Journal of Universal Computer Science, vol. 8, no. 2(2002), 260-269
submitted: 15/9/02, accepted: 4/2/02, appeared: 28/2/02 © J.UCS

Some Remarks on Codes Defined by Petri Nets[1]

Masami Ito
(Department of Mathematics, Sangyo University Kyoto, Kyoto 603–8555, Japan
Email: ito@ksuvx0.kyoto-su.ac.jp)

Jürgen Dassow
(Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120,
D–39016 Magdeburg, Germany
Email: dassow@iws.cs.uni-magdeburg.de)

Ralf Stiebe
(Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120,
D–39016 Magdeburg, Germany
Email: stiebe@iws.cs.uni-magdeburg.de)

Abstract: With any Petri net we associated its CPN language which consists of all sequences of transitions which reach a marking with an empty place whereas all proper prefixes of the sequence lead to positive markings.
We prove that any CPN language can be accepted by a partially blind multicounter machine, and that any partially blind multicounter language is the morphic image of some CPN language. As a corollary we obtain the decidability of membership, emptiness and finiteness problem for CPN languages. We characterize the very strictly bounded regular languages, which are CPN languages, and give a condition for a Petri net, which ensures that its generated language is regular. We give a dense CPN language and prove that no dense regular language is a CPN language.

Key Words: Petri nets, codes, formal languages

Category: F. 4.2, F. 4.3

1 Introduction

Let $D = (P, T, \delta, \mu_0)$ be a Petri net where P is the set of places, T is the set of transitions, δ is the transition function and μ_0 is the initial marking. δ can be given by the values $\#(p, I(t))$ and $\#(p, O(t))$ which give the numbers of tokens taken from and given to the place p, respectively, if the transition t fires. By $\pi_p(\mu)$ we denote the number of tokens at place $p \in P$ of the marking μ.

The languages of Petri nets are formed by sequences of firing transitions. We are interested in the CPN[2] language of a Petri net which was introduced by Tanaka in [7] and is defined as the set of all sequences $u \in T^*$ such that there is a place $p \in P$ with $\pi_p(\delta(\mu_0, u) = 0$ and $\pi_q(\mu_0, v) > 0$ for all $q \in P$ and all proper prefixes v of u. Intuitively, we take those sequences in the language which lead to a marking with at least one empty place and where all intermediate markings

[1] C. S. Calude, K. Salomaa, S. Yu (eds.). *Advances and Trends in Automata and Formal Languages. A Collection of Papers in Honour of the 60th Birthday of Helmut Jürgensen.*
[2] C, P and N stand for code, Petri and net.

have no empty places. By $\mathcal{L}(D)$ we denote the CPN language generated by D. By *CPN* we denote the family of all CPN languages generated by Petri nets.

CPN languages are of interest because they are prefix codes. This follows from the fact that we take sequences where an empty place occurs for the first time.

In [7] and [4] CPN languages have been investigated with respect to properties which are of interest from the point of coding theory. For instance, if C^n with $n \geq 2$ is a finite maximal prefix code and a CPN language, then this holds for any C^k with $1 \leq k \leq n$, too, and C is a full uniform code.

In this paper, we study relations of the language family *CPN* to other (classical) language families. In [4] it was shown that

- *CPN* is a proper subclass of the family of context-sensitive languages,
- *CPN* contains non-context-free languages, and
- there are finite languages which do not belong to *CPN*.

From this we obtain that *CPN* is incomparable with the families of regular and context-free languages, respectively.

We improve the upper bound by showing that *CPN* is a proper subclass of the family $PBLIND(n)$ of languages accepted by quasirealtime blind multicounter machines. As a corollary we obtain the decidability of membership, emptiness and finiteness problems for CPN languages. On the other hand, we can also characterize $PBLIND(n)$ by means of CPN languages.

Furthermore, we study in more detail the relation to regular languages. For a special class of regular languages, the so-called very strictly bounded languages, we give a characterization of the CPN languages. Furthermore, we give a condition which ensures that the generated language of a Petri net is regular.

Finally, we show that *CPN* contains dense languages, but no regular dense languages, where density of $L \subseteq T^*$ is defined as the property that, for any $u \in T^*$, there are words x and y with $xuy \in L$.

We assume that the reader is familiar with the basic concepts of formal language theory (see [6]) and Petri nets (see [5]). We recall here some notations.

A deterministic finite automaton is specified as a quintuple $\mathcal{A} = (Z, X, z_0, F, \delta)$ where X is the input alphabet, Z is the set of states, z_0 is the initial state, F is the set of accepting states and $\delta : Z \times X \to Z$ is the transition function. $T(\mathcal{A})$ denotes the set of words accepted by $\mathcal{A}$. By REG we denote the family of regular languages (i.e., the family of languages acceptable by deterministic finite automata.

Partially blind counter machines, introduced by Greibach [2], are one-way multicounter machines where zero tests of the counters is not possible during a computation, but where a computation fails whenever a counter is below zero. Formally, a *partially blind k-counter machine* is a construct $\mathcal{A} = (Z, X, z_0, F, \delta)$, where Z, X, z_0 and F are defined as for a finite automaton and $\delta \subset Z \times (X \cup \{\lambda\}) \times Z \times \mathbb{Z}^k$ is a finite transition relation. An *instantaneous description* (ID) of $\mathcal{A}$ is a $(k+2)$-tuple $(z, w, n_1, \ldots, n_k)$ with $z \in Z$, $w \in X^*$ and $n_1, \ldots, n_k \in \mathbb{N}$. If $(z, a, z', m_1, \ldots, m_k) \in \delta$, $(z, aw, n_1, \ldots, n_k)$ is an ID and $n_i + m_i \geq 0$ for

$1 \leq i \leq k$, then we write $(z, aw, n_1, \ldots, n_k) \vdash (z', w, n_1 + m_1, \ldots, n_k + m_k)$. This changing of the ID is called a *step*. Moreover, if $a = \lambda$, the step is called an λ-*step*. $\mathcal{A}$ is called quasirealtime of delay d if $(z, \lambda, n_1, \ldots, n_k) \vdash^r (z', \lambda, n_1', \ldots, n_k')$ implies $r \leq d$. The language accepted by $\mathcal{A}$ is

$$L(\mathcal{A}) = \{w \in X^* \mid (z_0, w, 0, \ldots, 0) \vdash^* (q, \lambda, 0, \ldots, 0), \text{ for some } q \in F\},$$

where $\vdash^*$ is the transitive and reflexive closure of $\vdash$. The family of languages accepted by (quasirealtime) partially blind counter machines is denoted by *PBLIND* (*PBLIND*(n)).

For an n-dimensional vector $\mu = (a_1, a_2, \ldots, a_n)$, we set $|\mu| = \sum_{i=1}^{n} a_i$. If μ is a marking of a Petri net, then $|\mu|$ is the total number of tokens in the net.

2 CPN Languages and Partially Blind Counter Machines

Greibach [2] gave a characterization of *PBLIND*(n) by means of a class of Petri net languages. More specifically, a *Petri net machine* is a quintuple $\mathcal{A} = (P, T, \delta, \mu_0, F)$, where (P, T, δ, μ_0) is a Petri net and $F \subseteq P$ is a set of accepting places. The computation sequence set (CSS language) accepted by $\mathcal{A}$ consists of all transition sequences that transform μ_0 to a marking μ with $\pi_q(\mu) = 1$, for some $q \in F$, and $\pi_p(\mu) = 0$, for $p \neq q$. The family of CSS languages accepted by Petri net machines is denoted by *CSS*.

Lemma 1 (Greibach [2]). *1. CSS $\subseteq$ PBLIND(n).*

2. Any language in PBLIND(n) is the projection of a language in CSS. [3] □

In what follows, we will relate CPN and CSS languages, and thus the families *CPN* and *PBLIND*(n).

Lemma 2. *For any $L \in CSS$, $L \subseteq T^*$, the language $L\$$, $\$ \notin T$, is in CPN.*

Proof. Consider a Petri net machine $\mathcal{M} = (P, T, \delta, \mu_0, F)$. We construct the Petri net $D = (P \cup \{sum, run\}, T' \cup \{e\}, \delta', \mu_0')$, where

- $sum, run \notin P$ are additional positions,
- T' is a disjoint copy of T, the copy of $t \in T$ is denoted by t', $e \notin T \cup T'$ is an additional transition,
- $\pi_p(\mu_0') = 2\pi_p(\mu_0) + 1$, for $p \in P$,
  $\pi_{sum}(\mu_0') = 2\sum_{p \in P \setminus F} \pi_p(\mu_0) + \sum_{p \in F} \pi_p(\mu_0) + 1$,
  $\pi_{end} = 2$,

[3] A projection is a morphism mapping a letter to a letter. In [2], the projection is already included in the definition of the CSS language. Therefore, the result looks somewhat different here.

– and δ' is defined by
$\#(p, I(t')) = 2\#(p, I(t))$, for $p \in P, t \in T$,
$\#(run, I(t')) = 2$, for $t \in T$,
$\#(sum, I(t')) = 2 \sum\limits_{p \in P \setminus F} \#(p, I(t)) + \sum\limits_{p \in F} \#(p, I(t))$, for $t \in T$,

$\#(p, O(t')) = 2\#(p, O(t))$, for $p \in P, t \in T$,
$\#(run, O(t')) = 2$, for $t \in T$,
$\#(sum, O(t')) = 2 \sum\limits_{p \in P \setminus F} \#(p, O(t)) + \sum\limits_{p \in F} \#(p, O(t))$, for $t \in T$,

$\#(p, I(e)) = 0$, for $p \in P$,
$\#(run, I(e)) = 2$,
$\#(sum, I(e)) = 2$,
$\#(p, O(e)) = 0$, for $p \in P$,
$\#(run, I(e)) = 1$,
$\#(sum, I(e)) = 0$.

It can be shown by induction that the sequence of transitions $t'_1 t'_2 \cdots t'_n$ can be fired in D and yields the marking μ' with $\pi_p(\mu') = 2\pi_p(\mu) + 1$, for $p \in P$, $\pi_{sum}(\mu') = 2 \sum_{p \in P \setminus F} \pi_p(\mu) + \sum_{p \in F} \pi_p(\mu) + 1$ and $\pi_{run}(\mu') = 2$ iff $t_2 t_2 \cdots t_n$ can be fired in $\mathcal{M}$ and yields μ. The transition e can be fired once and stops the computation, as it leaves the position *run* with 1 token.

Hence, an accepting marking μ'' of D satisfies $\pi_{sum}(\mu'') = 0$ and is reached after a transition sequence $t'_1 t'_2 \cdots t'_n e$. The transition μ' reached after $t'_1 t'_2 \cdots t'_n$ satisfies $\pi_{sum}(\mu') = 2$. For the marking μ obtained in $\mathcal{M}$ after the sequence $t_1 t_2 \cdots t_n$, it follows that $\pi_q(\mu) = 1$, for some $q \in F$, and $\pi_p(\mu) = 0$, for all $p \in P \setminus \{q\}$.

Consequently, a sequence is accepted by D iff it has the form $t'_1 t'_2 \cdots t'_n e$ and $t_1 t_2 \cdots t_n$ is in the CSS language of $\mathcal{M}$.

Lemma 3. *$CPN \subseteq PBLIND(n)$.*

Proof. Given a Petri net $D = (P, T, \delta, \mu_0)$ with $P = \{p_1, \ldots, p_k\}$, we construct the partially blind k-counter machine $\mathcal{A} = (Z, T, z_0, F, \delta')$ with

$$Z = \{z_0\} \cup \{z_i \mid 1 \le i \le k\} \cup \{z'_i \mid 1 \le i \le k\} \cup \{z_{i,t} \mid 1 \le i \le k, t \in T\},$$
$$F = \{z_i \mid 1 \le i \le k\},$$

and δ' contains the following transitions for $1 \le i \le k$:

– $(z_0, \lambda, z_i, \boldsymbol{\alpha})$ with $\alpha_i = \pi_{p_i}(\mu_0)$, $0 \le \alpha_j \le \pi_{p_j}(\mu_0)$ for $j \ne i$,
– $(z_i, \lambda, z'_i, (-1, \ldots, -1))$,
– $(z'_i, t, z_{i,t}, \boldsymbol{\beta})$ with $\beta_j = 1 - \#(p_j, I(t))$, for $1 \le j \le k$, $t \in T$,
– $(z_{i,t}, \lambda, z_i, \boldsymbol{\gamma})$ with $\gamma_i = \#(p_i, O(t))$, $-1 \le \gamma_j \le \#(p_j, O(t))$, for $i \ne j$, $t \in T$.

The machine works as follows. In the first step, by choosing state z_i, it guesses that place p_i is the first place which becomes empty. In the sequel, a run of D

is simulated. The number of tokens on p_i is stored in the i-th counter, while the contents of the j-th counter can be any number less than or equal to the number of tokens on place p_j, for $j \neq i$. The firing of a transition t is simulated by $\mathcal{A}$ in 3 steps:

- The application of $(z_i, \lambda, z_i', (-1, \ldots, -1))$ tests that all counters have a contents of at least 1.
- The application of $(z_i', t, z_{i,t}, \boldsymbol{\beta})$ decreases the counters by the number of tokens needed for firing t.
- The application of $(z_{i,t}, \lambda, z_i, \boldsymbol{\gamma})$ increases counter i by the number of tokens fired by t to p_i, and the counters j, $j \neq i$, by at most the number of tokens fired by t to p_j.

An ID $(z, \lambda, \mathbf{0})$ of $\mathcal{A}$ can be reached iff the input reaches in D a marking μ with $\pi_{p_i}(\mu) = 0$, and all previous markings are positive for all places.

Corollary 4. *1. CPN* $\subset$ *PBLIND*(n).

2. *For any* $L \in$ *PBLIND*(n)*, there are a language* L' *and a projection* h *such that* $L = h(L')$ *and* $L'\$ \in$ *CPN.*

Hitherto decision problems have not been studied for CPN languages. From the decidability properties of *PBLIND* we obtain the following results as direct consequences.

Corollary 5. *The membership, emptiness, finiteness, and intersection emptiness problems are decidable for CPN languages.* □

3 CPN languages versus regular languages

Obviously, since *CPN* only contains prefix codes and there are regular languages, which are not prefix codes, there exist regular languages which are not in *CPN*. Conversely, in [4], pages 88/89, an example of a CPN language L is given, which is not context-free. Therefore *CPN* and *REG* are incomparable. However, the second claim of Corollary 4 holds obviously for regular languages L, too. In this section we investigate the relation between *CPN* and *REG* more detailed.

A language K is called strictly bounded if there are letters $a_1, a_2, \ldots, a_n$ such that

$$K \subseteq a_1^* a_2^* \ldots a_n^* .$$

By [1], a strictly bounded language is regular if and only if it is a finite union of languages of the form

$$L = \{a_1^{r_1+k_1t_1} a_2^{r_2+k_2t_2} \ldots a_n^{r_n+k_nt_n} \mid k_i \geq 0, 1 \leq i \leq n\} \qquad (1)$$

for some $r_i \geq 0$ and $t_i \geq 0$ for $1 \leq i \leq n$. We say that a language is a very strictly bounded language if it is of the form (1). By definition, very strictly bounded languages are regular.

We now characterize the very strictly bounded languages in *CPN*.

Theorem 6. *A very strictly bounded language L is in CPN if and only if*

$$L = a_1^{u_1} a_2^{u_2} \dots a_n^{u_n}$$

where $n \geq 1$, $u_i \in \mathbb{N} \cup \{*\}$ for $1 \leq i \leq n$, $u_j = *$ for some j with $1 \leq j \leq n$ implies $u_{j+1} \in \mathbb{N}$ and $u_n \in \mathbb{N}$.

Proof. Let L be a very strictly bounded language in *CPN* and let $D = (P, T, \delta, \mu_0)$ be a Petri net with $L = \mathcal{L}(D)$. Obviously, $T = \{a_1, a_2, \dots, a_n\}$.

By supposition, L is of the form (1). Assume that $t_j > 0$ for some j, $1 \leq j \leq n$. Then

$$L' = \{a_1^{r_1} a_2^{r_2} \dots a_{j-1}^{r_{j-1}} a_j^{r_j + k_j t_j} a_{j+1}^{r_{j+1}} \dots a_n^{r_n} \mid k_j \geq 0\}$$

is an infinite subset of L. If there is a place p such that $\#(p, I(a_j)) > \#(p, O(a_j))$, then transition a_j can only fire at most q times where q depends on $r_1, r_2, \dots, r_{j-1}$. This contradicts the infinity of L'. Thus

$$\#(p, I(a_j)) \leq \#(p, O(a_j)) \text{ for } p \in P\,.$$

Now let p' be a place such that p' does not contain a token after using the sequence

$$v = a_1^{r_1} a_2^{r_2} \dots a_{j-1}^{r_{j-1}} a_j^{r_j} a_{j+1}^{r_{j+1}} \dots a_n^{r_n}\,.$$

If $\#(p', I(a_j)) < \#(p', O(a_j))$ for all $p' \in P$, then every $p \in P$ contains a token using

$$w = a_1^{r_1} a_2^{r_2} \dots a_{j-1}^{r_{j-1}} a_j^{r_j + t_j} a_{j+1}^{r_{j+1}} \dots a_n^{r_n}\,.$$

Thus $w \notin \mathcal{L}(D)$ by definition whereas in contrast to this $w \in L$ and $L = \mathcal{L}(D)$. Thus there is at least one place p' such that $\#(p', I(a_j)) = \#(p', O(a_j))$ and p' contains no token after using v. Now, obviously, p' also has no token after

$$u = a_1^{r_1} a_2^{r_2} \dots a_{j-1}^{r_{j-1}} a_j^{r_j + 1} a_{j+1}^{r_{j+1}} \dots a_n^{r_n}\,.$$

This implies $t_j = 1$.

Analogously, we can show that $r_j = 0$.

Thus L is of the form $L = a_1^{u_1} a_2^{u_2} \dots a_n^{u_n}$ with $u_i \in \mathbb{N} \cup \{*\}$. Let us assume that $u_j = u_{j+1} = *$ for some j, $1 \leq j \leq n$. Since $\#(p, I(a_j)) \leq \#(p, O(a_j))$ and $\#(p, I(a_{j+1})) \leq \#(p, O(a_{j+1}))$ for any $p \in P$, the sequences

$$f = a_1^{r_1} a_2^{r_2} \dots a_{j-1}^{r_{j-1}} a_j^{s_j} a_{j+1}^{s_{j+1}} a_{j+2}^{r_{j+2}} \dots a_n^{r_n}$$

and

$$g = a_1^{r_1} a_2^{r_2} \dots a_{j-1}^{r_{j-1}} a_j^{s_j - 1} a_{j+1} a_j a_{j+1}^{s_{j+1} - 1} a_{j+2}^{r_{j+2}} \dots a_n^{r_n}$$

induce the same markings for some $s_j, s_{j+1} \geq 1$. Since $f \in L$, we have $f \in \mathcal{L}(D)$ and therefore $g \in \mathcal{L}(D)$ in contrast to $g \notin L$ and $L = \mathcal{L}(D)$. Hence, for any j, $1 \leq j \leq n$, $u_j = *$ implies $u_{j+1} \neq *$.

Moreover, if $u_n = *$, then $L = \mathcal{L}(D)$ is not a prefix code in contrast to the fact mentioned in the Introduction.

Furthermore, if $u_i = 0$ for some j, then we can omit the letter a_i without changing L.

Therefore L has the desired form.

Conversely, assume that $a_1^{u_1} a_2^{u_2} \dots a_n^{u_n} \in T^+$ satisfies the condition of the theorem. It is obvious that $a_n^{u_n} \in CPN$. Assume that $a_{i+1}^{u_{i+1}} a_{i+2}^{u_{i+2}} \dots a_n^{u_n} \in CPN$ for some $i, 1 \leq i \leq n-1$. Let $D = (P, T, \delta, \mu_0)$ such that $\mathcal{L}(D) = a_{i+1}^{u_{i+1}} a_{i+2}^{u_{i+2}} \dots a_n^{u_n}$ where $T = \{a_{i+1}, a_{i+2}, \dots, a_n\}$. Now we construct the Petri net $\overline{D} = (\overline{P}, \overline{T}, \overline{\delta}, \overline{\mu}_0)$ such that $\mathcal{L}(\overline{D}) = a_i^{u_i} a_{i+1}^{u_{i+1}} \dots a_n^{u_n}$ where $\overline{T} = T \cup \{a_i\}$.

Case 1. $u_i \in \mathbf{N}$. Let $\overline{P} = P \cup \{\natural_1, \natural_2\}$. To define $\overline{\delta}$, we add the following values to those of δ:

$\#(\natural_1, I(a_i)) = 2,$

$\#(\natural_2, O(a_i)) = 1,$

$\#(\natural_2, I(a_{i+1})) = \#(\natural_2, O(a_{i+1})) = u_i + 1.$

Moreover, we define $\overline{\mu}_0$ as follows:

$\pi_p(\overline{\mu}_0) = \pi_p(\mu_0)$ for $p \in P$,

$\pi_{\natural_1}(\overline{\mu}_0) = 2u_i + 1,$

$\pi_{\natural_2}(\overline{\mu}_0) = 1.$

Then it can be seen that a_{i+1} can be fired only after the configuration of $a_i^{u_i}$ is performed and after that $\overline{D}$ simulates D. Thus $a_i^{u_i} a_{i+1}^{u_{i+1}} \dots a_n^{u_n} = \mathcal{L}(\overline{D})$.

Case 2. $u_i =$. Let $\overline{P} = P \cup \{\natural\}$. To define $\overline{\delta}$, we add the following values to those of δ:

$\#(\natural, I(a_i)) = \#(\natural, O(a_i)) = 2u_{i+1} + 1,$

$\#(\natural, I(a_{i+1})) = 2.$

Moreover, we define $\overline{\mu}_0$ as follows:

$\pi_p(\overline{\mu}_0) = \pi_p(\mu_0)$ for $p \in P$,

$\pi_\natural(\overline{\mu}_0) = 2u_{i+1} + 1,$

Then it is easy to see that $\overline{D}$ simulates D after calculating $a_i^t, t \geq 0$ and starting firing a_{i+1}. Thus $a_i^{u_i} a_{i+1}^{u_{i+1}} \dots a_n^{u_n} = \mathcal{L}(\overline{D})$. Hence, by induction hypothesis, $a_1^{u_1} a_2^{u_2} \dots a_n^{u_n} \in CPN$.

Theorem 7. *Let $D = (P, T, \delta, \mu_0)$ be a Petri net with*

– $P = \{p_1, p_2, \dots, p_r\}$,

– $T = \{a_1, a_2, \dots, a_n\}$, *and*

– *for* $1 \leq i \leq n$, $\sum_{j=1}^{r} m_{ij} \leq 0$ *where* $m_{i,j} = \#(p_j, O(a_i)) - \#(p_j, I(a_i))$.

Then $\mathcal{L}(D)$ is regular.

Proof. For any marking μ reachable by D and any i, $1 \leq i \leq n$, such that $\delta(\mu, a_i) \neq \emptyset$, we have

$$|\delta(\mu, a_i)| = |\mu| + \sum_{i=1}^{r} m_{ij} \leq |\mu|$$

and thus by induction

$$|\mu| \leq |\mu_0| .$$

Hence the set R of reachable markings of D is finite.

We now construct the finite automaton

$$\mathcal{A} = (R \cup \{e\}, T, \mu_0, F, \overline{\delta})$$

where e is an additional element,

$$F = \{(k_1, k_2, \ldots, k_r) \mid (k_1, k_2, \ldots, k_r) \in R, \min\{k_j \mid 1 \leq j \leq r\} = 0\}$$

and $\overline{\delta}$ is defined as follows:

- $\overline{\delta}(\mu, a) = \delta(\mu, a)$ for $\mu \in R \subseteq F$, $a \in T$ and $\delta(\mu, a) \neq \emptyset$,
- $\overline{\delta}(\mu, a_i) = e$ for $\mu \in R$, $a \in T$ and $\delta(\mu, a) = \emptyset$,
- $\overline{\delta}(\mu, a) = e$ for $\mu \in F$,
- $\overline{\delta}(e, a) = e$ for $a \in T$.

It is easy to see that $u = b_1 b_2 \ldots b_m \in \mathcal{L}(D)$ if and only if $\delta(\mu_0, u) \in F$ and $\delta(\mu_0, b_1 b_2 \ldots b_k) \in R \setminus F$ for $1 \leq k < m$ if and only if $\overline{\delta}(\mu_0, u) \in F$ if and only if $u \in \mathcal{L}(\mathcal{A})$.

Theorem 7 cannot be extended to the case where $\sum_{i=1}^{r} m_{ij} > 0$ for some i, $1 \leq i \leq n$. In order to see this we consider the Petri nets D_1 and D_2 in Figure 1. In both cases we get the same elements

$$m_{11} = m_{21} = 0, \; m_{12} = 1 \text{ and } m_{22} = -1.$$

$\mathcal{L}(D_1)$ is not regular, because

$$\mathcal{L}(D_1) \cap a_1^+ a_2^+ = \{a_1^n a_2^{n+1} \mid n \geq 0\} ,$$

and $\mathcal{L}(D_2) = \{a_2\}$ is regular.

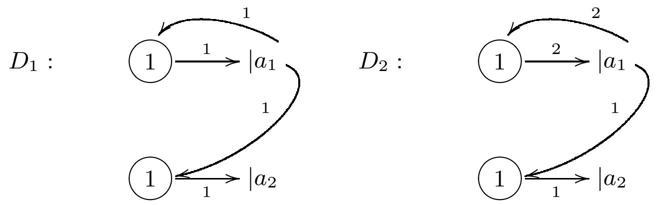

Figure 1: Petri nets D_1 and D_2

4 Density of CPN languages

By $Q(T)$ we denote the set of all primitive words over T, i.e., the family of all words q such that $q \neq p^j$ for all $p \in T^*$ and $j \geq 2$.

For a language L over T, we define the degree $deg(L)$ of L by

$$deg(L) = \{i \mid q^i \in L \text{ for some } q \in Q(T)\,.$$

Lemma 8. *Let $D = (P, T, \delta, \mu_0)$ be a Petri net with $\mu_0 = (k_1, k_2, \ldots, k_r)$. Further, let $k = \max\{k_i \mid 1 \leq i \leq r\}$. Then $deg(\mathcal{L}(D))$ is finite, more exactly, $deg(\mathcal{L}(D))$ contains at most k numbers.*

Proof. Suppose that $deg(\mathcal{L}(D))$ contains more than k numbers. Then there exists a positive integer i such that $i > k$ and $q^i \in \mathcal{L}(D)$ for some $q \in Q(T)$. By $q^i \in \mathcal{L}(D)$, there exists a place p such that $\pi_p(\delta(\mu_0, q^i)) = 0$. This implies $\pi_p(\delta(\mu, q)) < \pi_p(\mu)$ for all intermediate markings μ. Thus

$$0 = \pi_p(\delta(\mu_0, q^i)) \leq \pi_p(\mu_0) - i.$$

On the other hand, $\pi_p(\mu_0) - i < 0$ by the choice of i. This contradiction proves the lemma.

A language $L \subseteq T^*$ is called dense, if, for $u \in T^*$, there are words x and y in T^* such that $xuy \in L$.

Theorem 9. *Let T be an alphabet with at least two letters. Then there exist a dense language L in CPN but no regular dense language is in CPN.*

Proof. The first statement follows by the Petri net D given in Figure 2. It is easy to see that

$$a_1^{|u|} u a_2^* \cap \mathcal{L}(D) \neq \emptyset \text{ for } u \in T^*.$$

Therefore $\mathcal{L}(D)$ is dense.

Now suppose that there is a dense regular language L in *CPN*. By [3], the degree of any regular dense language is infinite. Thus $deg(L)$ is infinite. However, this contradicts Lemma 8.

Figure 2: A Petri net that generates a dense language

References

1. S. Ginsburg and E. H. Spanier, Bounded regular sets. *Proc. Amer. Math. Soc.* **17** (1966) 1043–1049.
2. S. A. Greibach, Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7:311–324, 1978.
3. M. Ito and M. Katsura, Dense languages and languages related to primitive words. In: *Proceedings of Asian Mathematical Conference 1990*, World Scientific, Singapore, 1992, 215–220.
4. M. Ito and Y. Kunimochi, Some Petri Net Languages and Codes. In: W. Kuich (ed.), *Developments in Language Theory*, Preproceedings of 5th Int. Conf. DLT, Vienna, 2001.
5. J. L. Petersen, *Petri Net Theory and the Modelling of Systems.* Prentice-Hall, New Jersey, 1981.
6. G. Rozenberg and A. Salomaa (eds.), *Handbook of Formal Languages*, Vol. I – III, Springer-Verlag, Heidelberg, Berlin, 1997.
7. G. Tanaka, Prefix codes determined by Petri nets. *Algebra Colloquium* **5** (1998) 255–264.