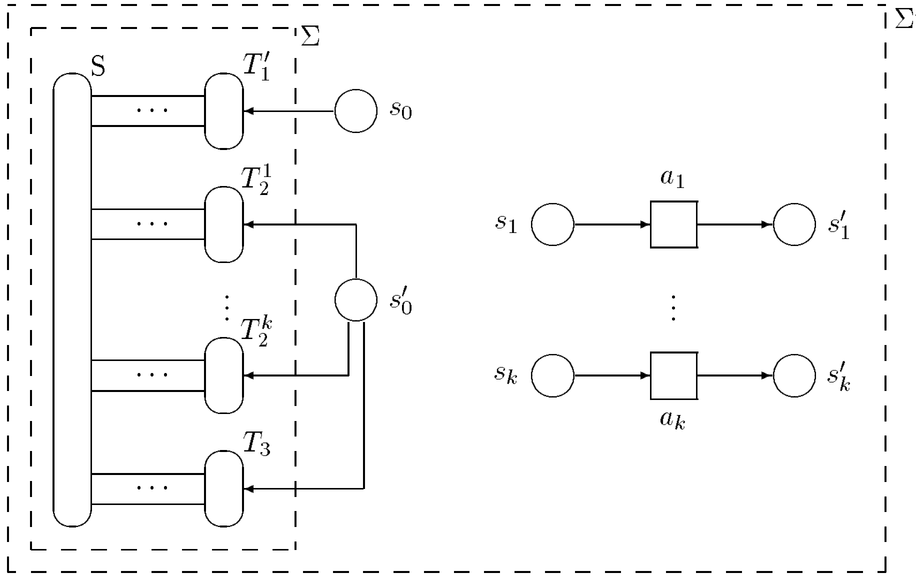

Figure 3.1

By induction on $|w|$ it is enough to prove that if $M_1 \in \mathbf{N}^S$, $t \in T$ and $M_1[t\rangle_{\gamma,j} M_2$ then

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0)[l(t)\rangle_{\gamma',j}(M_2, \ldots, 0, 0, 0, 0, \ldots, 0, 0).$$

Suppose $M_1[t\rangle_{\gamma,j} M_2$. Then, there exist $M_3, M_4 \in \mathbf{N}^S$ such that $M_1 R^* M_3 [t\rangle_\Sigma M_4 R^* M_2$. We have to consider three cases.

Case 1 $t \in T_1$. Then $l(t) \in T_1'$ and

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_3, 0, 0, 0, 0, \ldots, 0, 0) R_1$$

$$(M_3, 1, 0, 0, 0, \ldots, 0, 0)[l(t)\rangle_{\Sigma'}(M_4, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_2, 0, 0, 0, 0, \ldots, 0, 0).$$

Case 2 $t \in T_2^i$ for some i, $1 \le i \le k$. Then,

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_3, 0, 0, 0, 0, \ldots, 0, 0) R_2'$$

$$(M_3, 0, 0, 0, 0, \ldots, \underbrace{1}_{s_i}, \underbrace{0}_{s_i'}, \ldots, 0, 0)[a_i = l(t)\rangle_{\Sigma'}(M_3, 0, 0, 0, 0, \ldots, \underbrace{0}_{s_i}, \underbrace{1}_{s_i'}, \ldots, 0, 0)$$

$$R_2''(M_4, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_2, 0, 0, 0, 0, \ldots, 0, 0).$$

Case 3 $t \in T_3$. Then,

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_3, 0, 0, 0, 0, \ldots, 0, 0) R_3$$

$$(M_4, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_2, 0, 0, 0, 0, \ldots, 0, 0),$$

that is $(M_1, 0, 0, 0, 0, \ldots, 0, 0)[\lambda = l(t)\rangle_{\gamma',j}(M_2, 0, 0, 0, 0, \ldots, 0, 0)$.

Hence, if $M_1[t\rangle_{\gamma,j}M_2$ then

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0)[l(t)\rangle_{\gamma',j}(M_2, 0, 0, 0, 0, \ldots, 0, 0).$$

Thus we have $L \subseteq L(\gamma')$.

As for the converse ($L(\gamma') \subseteq L$) it is necessary to note that each j-reachable marking in γ' is of one of the following forms:

(1) $(M, 0, 0, 0, 0, \ldots, 0, 0)$,

(2) $(M, 1, 0, 0, 0, \ldots, 0, 0)$,

(3) $(M, 0, 0, 0, 0, \ldots, \underbrace{1}_{s_i}, \underbrace{0}_{s_i'}, \ldots, 0, 0)$,

(4) $(M, 0, 0, 0, 0, \ldots, \underbrace{0}_{s_i}, \underbrace{1}_{s_i'}, \ldots, 0, 0)$,

where $M \in [M_0\rangle_{\gamma,j}$ and $i \in \{1, \ldots, k\}$.
No transition of γ' is enabled at a (1)- or (4)-type marking. Only a jump by R' makes it possible to pass from a (1)- or (4)-type marking to a (1)-, (2)- or (3)-type marking. In γ' the transitions are enabled only at the (2)- or (3)-type markings. Now it is enough to prove that if $u \in (T')^*$ and

$$(M_0, 0, 0, 0, 0, \ldots, 0, 0)[u\rangle_{\gamma',j}(M, \theta_0, \theta_0', \theta_1, \theta_1', \ldots, \theta_k, \theta_k')$$

then there is $w \in T^*$ such that $l(w) = u$ and $M_0[w\rangle_{\gamma,j}M$. Taking into account that any jump by R_3 can be simply simulated by a transition labelled by λ, it remains to be shown that if $t' \in T' - T_3$, $M_1 \in \mathbf{N}^S$,

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0)[t'\rangle_{\gamma',j}(M_2, 0, 0, 0, 0, \ldots, 0, 0)$$

and no jump by R_3 is used in this computation, then there is $t \in T$ such that $l(t) = t'$ and $M_1[t\rangle_{\gamma,j}M_2$. We have to consider two cases.
Case $1'$ $t' \in T_1' = l(T_1)$. Then there is an unique $t \in T$ such that $l(t) = t'$. From

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0)[t'\rangle_{\gamma',j}(M_2, 0, 0, 0, 0, \ldots, 0, 0)$$

it follows that there exist $M_3, M_4 \in \mathbf{N}^S$ such that

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0)R_0^*(M_3, 0, 0, 0, 0, \ldots, 0, 0)R_1$$

$$(M_3, 1, 0, 0, 0, \ldots, 0, 0)[t' = l(t)\rangle_{\Sigma'}(M_4, 0, 0, 0, 0, \ldots, 0, 0)R_0^*(M_2, 0, 0, 0, 0, \ldots, 0, 0).$$

But $l(t)$ acts as t on the same places of S and hence $M_1R^*M_3[t\rangle_\Sigma M_4R^*M_2$ which shows us that $M_1[t\rangle_{\gamma,j}M_2$.
Case $2'$ There is $i \in \{1, \ldots, k\}$ such that $t' = a_i$. Then, for any $t \in T_2^i$ we have $l(t) = a_i$. From

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0)[t' = a_i\rangle_{\gamma',j}(M_2, 0, 0, 0, 0, \ldots, 0, 0)$$

it follows that there exist $M_3, M_4 \in \mathbf{N}^S$ such that

$$(M_1, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_3, 0, 0, 0, 0, \ldots, 0, 0) R_2'$$

$$(M_3, 0, 0, 0, 0, \ldots, \underbrace{1}_{s_i}, \underbrace{0}_{s_i'}, \ldots, 0, 0)[a_i\rangle_{\Sigma'} (M_3, 0, 0, 0, 0, \ldots, \underbrace{0}_{s_i}, \underbrace{1}_{s_i'}, \ldots, 0, 0) R_2''$$

$$(M_4, 0, 0, 0, 0, \ldots, 0, 0) R_0^* (M_2, 0, 0, 0, 0, \ldots, 0, 0).$$

From the definition of R_2'' it follows that there is $t \in T_2^i$ such that $M_3[t\rangle_\Sigma M_4$. Moreover $l(t) = a_i = t'$. Hence we have $M_1 R^* M_3 [t\rangle_\Sigma M_4 R^* M_2$ which shows that $M_1[t\rangle_{\gamma,j} M_2$.

We have proved that $\mathbf{RL}^\lambda = \mathbf{RL}^\mathbf{f}$. To accomplish the proof of the theorem we note that for the case $X = P$ the net γ' is constructed in the same way as it was described. The difference is that the final markings are not used. □

Remark 3.1 *Usually in Petri net theory, isolated places and transitions are not allowed, and this is the reason that in the proof of Theorem 3.1 the set $T_2 \cup T_3$ has not been removed from T'.*

Theorem 3.2 $\mathbf{RL}^\mathbf{f} = \mathbf{RL} = \mathbf{RL}^\lambda = \mathcal{L}_0$.

Proof The equalities $\mathbf{RL}^\mathbf{f} = \mathbf{RL} = \mathbf{RL}^\lambda$ have been already established. The equality with the set of all recursively enumerable languages can be obtained as follows.

In [18] it has been proved that jumping Petri nets can simulate inhibitor nets (which have the power of Turing machines). As a consequence, $\mathcal{L}_0 \subseteq \mathbf{RL}^\lambda$. Now we prove that $\mathbf{RL}^\mathbf{f} \subseteq \mathcal{L}_0$. Let $\gamma = (\Sigma, R, M_0, \mathcal{M})$ be an $mJPTNf$. We show that there is an algorithm $\mathcal{A}$ such that for all $w \in T^*$ we have

$w \in L(\gamma)$ iff $\mathcal{A}$ beginning with the input w it will eventually halt accepting w.

First we have to remark that $w \in L(\gamma)$ iff there is a computation in γ of the form

$$M_0 R^* M_0' [w_1\rangle M_1 R^+ M_1' \cdots M_{k-1} R^+ M_{k-1}' [w_k\rangle M_k R^* M,$$

where $M \in \mathcal{M}$ and $w_1, \ldots, w_k$ ($k \geq 1$) is a decomposition of w into non-empty words, that is $w = w_1 \cdots w_k$ and none of w_i is empty. All the computations of the above form will be called *terminal computations* in γ. A terminal computation can be written as a (formal) string

$$(M_0, M_0') w_1 (M_1, M_1') \cdots (M_{k-1}, M_{k-1}') w_k (M_k, M),$$

where $(M_0, M_0'), (M_k, M) \in R^*$, $(M_1, M_1'), \ldots, (M_{k-1}, M_{k-1}') \in R^+$ and $w_1, \ldots, w_k \in T^+$ (the empty transition sequence is identified by a string of the form $(M_0 R^* M_0')$).

It is clear that not any string of the above form describes a terminal computation in γ. But if we have such a string we can effectively decide whether or not it describes a terminal computation in γ.

Since R is recursive, R^* is recursively enumerable and consequently, we can enumerate R^* by

$$r_0, r_1, \ldots, r_n, \ldots$$

(for any $n \geq 0$, r_n is a couple (M, M') satisfying MR^*M').

Any $w \in T^*$ has finitely many decompositions $w = w_1 \cdots w_k$ ($k \geq 1$) with $w_i \in T^+$ for all i, and let $d_1, \ldots, d_m$ ($m \geq 1$) be all these decompositions. For any decomposition d_i ($1 \leq i \leq m$),

$$d_i : \quad w = w_1 \cdots w_{k_i},$$

we consider the N-indexed sequence S_i defined by:

- consider first all strings obtained from d_i and r_0 as above (in this case we have only one string $r_0 w_1 r_0 \cdots r_0 w_{k_i} r_0$);
- consider then, in an arbitrary but fixed order, all strings as above obtained from d_i and r_0, r_1 (for example, $r_0 w_1 r_0 \cdots r_0 w_{k_i} r_1$ is such a string);
- and so on.

We obtain, using all decompositions of w, m sequences:

$$\begin{array}{llllll} S_1 : & c_1^1, & c_2^1, & \ldots & c_n^1, & \ldots \\ S_2 : & c_1^2, & c_2^2, & \ldots & c_n^2, & \ldots \\ \ldots & & & & & \\ S_m : & c_1^m, & c_2^m, & \ldots & c_n^m, & \ldots \end{array}$$

Now, the activity of the algorithm $\mathcal{A}$ on the input $w \in T^*$ can be described as follows:

1. $\mathcal{A}$ computes all decompositions of w; let $d_1, \ldots, d_m$ ($m \geq 1$) be these decompositions;
2. $\mathcal{A}$ searchs the sequences $S_1, \ldots, S_m$ (as above) in the order
$$c_1^1, c_1^2, \ldots, c_1^m, c_2^1, c_2^2, \ldots, c_2^m, \ldots$$
3. for a string c_i^j ($i \geq 1$, $1 \leq j \leq m$) the algorithm $\mathcal{A}$ can effectively decide whether or not c_i^j describes a terminal computation of w in γ. If this is the case, then $\mathcal{A}$ halts with the answer "w is a member of $L(\gamma)$"; otherwise, $\mathcal{A}$ will continue the searching.

It is easy to see that $\mathcal{A}$ halts on the input w iff $w \in L(\gamma)$. We conclude that $L(\gamma) \in \mathcal{L}_0$ and so, $\mathbf{RL^f} \subseteq \mathcal{L}_0$. Combining this inclusion with the other one we obtain the theorem. □

Remark 3.2 *In our study only recursive sets of jumps have been considered. But, the construction in [18] showing that jumping Petri nets can simulate inhibitor nets needs only jumping nets in which the set R of jumps is even more restrictive than recursive. Namely, the simulation is possible with nets where the set $\{M' | (M, M') \in R\}$ is finite for each marking M. Hence, we could add this restriction to our definition for jumping Petri nets without affecting their computational power. This restriction would also allow a direct simulation of jumping Petri nets by non-deterministic Turing machines.*

Remark 3.3 *If we know that $[M_0\rangle_{\gamma,j}$ is recursive, we can replace $\mathbf{N}^S$ by $[M_0\rangle_{\gamma,j}$ in the definition of R' in the proof of Theorem 3.1. Consider now the case where $[M_0\rangle_{\gamma,j}$ is finite.*

Case 1 *R is finite. Then R' will be finite. Moreover, it is easy to see that the set $[M'_0\rangle_{\gamma',j}$ is finite. Denote by $\mathbf{RX}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}}$ ($\mathbf{RX}(\mathbf{fss})_{\mathbf{fin}}$, $\mathbf{RX}(\mathbf{fss})^{\lambda}_{\mathbf{fin}}$, resp.) the family of X-type languages generated by FJN's which have a* finite state space *(i.e., a finite set of j-reachable markings). We have then*

$$\mathbf{RX}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}} = \mathbf{RX}(\mathbf{fss})_{\mathbf{fin}} = \mathbf{RX}(\mathbf{fss})^{\lambda}_{\mathbf{fin}},\ \forall X \in \{P, L\}.$$

Case 2 *R is infinite. Then $[M'_0\rangle_{\gamma',j}$ is also finite. Denote by $\mathbf{RX}(\mathbf{fss})^{\mathbf{f}}$ ($\mathbf{RX}(\mathbf{fss})$, $\mathbf{RX}(\mathbf{fss})^{\lambda}$, resp.) the family of X-type languages generated by JN's which have a finite set of j-reachable markings. We have then*

$$\mathbf{RX}(\mathbf{fss})^{\mathbf{f}} = \mathbf{RX}(\mathbf{fss}) = \mathbf{RX}(\mathbf{fss})^{\lambda}, \quad \forall X \in \{P, L\}.$$

Moreover, in this case, only finitely many jumps of R are j-enabled from M_0 ($[M_0\rangle_{\gamma,j}$ is finite); hence $\mathbf{RX}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}} = \mathbf{RX}(\mathbf{fss})^{\mathbf{f}}$ and thus, for any $X \in \{P, L\}$ we have

$$\mathbf{RX}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}} = \mathbf{RX}(\mathbf{fss})_{\mathbf{fin}} = \mathbf{RX}(\mathbf{fss})^{\lambda}_{\mathbf{fin}} = \mathbf{RX}(\mathbf{fss})^{\mathbf{f}} = \mathbf{RX}(\mathbf{fss}) = \mathbf{RX}(\mathbf{fss})^{\lambda}.$$

There is a strong connection between these families of languages and the family of regular languages. Indeed, if we denote by $\mathbf{X}(\mathbf{fss})^{\mathbf{f}}$ ($\mathbf{X}(\mathbf{fss})$, $\mathbf{X}(\mathbf{fss})^{\lambda}$, resp.) the family of X-type languages generated by PN's which have a finite set of reachable markings, then we have

$$\mathcal{L}_3 \subseteq \mathbf{L}(\mathbf{fss}) \subseteq \mathbf{L}(\mathbf{fss})^{\lambda} \subseteq \mathbf{RL}(\mathbf{fss})^{\lambda}_{fin} = \mathbf{RL}(\mathbf{fss})^{f}_{fin} \subseteq \mathcal{L}_3.$$

The first inclusion follows from the fact that the transition graph of a deterministic finite automaton can be easily transformed into an $lmPTNf$ which has a finite set of reachable markings; the last inclusion follows from the fact that any $mFJPTNf$ which has a finite set of j-reachable markings can be transformed into a finite automaton with λ-moves whose set of states is the set of j-reachable markings and whose transition function is given in an obvious way (the λ-moves are given by R)

$$\delta(M, x) = \begin{cases} M', & \text{if } x \text{ is a transition and } M[x\rangle_{\Sigma} M' \\ \{M' | (M, M') \in R\}, & \text{if } x = \lambda. \end{cases}$$

Analogously we have

$$\mathcal{L}_{3,pref} \subseteq \mathbf{P}(\mathbf{fss}) \subseteq \mathbf{P}(\mathbf{fss})^{\lambda} \subseteq \mathbf{RP}(\mathbf{fss})^{\lambda}_{\mathbf{fin}} = \mathbf{RP}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}} \subseteq \mathcal{L}_{3,pref}.$$

Thus we have obtained

Theorem 3.3

(1) $\mathcal{L}_3 = \mathbf{L}(\mathbf{fss}) = \mathbf{L}(\mathbf{fss})^{\lambda} = \mathbf{RL}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}} = \mathbf{RL}(\mathbf{fss})_{\mathbf{fin}} = \mathbf{RL}(\mathbf{fss})^{\lambda}_{\mathbf{fin}} = \mathbf{RL}(\mathbf{fss})^{\mathbf{f}} = \mathbf{RL}(\mathbf{fss}) = \mathbf{RL}(\mathbf{fss})^{\lambda}$;

(2) $\mathcal{L}_{3,pref} = \mathbf{P}(\mathbf{fss}) = \mathbf{P}(\mathbf{fss})^{\lambda} = \mathbf{RP}(\mathbf{fss})^{\mathbf{f}}_{\mathbf{fin}} = \mathbf{RP}(\mathbf{fss})_{\mathbf{fin}} = \mathbf{RP}(\mathbf{fss})^{\lambda}_{\mathbf{fin}} = \mathbf{RP}(\mathbf{fss})^{\mathbf{f}} = \mathbf{RP}(\mathbf{fss}) = \mathbf{RP}(\mathbf{fss})^{\lambda}$;

(3) $\mathbf{X}(\mathbf{fss})^{\mathbf{f}} \subset \mathbf{X}(\mathbf{fss})$, *for any* $X \in \{P, L\}$.

Proof (1) and (2) directly follow from Remark 3.1. (3) follows from definitions and from the fact that $L_1 = \{baa\} \cup \{abb\} \notin \mathbf{L}^{\mathbf{f}}$ ([11]) and $L_2 = \{a^n | n \geq 0\} \cup \{b^n | n \geq 0\} \notin \mathbf{P}^{\mathbf{f}}$ ([17]). It is easy to see that $L_1 \in \mathbf{L}(\mathbf{fss})$ and $L_2 \in \mathbf{P}(\mathbf{fss})$. □

4 Characterization Results and Consequences

In this section we focus on finite jumping nets. We shall prove that any language $L \in \mathbf{RL}^{\mathbf{f}}_{\mathbf{fin}}$ ($\mathbf{RL}_{\mathbf{fin}}$, $\mathbf{RL}^{\lambda}_{\mathbf{fin}}$, resp.) can be represented as $L = \varphi(L')$, where L' is a regular language and φ is a substitution with λ-free languages. Similar results hold true for P-type jumping Petri net languages.

Theorem 4.1 *For any $L \in \mathbf{RL}^{\mathbf{f}}_{\mathbf{fin}}$ ($\mathbf{RL}_{\mathbf{fin}}$, $\mathbf{RL}^{\lambda}_{\mathbf{fin}}$, resp.) there exist a language $L' \in \mathcal{L}_3$ and a substitution with λ-free languages φ from $alph(L')$ into $\mathbf{L}^{\mathbf{f}}$ ($\mathbf{L}$, $\mathbf{L}^{\lambda}$, resp.) such that $L = \varphi(L')$.*

Proof Let $L \in \mathbf{RL}^{\mathbf{f}}_{\mathbf{fin}}$. There is an $mFJPTNf$ $\gamma = (\Sigma, R, M_0, \mathcal{M})$ such that $L = L(\gamma)$. We construct a finite automaton with λ-moves, $A = (Q, I, \delta, q_0, Q_f)$, as follows:

(i) $Q = \{M_0\} \cup dom(R) \cup cod(R) \cup \mathcal{M}$;

(ii) $I = \{a_{M',M} | M', M \in Q$ and M is reachable from M' in Σ by a non-empty sequence of transitions$\}$;

(iii) $\delta : Q \times (I \cup \{\lambda\}) \to Q$ is given by:

- $\delta(M', a_{M',M}) = \{M\}$ if $a_{M',M} \in I$,
- $\delta(M, \lambda) = \{M' | (M, M') \in R\}$,
- undefined, otherwise;

(iv) $q_0 = M_0$;

(v) $Q_f = \mathcal{M}$.

Let $L' = L(A)$ and $\varphi : alph(L') \to \mathbf{L}^{\mathbf{f}}$ given by $\varphi(a_{M',M}) = L(M', M) - \{\lambda\}$, where $L(M', M)$ is the language generated by the $mPTNf$ $(\Sigma, M', \{M\})$.

We have $L' \in \mathcal{L}_3$. Let us prove that $L = \varphi(L')$. First, $\lambda \in L$ iff $\lambda \in L'$ and hence $\lambda \in L$ iff $\lambda \in \varphi(L')$. Let now $w \in L$, $w \neq \lambda$. There is a decomposition of w, $w = w_1 \cdots w_{m+1}$, $m \geq 0$, such that

$$M_0 R^* M_0' [w_1\rangle_\Sigma M_1 R^+ M_1' \ldots R^+ M_m' [w_{m+1}\rangle_\Sigma M_{m+1} R^* M_{m+1}',$$

where $M_{m+1}' \in \mathcal{M}$ and M_i and M_i' are markings of γ and $w_i \neq \lambda$ for any $0 \leq i \leq m+1$.

The sequence $u = a_{M_0',M_1} a_{M_1',M_2} \cdots a_{M_m',M_{m+1}}$ determines a unique path, excepting λ-moves, from M_0 to M_{m+1}' in the automaton A. Hence $u \in L'$ (the computation in γ and the path in the automaton A are shown in Figure 4.1). For any i, $1 \leq i \leq m+1$, we have $w_i \in L(M_{i-1}', M_i) - \{\lambda\}$ which shows that $w \in \varphi(u)$, i.e. $w \in \varphi(L')$. Thus the inclusion $L \subseteq \varphi(L')$ is proved. The other inclusion can be proved analogously.

The case $L \in \mathbf{RL}_{\mathbf{fin}}$ can be simply settled by starting from the remark that if $L = L(\gamma)$, $\gamma = (\Sigma, R, M_0, \mathcal{M}, l)$, then $L = l(L(\gamma'))$, where $\gamma' = (\Sigma, R, M_0, \mathcal{M})$. Now, there exist a regular language L' and a substitution with λ-free languages ψ from $alph(L')$ into $\mathbf{L}^{\mathbf{f}}$ such that $L(\gamma') = \psi(L')$. Define $\varphi = l \circ \psi$ which is a substitution with λ-free languages. We have $L = \varphi(L')$.

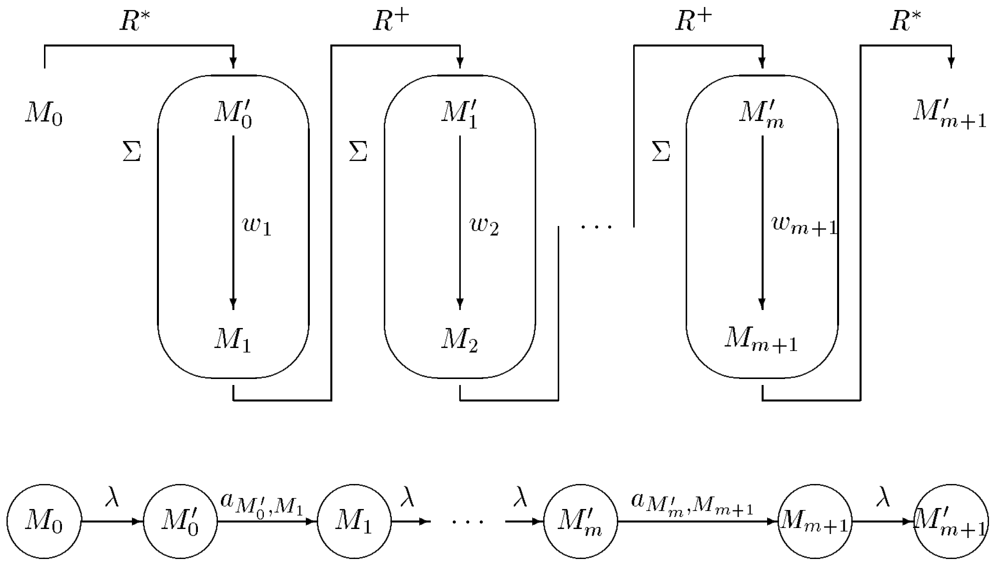

Figure 4.1

The previous idea does not work for the family $\mathbf{RL}^{\lambda}_{\mathbf{fin}}$ because l is an arbitrary labelling function and, for some a, $(l \circ \psi)(a)$ could contain λ. We modify the construction given in the case of $\mathbf{RL}^{\mathbf{f}}_{\mathbf{fin}}$ by setting

$$\varphi(a_{M',M}) = l(L(M', M)) - \{\lambda\},$$

for any $a_{M',M}$, and adding arcs $(\overline{M}, \overline{M}')$ labelled by λ to A whenever there exist in A the arcs $(\overline{M}, M')$ and $(M, \overline{M}')$ labelled by λ and (M', M) labelled by $a_{M',M}$ and $\lambda \in l(L(\Sigma, M', \{M\}))$ (since Petri net languages are recursive languages ([11]) we can effectively decide whether or not λ is in such a language). Figure 4.2 shows this construction. It is

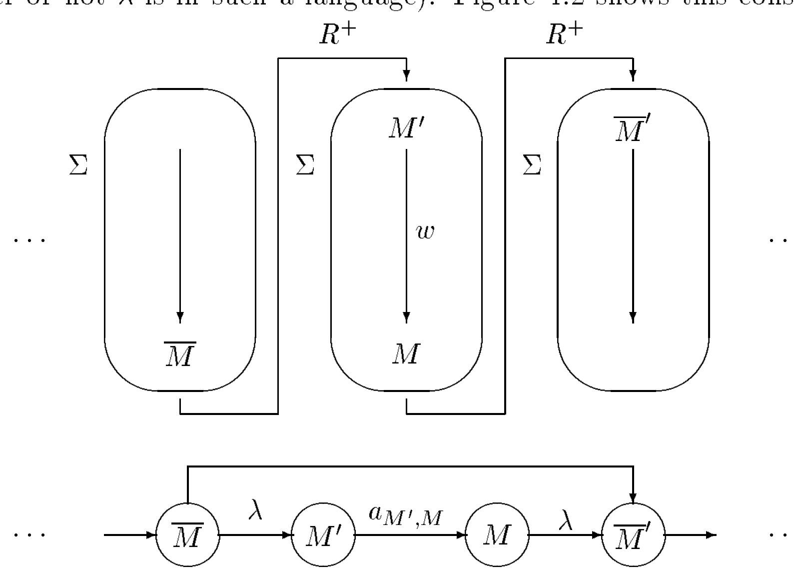

Figure 4.2 (The case $l(w) = \lambda$)

easy to see that the theorem holds also true in this case. □

The proof of Theorem 4.1 is effective. This fact permits us to show that terminal jumping Petri net languages are recursive.

Corollary 4.1 $\mathbf{RL}^{\lambda}_{\mathbf{fin}} \subseteq \mathcal{L}_{rec}$.

Proof We show that the membership problem for the family $\mathbf{RL}^{\lambda}_{\mathbf{fin}}$ is decidable. Let $\gamma = (\Sigma, R, M_0, \mathcal{M}, l)$ be an $l^{\lambda}mFJPTNf$, T the set of its transitions and V the range of l.

¿From Theorem 4.1 it follows that we can effectively compute a regular language L' (given by a finite automaton) and a substitution with λ-free languages $\varphi : alph(L') \rightarrow \mathbf{L}^{\lambda}$ such that $L(\gamma) = \varphi(L')$. Let $w \in V^*$. Since φ is a substitution with λ-free languages we have:

- $\lambda \in L(\gamma)$ iff $\lambda \in L'$;
- if $w \neq \lambda$, $w = a_1 \cdots a_n$ $(n \geq 1)$, then $w \in L(\gamma)$ iff there exist $b_1 \cdots b_m \in L'$ $(1 \leq m \leq n)$ and $u_i \in \varphi(b_i)$, $1 \leq i \leq m$, such that $w = u_1 \cdots u_m$.

Consequently, the membership problem for $L(\gamma)$ can be reduced to the membership problem for a regular language and for some arbitrary Petri net languages. Since Petri net languages are recursive ([11]) we conclude that the membership problem for $\mathbf{RL}^{\lambda}_{\mathbf{fin}}$ is decidable, and so $\mathbf{RL}^{\lambda}_{\mathbf{fin}} \subseteq \mathcal{L}_{rec}$. □

Corollary 4.2 $\mathbf{RL}_{\mathbf{fin}} \subseteq \mathcal{L}_1$.

Proof For any language $L \in \mathbf{RL}_{\mathbf{fin}}$ there exist a regular language L' and a substitution with λ-free languages φ from $alph(L')$ into $\mathbf{L}$ such that $L = \varphi(L')$. But $\mathbf{L} \subset \mathcal{L}_1$ ([11]) and $\mathcal{L}_1$ is closed under substitutions with λ-free languages, from which the theorem follows. □

The converse of Theorem 4.1 holds true for labelled and arbitrarily labelled jumping nets.

Theorem 4.2 *If $L \in \mathcal{L}_3$ and φ is a substitution from $alph(L)$ into $\mathbf{L}$ ($\mathbf{L}^{\lambda}$, resp.) then $\varphi(L) \in \mathbf{RL}_{\mathbf{fin}}$ ($\mathbf{RL}^{\lambda}_{\mathbf{fin}}$, resp.).*

Proof Let $L \in \mathcal{L}_3$ and $\varphi : alph(L) \rightarrow \mathbf{L}$. By Theorem 3.3 there is an $lmPTNf$ $\gamma = (\Sigma, M_0, \mathcal{M}, l)$ such that $L(\gamma) = L$, and $[M_0\rangle$ is finite. Let $alph(L) = \{a_1, \ldots, a_n\}$, $n \geq 1$, and $L_i = \varphi(a_i)$, $1 \leq i \leq n$. There exists an $lmPTNf$ $\gamma_i = (\Sigma_i, M_0^i, \mathcal{M}_i, l_i)$, $1 \leq i \leq n$, such that $L_i = L(\gamma_i)$ for each i.

Construct an $lmJPTNf$ $\gamma' = (\Sigma', R', M_0', \mathcal{M}', l')$ such that $\varphi(L) = L(\gamma')$, as follows. The nets $\Sigma, \Sigma_1, \ldots, \Sigma_n$ will be subnets of Σ' and initially they will be "blocked". When a transition t labelled by $l(t) = a_i$, $1 \leq i \leq n$, occurs in γ then in γ' the subnet Σ_i will be relieved (by means R') and a transition sequence w in γ_i can now occur in γ'. When a final marking is reached in Σ_i this subnet will be blocked again (by means of R').

Without loss of generality we may assume that $S_i \cap S_j = \emptyset$, $T_i \cap T_j = \emptyset$, $T \cap T_i = \emptyset$ and $S \cap S_i = \emptyset$ for any $i, j \in \{1, \ldots, n\}$, $i \neq j$. Now γ' is given by:

(i) $S' = S \cup \bigcup_{i=1}^{n} S_i \cup \{s_0, s_1, \ldots, s_n\}$, where $s_0, s_1, \ldots, s_n$ are new places. Any marking of γ' will be written in the form

$$M' = (M, \underbrace{\alpha_0}_{s_0}, M_1, \underbrace{\alpha_1}_{s_1}, \ldots, M_n, \underbrace{\alpha_n}_{s_n}),$$

where $M \in \mathbf{N}^S$ and $M_i \in \mathbf{N}^{S_i}$, $1 \leq i \leq n$;

(ii) $T' = T \cup \bigcup_{i=1}^{n} T_i$;

(iii) $F' = F \cup \bigcup_{i=1}^{n} F_i \cup \{(s_0, t) | t \in T\} \cup \{(s_i, t), (t, s_i) | t \in T_i,\ 1 \leq i \leq n\}$;

(iv) $W'(x,y) = \begin{cases} W(x,y), & \text{if } (x,y) \in F \\ W_i(x,y), & \text{if } (x,y) \in F_i \\ 1, & \text{otherwise;} \end{cases}$

(v) $R' = R^1 \cup R^2$, where
$R^1 = \{((M_1, 0, M_0^1, 0, \ldots, M_0^n, 0), (M_2, 0, M_0^1, 0, \ldots, M_0^i, 1, \ldots, M_0^n, 0)) |$
$1 \leq i \leq n$ and $(\exists t \in T : l(t) = a_i$ and $M_1 \in [M_0\rangle$ and $M_1[t\rangle M_2)\}$,
$R^2 = \{((M, 0, M_0^1, 0, \ldots, M_i, 1, \ldots, M_0^n, 0), (M, 0, M_0^1, 0, \ldots, M_0^i, 0, \ldots, M_0^n, 0)) |$
$M \in [M_0\rangle$ and $M_i \in \mathcal{M}_i$, $1 \leq i \leq n\}$;

(vi) $M_0' = (M_0, 0, M_0^1, 0, \ldots, M_0^n, 0)$;

(vii) $\mathcal{M}' = \{(M, 0, M_0^1, 0, \ldots, M_0^n, 0) | M \in \mathcal{M}\}$;

(viii) $l'(t) = \begin{cases} t & \text{if } t \in T \\ l_i(t) & \text{if } t \in T_i,\ 1 \leq i \leq n. \end{cases}$

Graphically, the net Σ' is shown in Figure 4.3. The relation R^1 is finite because $[M_0\rangle_\gamma$ is

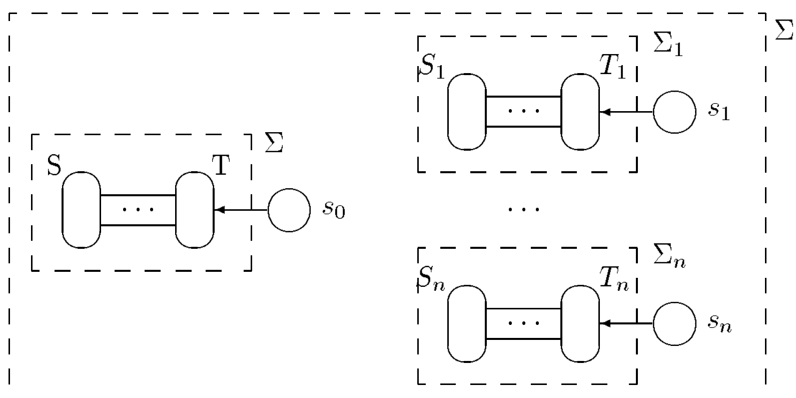

Figure 4.3

finite and R^2 is finite because the sets $\mathcal{M}_i$, $1 \leq i \leq n$, are finite. Hence R', is finite and γ' is an $lmFJPTNf$. The j-reachable markings of γ' are of the following types:

(1) $(M, 0, M_0^1, 0, \ldots, M_0^n, 0)$, where $M \in [M_0\rangle_\gamma$;

(2) $(M, 0, M_0^1, 0, \ldots, M^i, 1, \ldots, M_0^n, 0)$, where $M \in [M_0\rangle_\gamma$ and $M^i \in [M_0^i\rangle_{\gamma^i}$, $1 \le i \le n$.

No transition of γ' is enabled at an (1)-type marking. Only by jumps from R^1 we can pass from an (1)-type marking to a (2)-type marking; conversely only by jumps from R^2. It is not difficult to show that $L(\gamma') = \varphi(L)$ and hence, $\varphi(L) \in \mathbf{RL_{fin}}$.

A similar argument holds for the family $\mathbf{RL_{fin}^\lambda}$. $\square$

Corollary 4.3 *$L \in \mathbf{RL_{fin}}$ ($\mathbf{RL_{fin}^\lambda}$, resp.) iff there exist $L' \in \mathcal{L}_3$ and a substitution with λ-free languages $\varphi : alph(L) \to \mathbf{L}$ ($\mathbf{L}^\lambda$, resp.) such that $L = \varphi(L')$.*

A result similar to that in Theorem 4.1 holds for P-type jumping languages.

Theorem 4.3 *For any $L \in \mathbf{RP_{fin}^f}$ ($\mathbf{RP_{fin}}$, $\mathbf{RP_{fin}^\lambda}$, resp.) there exist a language $L' \in \mathcal{L}_{3,pref}$, a substitution with λ-free languages φ from $alph(L')$ into $\mathbf{L^f}$ ($\mathbf{L}$, $\mathbf{L}^\lambda$, resp.), and the languages P_0 and P_a, $a \in alph(L')$, such that*

$$L = P_0 \cup \bigcup_{a \in alph(L')} \varphi(\partial_a^r(L')\{a\})P_a$$

(∂^r denotes the right derivative). Moreover, the languages P_0 and P_a, $a \in alph(L')$, are finite unions of free P-type languages (P-type languages, arbitrary P-type languages, resp.).

Proof Let $\gamma = (\Sigma, R, M_0)$ be an $mFJPTN$ such that $L = L(\gamma)$. We construct a finite automaton with λ-moves, $A = (Q, I, \delta, q_0, Q_f)$, similar to that described in the proof of Theorem 4.1, excepting only that the sets of states and final states are $Q = \{M_0\} \cup dom(R) \cup cod(R)$ and $Q_f = Q$. Next we consider

- $L' = L(A)$ which is a prefix regular language;
- the substitution φ as in the proof of Theorem 4.1;
- $P_0 = \bigcup_{(M_0, M) \in R^*} P(\Sigma, M)$;
- $P_{a_{M',M}} = \bigcup_{(M, M'') \in R^+} P(\Sigma, M'')$, for any $a_{M',M} \in alph(L')$.

Now, let us prove the equality in theorem. Let $w \in L$.

If $w = \lambda$ or the computation induced by w contains a group of jumps only at the beginning ($M_0 R^* M_0'[w\rangle M$) then $w \in P_0$. Otherwise there is a decomposition of w, $w = w_1 \cdots w_{m+1}$, $m \ge 1$, such that

$$M_0 R^* M_0'[w_1\rangle M_1 R^+ M_1' \ldots [w_m\rangle M_m R^+ M_m'[w_{m+1}\rangle M \in \mathbf{N}^S,$$

where $w_i \ne \lambda$ for any $1 \le i \le m+1$.

The sequence $u = a_{M_0', M_1} a_{M_1', M_2} \ldots a_{M_{m-1}', M_m}$ determines a unique path (from M_0 to M_m) in the automaton A and hence $u \in L'$.

For any i, $1 \leq i \leq m$, we have $w_i \in L(M'_{i-1}, M_i) = \varphi(a_{M'_{i-1},M_i})$ which shows that $w_1 \cdots w_m \in \varphi(u)$, i.e. $w_1 \cdots w_m \in \varphi(u) \subseteq \varphi(\partial^r_{a(M'_{m-1},M_m)}(L')\{a_{M'_{m-1},M_m}\})$. But, it is clear that $w_{m+1} \in P_{a_{M'_{m-1},M_m}}$, and thus we obtain

$$w \in \varphi(\partial^r_{a_{M'_{m-1},M_m}}(L')\{a_{M'_{m-1},M_m}\})P_{a_{M'_{m-1},M_m}} \subseteq \bigcup_{a \in alph(L')} \varphi(\partial^r_a(L')\{a\})P_a.$$

The other inclusion can be proved analogously.

The case $L \in \mathbf{RL_{fin}}$ ($L \in \mathbf{RL^\lambda_{fin}}$, resp.) can be settled as in the proof of Theorem 4.1. We only mention that the languages P_0 and P_a are images by the labelling homomorphism l of finite unions of free P-type Petri net languages; that is, P_0 and P_a are finite unions of P-type Petri net languages (arbitrary P-type Petri net languages, resp.). □

Corollary 4.4 *For any $L \in \mathbf{RP_{fin}}$ ($\mathbf{RP^\lambda_{fin}}$, resp.) there exist a language $L' \in \mathcal{L}_{3,pref}$, a substitution with λ-free languages φ from $alph(L')$ into $\mathbf{L}$ ($\mathbf{L}^\lambda$, resp.), and P-type languages (arbitrary P-type languages, resp.) P_0 and P_a, $a \in alph(L')$, such that*

$$L = P_0 \cup \bigcup_{a \in alph(L')} \varphi(\partial^r_a(L')\{a\})P_a.$$

Proof The family of P-type languages (arbitrary P-type languages, resp.) is closed under union ([11]). □

Remark 4.1 *The idea in the proof of Theorem 4.2 cannot be used for the family $\mathbf{RL^f_{fin}}$ because it is not generally true that $T_i \cap T_j = \emptyset$ for any $i \neq j$, and it cannot be used for P-type languages because the relation R^2 is, in general, infinite.*

Using similar constructions as for classical Petri net languages it is easy to prove that the families $\mathbf{RL_{fin}}$ and $\mathbf{RL^\lambda_{fin}}$ are closed under finite union and catenation (one can use also the power of jumping relation in correlation with final markings). Then we have:

Corollary 4.5 $\mathbf{RP_{fin}} \subseteq \mathbf{RL_{fin}}$ *and* $\mathbf{RP^\lambda_{fin}} \subseteq \mathbf{RL^\lambda_{fin}}$.

Proof We will prove only the inclusion $\mathbf{RP_{fin}} \subseteq \mathbf{RL_{fin}}$, the other one being similar to this one. If $L \in \mathbf{RP_{fin}}$ then L can be written as in Theorem 4.3

$$L = P_0 \cup \bigcup_{a \in alph(L')} \varphi(\partial^r_a(L')\{a\})P_a.$$

$\partial^r_a(L')\{a\}$ is a regular language and so, by Theorem 4.2, we have $\varphi(\partial^r_a(L')\{a\}) \in \mathbf{RL_{fin}}$ for any $a \in alph(L')$.

It is well-known that P-type Petri net languages are also L-type Petri net languages ([11]), that is $\mathbf{P} \subseteq \mathbf{L}$, and so $P_0, P_a \in \mathbf{L} \subseteq \mathbf{RL_{fin}}$. Using the remark above concerning the closedness of $\mathbf{RL_{fin}}$ under finite union and catenation, we obtain $L \in \mathbf{RL_{fin}}$. □

For P-type languages the following pumping lemma holds true.

Theorem 4.4 *For any $L \in \mathbf{RP}^{\lambda}_{\mathbf{fin}}$ there is a number $k \in \mathbf{N}$ such that for each word $w \in L$, if $|w| \geq k$ then there is a prefix w' of w which has a decomposition $w' = xyz$ such that:*

(i) $|y| \geq 1$,

(ii) $xy^{m+1}z \in L$, *for any* $m \geq 0$.

Proof Let $\gamma = (\Sigma, R, M_0, l)$ be an $l^{\lambda}mFJPTN$ such that $L = P(\gamma)$. Consider the automaton A, the substitution φ and the languages L', P_0 and P_a $(a \in I)$ as in the proof of Theorem 4.3 (the languages P_0 and P_a, $a \in I$, are arbitrary P-type Petri net languages). We have

$$L = P_0 \cup \bigcup_{a \in alph(L')} \varphi(\partial^r_a(L')\{a\})P_a.$$

Let k_1, k_0, k_a $(a \in I)$ be the constants from the pumping lemmata for the regular language L' ([10]) and for the arbitrary P-type Petri net languages P_0 and P_a, $a \in I$ ([4]). Consider $k_2 = max\{k_0, k_a | a \in I\}$ and $k = k_1 k_2$. We shall prove that the number k satisfies the theorem.

Let $w \in L$ such that $|w| \geq k$. If $w \in P_0$ then we apply the pumping lemma for w with respect to P_0 and we obtain the theorem, with $w' = w$. Otherwise, there is a word $u = a_1 \cdots a_s \in L'$ such that $w \in \varphi(u)P_{a_s}$. We have to consider two cases.

Case 1 $s \geq k_1$. From the pumping lemma for regular languages, u has a decomposition $u = u_1 u_2 u_3$ such that $|u_2| \geq 1$ and $u_1 u_2^i u_3 \in L'$ for any $i \geq 0$. Since $w \in \varphi(u)P_{a_s} = \varphi(u_1)\varphi(u_2)\varphi(u_3)P_{a_s}$, there exist $x \in \varphi(u_1)$, $y \in \varphi(u_2)$, $z \in \varphi(u_3)$ and $v \in P_{a_s}$ such that $w = xyzv$. φ being a substitution with λ-free languages it follows that $|y| \geq 1$.

From $u_1 u_2^i u_3 \in L'$ it follows that $\varphi(u_1)[\varphi(u_2)]^i \varphi(u_3)P_{a_s} \subseteq L$ for any $i \geq 0$. Hence, $xy^i zv \in L$ for any $i \geq 0$, and the theorem is satisfied with $w' = w$.

Case 2 $s < k_1$. From $w \in \varphi(a_1 \cdots a_s)P_{a_s}$ it follows that there exist $w_j \in \varphi(a_j)$, $1 \leq j \leq s$, and $w_{s+1} \in P_{a_s}$ such that $w = w_1 \cdots w_s w_{s+1}$. Since $|w| \geq k = k_1 k_2$ and $|w| = |w_1| + \ldots + |w_s| + |w_{s+1}|$ and $s < k_1$, there is $j \in \{1, \ldots, s+1\}$ such that $|w_j| > k_2 \geq k_{a_j}$.

If $j = s+1$ then we apply the pumping lemma for the language P_{a_s} and we obtain the theorem with $w' = w$.

If $j = 1$ then it is clear that $L(M'_0, M_1) \subseteq P_0$, where $a_1 = a_{M'_0, M_1}$ and $M_0 R^* M'_0 [w_1\rangle M_1$. Thus $w_1 \in P_0$, and now we have to apply the pumping lemma for the word w_1 with respect to P_0. Then $w_1 = x_1 y_1 z_1$ with $|y_1| \geq 1$ and $x_1 y_1^i z_1 \in P_0$ for any $i \geq 1$. Consider $w' = w_1$, $x = x_1$, $y = y_1$ and $z = z_1$ and the theorem is satisfied.

If $1 < j < s+1$ then let us suppose that $a_{j-1} = a_{M'_{j-2}, M_{j-1}}$ and $a_j = a_{M'_{j-1}, M_j}$. Then, $\varphi(a_j) = L(M'_{j-1}, M_j) = L(\Sigma, M'_{j-1}, \{M_j\})$ and $P_{a_{j-1}} = \bigcup_{(M_{j-1}, M) \in R^+} P(\Sigma, M)$. Since $M_{j-1} R^+ M'_{j-1}$ it follows that $\varphi(a_j) \subseteq P_{a_{j-1}}$ and $\varphi(a_1) \cdots \varphi(a_{j-1}) P_{a_{j-1}} \subseteq L$. Thus $w_1 \cdots w_{j-1} w_j \in \varphi(a_1) \cdots \varphi(a_{j-1}) P_{a_{j-1}}$, and now we have to apply the pumping lemma for the word w_j with respect to $P_{a_{j-1}}$. Then, $w_j = x_j y_j z_j$ with $|y_j| \geq 1$ and $x_j y_j^i z_j \in P_{a_{j-1}}$ for any $i \geq 1$. Consider $w' = w_1 \cdots w_j$, $x = w_1 \cdots w_{j-1} x_j$, $y = y_j$ and $z = z_j$ and the theorem is satisfied in this case too. □

5 Comparisons Between Families of Languages

Any family of L-type jumping Petri net languages is closed under "*" (the net jumps from any final marking to the initial marking). This proof also works for the family $\mathbf{RP^f}$, $\mathbf{RP}$, $\mathbf{RP^\lambda}$, but not for $\mathbf{RP^f_{fin}}$, $\mathbf{RP_{fin}}$, $\mathbf{RP^\lambda_{fin}}$. The closure under "*" of the family $\mathbf{RP^\lambda_{fin}}$ can be proved using the following idea. At any reachable marking of the net some λ-transitions are enabled. These transitions will reset the current marking to the zero-marking $\mathbf{0}$ (all the components are 0) and then, a jump from $\mathbf{0}$ to the initial marking will restart the net.

Theorem 5.1 *The family* $\mathbf{RP^\lambda_{fin}}$ *is closed under "*".*

Proof Let $\gamma = \{\Sigma, R, M_0, l\}$ be an $l^\lambda mFJPTN$. Consider the net Σ' as described in Figure 5.1 together with its labelling l'. For any marking M of Σ denote by M' the marking of

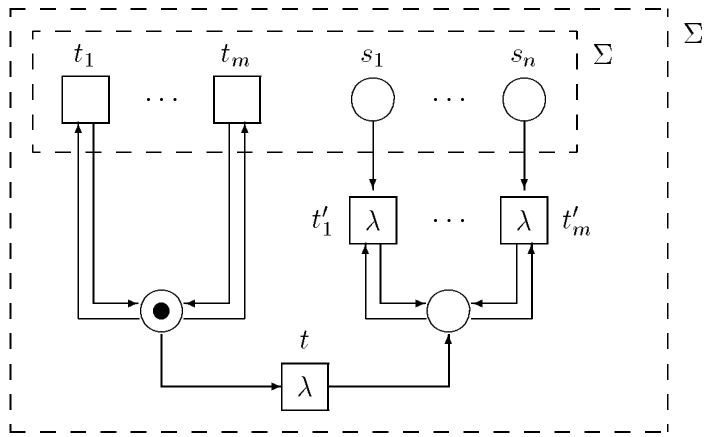

Figure 5.1

Σ' given by $M' = (M, \underbrace{1}_{s_0}, \underbrace{0}_{s'_0})$. Consider the binary relation $R' = \{(M'_1, M'_2) | (M_1, M_2) \in R\} \cup \{((0, \ldots, 0, 0, 1), M'_0)\}$ and then, the net $\gamma' = (\Sigma', R', M'_0, l')$ satisfies $P(\gamma') = (P(\gamma))^*$. □

The non-closure under Kleene star of the families of Petri net languages leads us to the following results:

Theorem 5.2

(1) $\mathbf{L^f} \subset \mathbf{RL^f_{fin}}$, $\mathbf{L} \subset \mathbf{RL_{fin}}$, $\mathbf{L^\lambda} \subset \mathbf{RL^\lambda_{fin}}$;

(2) $\mathbf{P^\lambda} \subset \mathbf{RP^\lambda_{fin}}$.

Theorem 5.3 $\mathcal{L}_3 \subset \mathbf{RL^f_{fin}}$.

Proof The inclusion follows from the fact that $\mathcal{L}_3 = \mathbf{RL(fss)^f_{fin}} \subseteq \mathbf{RL^f_{fin}}$ and the strict inclusion follows from the facts that $\mathcal{L}_3 \cup \mathbf{L}^f \subseteq \mathbf{RL}^f_{fin}$ and the families $\mathcal{L}_3$ and $\mathbf{L}^f$ are incomparable ([11]). □

Using the results from Section 4 we can prove:

Theorem 5.4 $L = \{a^n b^n \mid n \geq 0\} \notin \mathbf{RL^f_{fin}}$.

Proof For the sake of contradiction suppose that $L \in \mathbf{RL^f_{fin}}$. Then there exist a regular language L' and a substitution with λ-free languages $\varphi : alph(L') \to \mathbf{L^f}$ such that $L = \varphi(L')$.

Case 1 L' is infinite. There exist $u \in L'$ and a decomposition of u, $u = u_1 u_2 u_3$ such that $|u_2| \geq 1$ and $u_1 u_2^i u_3 \in L'$ for any $i \geq 0$ (the pumping lemma for regular languages).

Since $\varphi(u_1 u_2^i u_3) = \varphi(u_1)[\varphi(u_2)]^i \varphi(u_3) \subseteq L$ and $u_2 \neq \lambda$, it follows that there exist $w_1 \in \varphi(u_1)$, $w_2 \in \varphi(u_2)$ and $w_3 \in \varphi(u_3)$ such that $w_1 w_2^i w_3 \in L$ for any $i \geq 0$. It is easy to see that no matter how w_1, w_2, w_3 ($w_2 \neq \lambda$) are chosen we cannot have $w_1 w_2^i w_3 \in L$ for any $i \geq 0$.

Case 2 L' is finite. If so, let $L' = \{u_1, \ldots, u_k\}$, $k \geq 1$. Since L is infinite, there exists $j \in \{1, \ldots, k\}$ such that $\varphi(u_j)$ is infinite. Let $u_j = a_1 \ldots a_{m_j}$, $m_j \geq 1$, and $\varphi(u_j) = \{a^{i_1} b^{i_1}, a^{i_2} b^{i_2}, \ldots\}$, where $0 \leq i_1 < i_2 < \ldots$. Then there is $i \in \{1, \ldots, m_j\}$ such that $\varphi(a_i)$ is infinite. We have to consider now the next cases.

If $\varphi(a_i) = \{a^{\alpha_1}, a^{\alpha_2}, \ldots\}$, where $0 \leq \alpha_1 < \alpha_1 < \ldots$, then it is easy to see that no matter how the words in $\varphi(a_i)$ are catenated to the left or to the right we obtain also other words than those in $\varphi(u_j)$. Similar reason tells that $\varphi(a_i)$ cannot be $\{b^{\beta_1}, b^{\beta_2}, \ldots\}$, where $0 \leq \beta_1 < \beta_2 < \ldots$.

As any subset of $\varphi(u_j)$ of cardinality at least two is not a member of $\mathbf{L^f}$, the only case which remains to be considered is $\varphi(a_i) = \{a^{\alpha_1} b^{\beta_1}, a^{\alpha_2} b^{\beta_2}, \ldots\}$, where α's and β's are natural numbers and there is n such that $\alpha_n \neq \beta_n$. There is also p, $p \neq n$, such that either $\alpha_n \neq \alpha_p$ or $\beta_n \neq \beta_p$. A straightforward analysis shows us that no matter how the language $\varphi(a_i)$ is catenated to the left or to the right we we obtain also other words than those in $\varphi(u_j)$.

In both cases we have derived a contradiction and hence $L \notin \mathbf{RL^f_{fin}}$. $\Box$

Corollary 5.1 $\{a^n b^n \mid n \geq 0\} \in \mathbf{L} - \mathbf{RL^f_{fin}}$.

Corollary 5.2 *The families* $\mathbf{RL^f_{fin}}$ *and* $\mathcal{L}_2$ *are incomparable.*

Proof $\{a^n b^n \mid n \geq 0\} \in \mathcal{L}_2 - \mathbf{RL^f_{fin}}$ and $\{a^n d b^n e c^n \mid n \geq 1\} \in \mathbf{RL^f_{fin}} - \mathcal{L}_2$. $\Box$

Corollary 5.3 $\mathbf{RL^f_{fin}} \subset \mathbf{RL_{fin}}$.

Proof $\{a^n b^n \mid n \geq 0\} \in \mathbf{RL_{fin}}$ and $\{a^n b^n \mid n \leq 0\} \notin \mathbf{RL^f_{fin}}$. $\Box$

6 Finite Jumping Nets and Global Inhibitor Nets

We establish a connection between finite jumping nets and a subclass of inhibitor nets, global inhibitor nets. We recall that an inhibitor net ([9]) is a pair $\gamma = (\Sigma, I)$, where Σ is a Petri net and $I \subseteq S \times T$ such that $I \cap F = \emptyset$.

In an inhibitor net γ the transition t is *i-enabled* at a marking M, abbreviated $M[t\rangle_{\gamma,i}$, iff $t^- \leq M$ and $M(s) = 0$ for any $s \in \{s \in S | (s,t) \in I\}$. If $M[t\rangle_{\gamma,i}$ then t may occur yielding a new marking M', abbreviated $M[t\rangle_{\gamma,i} M'$, given by $M' = M + \Delta t$. As we can see, an inhibitor net has the capability to perform zero-tests on some places.

A *global inhibitor net* is defined as an inhibitor net performing zero-tests on all places, that is

$$(\forall t \in T)((\exists s \in S)((s,t) \in I) \quad \Rightarrow \quad (\forall s' \in S)((s',t) \in I)).$$

Now we show that *FJPTN*'s can be simulated by global inhibitor nets. Let $\gamma = (\Sigma, R, M_0, l)$ be an *FJPTN* with only one jump, $R = \{(M, M')\}$. Construct the following inhibitor net (the net is shown in Figure 6.1 and the relation I is given by $I = \{(s,t')|s \text{ is a place}\}$). It is clear that t' performs a zero-test on all places and so this net is

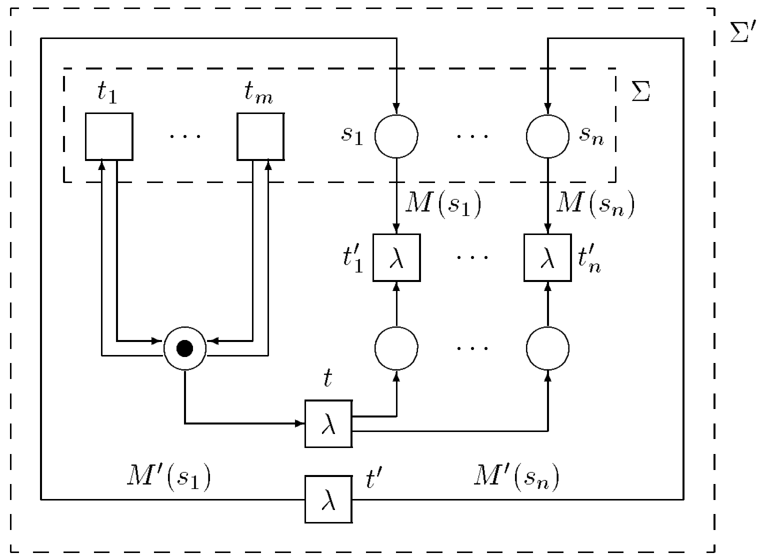

Figure 6.1

a global inhibitor net. Its activity can be described as follows:

- the transition t blocks Σ and then the transitions $t'_1, \ldots, t'_m$ check whether or not the current marking covers M (if all t'_i can occur then the current marking covers M). The zero-test performed by t' checks when the current marking is exactly M (t' can occur iff no token is in the net). If this is the case the marking M' is set for Σ.

The above construction can be easily generalized to an *FJPTN* with arbitrarily many jumps.

Now we show that any arbitrarily labelled global inhibitor net can be simulated by an *FJPTN*. Indeed, let $\gamma = (\Sigma, I, M_0, l)$ be such an inhibitor net. Assume $I = \{(s,t)|s \in S\}$, where t is a fixed transition.

If $l(t) = \lambda$ then we can simulate the extent of change caused by the occurrence of t using the jump $(\mathbf{0}, M)$, where $M(s) = W(t,s)$ for all $s \in S$ (we recall that $I \cap F = \emptyset$, that is $W(s,t) = 0$ for all $s \in S$).

If $l(t) = a \neq \lambda$ then we simulate the extent of change caused by t using the net shown in Figure 6.2 and the jump $\{((\mathbf{0}, 1, 0), (M, 0, 1))\}$ (M is as above). By this jump the net Σ will be blocked; it is relieved after an occurrence of the transition t labelled by a (t being a new transition).

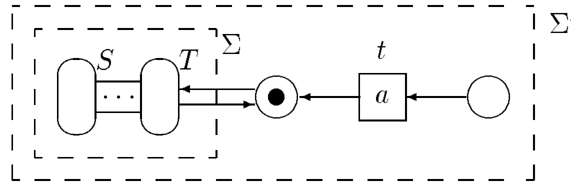

Figure 6.2

The generalization to an arbitrary global inhibitor net is straightforward.

Final Remarks

We consider that the extension of Petri nets allowing a finite number of jumps is quite reasonable: on the one hand such nets have the basic decision problems decidable and, on the other hand a finite number of jumps strictly increase the power of the nets.

We close with some important open problems.

P1. Petri nets may be considered as being jumping nets where the set of jumps is empty. Therefore, the construction given in the proof of Theorem 3.1 works in this case as well, and it shows us how we can replace (λ-)labelling functions of Petri nets by sets of jumps. As a conclusion, (λ-)labelled Petri nets are unlabelled jumping nets (via the interleaving semantics). The natural question now is the following: which class of JN corresponds exactly to (λ-)labelled Petri nets?

P2. Are the inclusions $\mathbf{RL_{fin}} \subseteq \mathcal{L}_1$, $\mathbf{RL^{\lambda}_{fin}} \subseteq \mathcal{L}_{rec}$, $\mathbf{RL_{fin}} \subseteq \mathbf{RL^{\lambda}_{fin}}$, $\mathbf{RP_{fin}} \subseteq \mathbf{RL_{fin}}$, $\mathbf{RP^{\lambda}_{fin}} \subseteq \mathbf{RL^{\lambda}_{fin}}$ proper or not?

P3. Are the families $\mathbf{RP^{f}_{fin}}$ and $\mathbf{RP_{fin}}$ closed under "*"?

P4. Define $\mathbf{RX^{f}_{k}}$ ($\mathbf{RX_{k}}$, $\mathbf{RX^{\lambda}_{k}}$, resp.) as being the family of X-type languages generated by jumping nets having at most k jumps ($k \geq 0$), that is $|R| \leq k$. We have

$$\begin{array}{lllllll}
\mathbf{X^{f}} & \subseteq & \mathbf{RX^{f}_{k}} & \subseteq & \mathbf{RX^{f}_{k+1}} & \subseteq & \mathbf{RX^{f}_{fin}} \\
\mathbf{X} & \subseteq & \mathbf{RX_{k}} & \subseteq & \mathbf{RX_{k+1}} & \subseteq & \mathbf{RX_{fin}} \\
\mathbf{X^{\lambda}} & \subseteq & \mathbf{RX^{\lambda}_{k}} & \subseteq & \mathbf{RX^{\lambda}_{k+1}} & \subseteq & \mathbf{RX^{\lambda}_{fin}}
\end{array}$$

for all $k \geq 1$ and $X \in \{P, L\}$.

Does this restriction define proper hierarchies of jumping Petri net languages?

P5. What about the connection between $FJPTN$'s and global inhibitor nets in the case we do not allow λ-transitions?

References

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

[2] H.D. Burkhard: *On priorities of parallelism; Petri nets under the maximum firing strategy*, Int. Conf. LOGLAN 77, Poznan, 1980.

[3] H.D. Burkhard: *The Maximum Firing Strategy in Petri Nets Gives More Power*, ICS-PAS Report 441, Warszaw, 24-2, 26, 1980.

[4] H.D. Burkhard: *Two pumping lemmata for Petri nets*, Journal of Information Processing and Cybernetics EIK, vol. 17, no. 7, 1981, 349 – 362.

[5] H.D. Burkhard: *Ordered firing in Petri nets*, Journal of Information Processing and Cybernetics EIK, vol. 17, no. 2/3, 1981, 71 – 86.

[6] H.D. Burkhard: *Control of Petri Nets by Finite Automata*, Preprint 26, Sektion Mathematik, Humboldt-Universität, Berlin, 1982.

[7] H.D. Burkhard: *What Gives Petri Nets More Computational Power*, Preprint 45, Sektion Mathematik, Humboldt-Universität, Berlin, 1982.

[8] H.J.M. Goeman, L.P.J. Groenwegen, H.C.M. Kleijn, G.Rozenberg: *Constrained Petri nets (Part I, II)*, Fundamenta Informaticae, vol. 6, no. 1, 1983.

[9] M. Hack: *Petri Net Languages*, CSG Memo 124, Project MAC, MIT, 1975.

[10] J.E. Hopcroft, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Mass., 1979.

[11] M. Jantzen: *Language Theory of Petri Nets*, Advances in Petri Nets 1986, Part I, LNCS 254, Springer-Verlag, 1987, 397–412.

[12] T. Jucan, C. Masalagiu, F.L. Ţiplea: *Relation Based Controlled Petri Nets*, Scientific Annals of the "Al. I. Cuza" University, Section Informatics, Tom 2, 1995.

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

[14] W. Reisig: *Place Transition Systems*, Advances in Petri Nets 1986, Part I, LNCS 254, Springer-Verlag, 1987, 117–141.

[15] G. Rozenberg, R. Verraedt: *Restricting the in-out structure of graph of Petri nets*, Fundamenta Informaticae VII.2, 1984, 151–189.

[16] F.L. Ţiplea, T. Jucan, C. Masalagiu: *Conditional Petri net languages*, J. Inf. Process. Cybern. EIK, vol. 27, no. 1, 1991, 55 – 66.

[17] F.L. Ţiplea: *Selective Petri net languages*, Intern. J. Computer Math., vol. 43, no. 1-2, 1992, 61 – 80.

[18] F.L. Ţiplea, T. Jucan: *Jumping Petri Nets*, Foundations of Computing and Decision Sciences, vol. 19, no. 4, 1994, 319 – 332.

[19] F.L. Ţiplea: *On Conditional Grammars and Conditional Petri Nets*, in: Mathematical Aspects of Natural and Formal Languages (Gh. Păun, ed.), World Scientific, Singapore, 1995, 431 – 455.

[20] T. Ushio: *On controllability of controlled Petri nets*, Control Theory and Advanced Technology, vol. 5, no. 3, 1989, 265 – 277.

[21] R. Valk: *Self-modifying nets, a natural extension of Petri nets*, Fifth Colloquium "Automata, Languages and Programming", 1978, LNCS 62, Springer-Verlag, 1978, 464–476.

[22] R. Valk: *On the computational power of extended Petri nets*, Proc. of the 7th Symposium "Mathematical Foundations of Computer Science", 1978, LNCS 64, Springer-Verlag, 1978, 526–535.