<table>
<thead>
<tr>
<th>Title</th>
<th>Circular Codes and Petri Nets (Algorithmic problems in algebra, languages and computation systems)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Tanaka, Genjiro</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (2006), 1503: 132-138</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2006-07</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/58465">http://hdl.handle.net/2433/58465</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
<tr>
<td>Institute</td>
<td>Kyoto University</td>
</tr>
</tbody>
</table>
Circular Codes and Petri Nets

Genjiro Tanaka

Dept. of Computer Science, Shizuoka Institute of Science and Technology,
Fukuroi-shi, 437-8555 Japan

Abstract

The purpose of this paper is to investigate the relationship between limited codes and Petri nets. For a given Petri net with an initial marking $\mu$, we can naturally define an automaton $A$ which has the initial marking $\mu$ as an initial state, the reachability set $Re(\mu)$ as a set of states, and the set of transitions as a set of inputs. We can define prefix codes by considering the set of firing sequences which arrive from the positive initial marking of a Petri net to a certain subset of the reachability set[10,12]. The set $M$ of all positive firing sequences which start from the positive initial marking $\mu$ of a Petri net and reach $\mu$ itself forms a pure monoid. Our main interest is in the base $D$ of $M$. The family of pure monoids contains the family of very pure monoids, and the base of a very pure monoid is a circular code. Therefore, we can expect that $D$ may be a circular code. Here, for “small” Petri nets, we discuss under what conditions $D$ is circular.

Key words: Petri net, Code, Prefix code, Circular code, Limited code.

1. Introduction

Let $A$ be an alphabet, $A^*$ the free monoid over $A$, and $1$ the empty word. A word $v \in A^*$ is a left factor of a word $u \in A^*$ if there is a word $w \in A^*$ such that $u = vw$. The left factor $v$ of $u$ is called proper if $v \neq u$. A right factor and a proper right factor of a word are defined in a symmetric manner.

For a word $w \in A^*$ and a letter $x \in A$ we let $|w|_x$ denote the number of $x$ in $w$. The length of $w$ is the number of letters in $w$. A non-empty subset $C$ of $A^+$ is said to be a code if for $x_1, \ldots, x_p, y_1, \ldots, y_q \in C$, $p, q \geq 1$,

$$x_1 \cdots x_p = y_1 \cdots y_q \text{ implies } p = q \text{ and } x_1 = y_1, \ldots, x_p = y_p.$$  

A subset $M$ of $A^*$ is a submonoid of $A^*$ if $M^2 \subseteq M$ and $1 \in M$. Every submonoid $M$ of a free monoid has a unique minimal set of generators

$$C = (M - \{1\}) - (M - \{1\})^2.$$  

$C$ is called the base of $M$.

This is the abstract and the details will be published elsewhere.
A submonoid $M$ is right unitary in $A^*$ if for all $u, v \in A^*$,
$$u, uv \in M \implies v \in M.$$  

$M$ is called left unitary in $A^*$ if it satisfies the dual condition. A submonoid $M$ is biunitary if it is both left and right unitary.

Definition 1.1. Let $M$ be a submonoid of a free monoid $A^*$, and $C$ its base. If $CA^+ \cap C = \emptyset$, (resp. $A^+C \cap C = \emptyset$), then $C$ is called a prefix (resp. suffix) code over $A$. $C$ is called a bifix code if it is a prefix and suffix code.

A submonoid $M$ of $A^*$ is right unitary (resp. biunitary) if and only if its minimal set of generator is a prefix code (bifix code) ([1,p.46]).

Definition 1.2. A Petri net is a 5-tuple, $PN = (P, A, F, W, \mu_0)$ where:
$P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places,  
$A = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions,  
$F \subseteq (P \times A) \cup (A \times P)$ is a set of arcs,  
$W : F \rightarrow \{1, 2, \ldots\}$ is a weight function,  
$\mu_0 : P \rightarrow \{0, 1, 2, \ldots\}$ is the initial marking,  
$P \cap A = \emptyset$ and $P \cup A \neq \emptyset$.

We use the following notations for a pre-set and a post-set:
\[ \cdot t = \{p | (p, t) \in F\}, \quad t \cdot = \{p | (t, p) \in F\} \]

In this paper we shall assume that a Petri net has no isolated transitions, i.e., no $t$ such that $\cdot t \cup t \cdot = \emptyset$. A marking $\mu_0$ can be represented by a vector:
$$\mu_0 = (\mu_0(p_1), \mu_0(p_2), \ldots, \mu_0(p_n)), \quad p_i \in P, \quad n = |P|.$$

For every $t \in A$ the vector $\Delta t$ is defined by
$$\Delta t = (\Delta t(p_1), \Delta t(p_2), \ldots, \Delta t(p_n)), \quad n = |P|,$$

where
$$\Delta t(p) = \begin{cases} 
-W(p, t) + W(t, p) & \text{if } p \in \cdot t \cap t \cdot, \\
-W(p, t) & \text{if } p \in \cdot t - t \cdot, \\
W(t, p) & \text{if } p \in t \cdot - t \cdot, \\
0 & \text{if } p \notin \cdot t \cup t \cdot.
\end{cases}$$

A transition $t \in A$ is said to be enabled in $\mu_0$, if $W(p, t) \leq \mu_0(p)$ for all $p \in \cdot t$. A firing of an enabled transition $t$ removes $W(p_1, t)$ tokens from each input place $p_1 \in \cdot t$, and adds $W(t, p_2)$ tokens to each output place $p_2 \in t$. Firing of an enabled transition $t$ at $\mu_0$ produces a new
marking \( \mu_1 \) such that

\[
\mu_1(p) = \begin{cases} 
\mu_0(p) - W(p,t) & \text{if } p \in t - t, \\
\mu_0(p) + W(t,p) & \text{if } p \in t \cdot t, \\
\mu_0(p) - W(p,t) + W(t,p) & \text{if } p \in t \cdot t, \\
\mu_0(p) & \text{otherwise.}
\end{cases}
\]

If we obtain the marking \( \mu' \) that results from a firing of \( t \) at \( \mu \), we write \( \delta(\mu,t) = \mu' \). A word \( w = t_1 t_2 \ldots t_r \), \((t_i \in A)\), of transitions is said to be a (firing) sequence from \( \mu_0 \) if there exist markings \( \mu_i, 1 \leq i \leq r \), such that \( \delta(\mu_{i-1},t_i) = \mu_i \) for all \( i \), \((1 \leq i \leq r)\). In this case, \( \mu_r \) is reachable from \( \mu_0 \) by \( w \) and we write \( \delta(\mu_0,w) = \mu_r \). The set of all possible markings reachable from \( \mu_0 \) is denoted by \( \text{Re}(\mu_0) \), and the set of all possible sequences from \( \mu_0 \) is denoted by \( \text{Seq}(\mu_0) \). The function \( \delta : \text{Re}(\mu_0) \times T \rightarrow \text{Re}(\mu_0) \) is called a next-state function of a Petri net \( PN \) [7,p.23]. We note that the above condition for \( r = 0 \) is understood to be \( \mu_0 \in \text{Re}(\mu_0) \).

A marking \( \mu \) is said to be positive if \( \mu(p) > 0 \) for all \( p \in P \). A sequence \( t_1 t_2 \ldots t_n \in \text{Seq}(\mu_0) \), \((t_i \in T)\), is called a positive sequence from \( \mu_0 \) if \( \delta(\mu_0,t_1 t_2 \ldots t_i) \) is positive for all \( i \), \((1 \leq i \leq n)\). The set of all positive sequences from \( \mu_0 \) is denoted by \( P\text{Seq}(\mu_0) \).

By \( P\text{Re}(\mu_0) \) we denote the set of all possible positive markings reachable from \( \mu_0 \);

\[
P\text{Re}(\mu_0) = \{ \delta(\mu_0,w) | w \in P\text{Seq}(\mu_0) \}.
\]

2. Some codes related to Petri nets

For a Petri net \( PN = (P,T,F,W,\mu_0) \) and a subset \( X \subseteq \text{Re}(\mu_0) \) we can define a deterministic automaton \( A(PN) \) as follows: \( \text{Re}(\mu_0) \), \( T \), \( \delta : \text{Re}(\mu_0) \times T \rightarrow \text{Re}(\mu_0) \), \( \mu_0 \), and \( X \), are regarded as a state set, an input set, a next-state function, an initial state, and a final set of \( A(PN) \), respectively. By using such automata, in [10,12] we defined four kinds of prefix codes and examined fundamental properties of these codes.

Let \( PN = (P,A,F,W,\mu) \) be a Petri net. The set

\[
\text{Stab}(PN) = \{ w | w \in \text{Seq}(\mu) \text{ and } \delta(\mu,w) = \mu \}
\]

forms a submonoid of \( A^* \). If \( \text{Stab}(PN) \neq \{1\} \), then we denote the base of \( \text{Stab}(PN) \) by \( S(PN) \). Since \( S(PN)A^+ \cap S(PN) = \emptyset \), \( S(PN) \) is a prefix code over \( A \).

A submonoid \( M \) of \( A^* \) is called pure [7] if for all \( x \in A^* \) and \( n \geq 1 \),

\[
x^n \in M \implies x \in M.
\]

A subsemigroup \( H \) of a semigroup \( S \) is extractable in \( S \) [9,p.191] if

\[
x,y \in S, z \in H, xyz \in H \implies xy \in H.
\]

Proposition 2.1. If \( \text{Stab}(PN) \neq \emptyset \), then \( \text{Stab}(PN) \) is a biunitary extractable pure monoid.
Definition 2.1. Let $PN = (P, A, F, W, \mu)$ be a Petri net with a positive marking $\mu$. Define the subset $D(PN)$ as a set of all positive sequence $w$ of $S(PN)$.

Since $D(PN)$ is a subset of $S(PN)$, $D(PN)$ is a bifix code over $A$.

Proposition 2.2. If $D(PN) \neq \emptyset$, then $D(PN)^*$ is a biunitary extractable pure monoid.

Example 2.1. Let $PN = (\{p, q\}, \{a, b\}, F, W, \mu_0)$ be a Petri net defined by $W(a, p) = W(p, b) = W(q, a) = W(b, q) = 1$, $\mu_0(p) = \mu_0(q) = 2$. Then $D(PN) = \{ab, ba\}$, therefore $\{ab, ba\}^*$ is pure [1, p.324, Ex.1.3].

Proposition 2.3. If $z, xzy \in D(PN), x, y \in A^+$, then $xz^*y \in D(PN)$.

A code $D$ is infix if $w, xwy \in D$ implies $x = y = 1$ [8, p.129].

Proposition 2.4. If $D(PN)$ is a non-empty finite set, then $D(PN)$ is a infix code.

3. Limited code

A submonoid $M$ of $A^*$ is very pure if for all $u, v \in A^*$,

$$u, v \in A^*, uv, vu \in M \Rightarrow u, v \in M.$$ 

The base of a very pure monoid is called a circular code.

Let $p, q \geq 0$ be two integers. If for any sequence $u_0, u_1, \ldots, u_{p+q}$ of words in $A^*$, the assumptions $u_{i-1}u_i \in M$ ($1 \leq i \leq p + q$) imply $u_p \in M$, then a submonoid $M$ is said to satisfy condition $C(p, q)$. If a submonoid $M$ of $A^*$ satisfies condition $C(p, q)$, then $M$ is very pure [1, p.329, Proposition 2.1], and its base is called a $(p, q)$-limited code.

If a subset $D$ of $A^*$ is a bifix $(1,1)$-limited code, then for any $u_0, u_1, u_2 \in A^*$ such that $u_0u_1, u_1u_2 \in D$ we have $u_1 \in D$. Thus $u_0u_1, u_1, u_2 \in D$. This implies that $u_0, u_1, u_2 \in D$, since $D$ is bifix. Therefore $D$ is $(2,0)$-, $(1,1)$- and $(0,2)$-limited.

Let $PN_0 = (\{p\}, \{a, b\}, F, W, \mu_0)$ be a Petri net such that $W(a, p) = \alpha, W(p, b) = \beta, \mu_0 = (\lambda_p), \lambda_p > 0$. 

Consider the set $\Omega$ of positive markings in $PN_0$:
\[
\Omega = \{ \mu | \mu = \mu_0 + \Delta(w), w \in PSeq(\mu_0) \}.
\]
$\alpha$ and $\beta$, and let $N = \{0, 1, 2, \cdots\}$ be a set of non-negative integers. Then we have

1. $D(PN_0)$ is dense.
2. If $\lambda_p < g$, then $\Omega = \{ \lambda_p + ng | n \in N \}$.
3. If $\lambda_p = sg, s \geq 0, s \in N$, then $\Omega = \{ng | n \geq 1, n \in N \}$.
4. If $\lambda_p = sg + t_p, s \geq 0, s \in N, 0 < t_p < g$, then $\Omega = \{t_p + ng | n \geq 0, n \in N \}$.

**Proposition 3.1.** If $\lambda_p > gcd(\alpha, \beta)$, then $D(PN_0)$ is not circular.

**Proposition 3.2.** $D(PN_0)$ is circular if and only if $\lambda_p \leq gcd(\alpha, \beta)$.

Let $PN_1 = (\{p, q\}, \{a, b\}, F, W, \mu_0)$ be a Petri net such that $W(a, p) = \alpha, W(p, b) = \alpha', W(q, a) = \beta, W(b, q) = \beta'$, $\mu_0(p) = \lambda_p, \mu_0(q) = \lambda_q$.

Suppose that $D(PN_1) \neq \emptyset$ and $w \in D(PN_1)$. Let $n = |w|_a$ and $m = |w|_b$, then $\Delta(w) = n\Delta(a) + m\Delta(b) = 0$ (zero vector). Consequently the linear equation
\[
\begin{pmatrix}
\alpha & -\alpha' \\
-\beta & \beta'
\end{pmatrix}
\begin{pmatrix}
n \\
m
\end{pmatrix}
= \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]
has a non-trivial solution in $N$. Thus $\alpha\beta' = \alpha'\beta$. Therefore, if $D(PN_1) \neq \emptyset$, then $PN_1 = (\{p, q\}, \{a, b\}, F, W, \mu_0)$ has the following forms:
$W(a, p) = \alpha, W(p, b) = k\alpha, W(q, a) = \beta, W(b, q) = k\beta, k > 0$.
Here we assume that $k$ is an integer. That is, we define a Petri net $PN_1 = (\{p, q\}, \{a, b\}, F, W, \mu_0)$ as follows
\[
\Delta(a) = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}, \quad \Delta(b) = \begin{pmatrix} -k\alpha \\ k\beta \end{pmatrix},
\]
where $k$ is a positive integer.

We define an integer $M_p$ as follows
\[
M_p = \begin{cases} 
\frac{\lambda_p}{\alpha} - 1, & \text{if } \frac{\lambda_p}{\alpha} \text{ is an integer,} \\
[\frac{\lambda_p}{\alpha}], & \text{if } \frac{\lambda_p}{\alpha} \text{ is not an integer.}
\end{cases}
\]
where \([\cdot]\) is the symbol of Gauss. Similarly we define an integer \(M_q\) as follows, 
\[M_q = \left\lfloor \frac{M}{q} \right\rfloor - 1\] 
if \(\frac{M}{q}\) is an integer, and \(M_q = \left\lfloor \frac{M}{q} \right\rfloor\) if \(\frac{M}{q}\) is not an integer.

**Proposition 3.3.** We have
(1) If \(M_p + M_q > k, M_p \geq k\) and \(M_q \geq 1\), then \(D(PN_1)\) is not circular.
(2) If \(M_p + M_q > k, M_p > 1\), then \(D(PN_1)\) is not circular.
(3) If \(M_p + M_q = k, M_p \geq 1\), then \(D(PN_1)\) is a singleton.
(4) If \(M_p + M_q \geq k, M_p = 0, M_q \geq 1\), then \(D(PN_1)\) is \((1,1)\)-limited.
(5) If \(M_p + M_q \geq k, M_p = 0, M_q = 0\), then \(D(PN_1)\) is \((1,1)\)-limited.

**Corollary 3.1.** Let \(n\) and \(k\) be arbitrary integers such that \(n > k > 1\). Define the automaton 
\[A_{(n,k)} = (\{1, 2, \ldots, n\}, \{a, b\}, f, 1, \{1\})\] 
by \(f(i, a) = i + 1, 1 \leq i \leq n-1\), \(f(j, b) = j - k, k + 1 \leq j \leq n\). Then the base of language \(L(A_{(n,k)})\) recognized by \(A_{(n,k)}\) is a \((1,1)\)-limited code.

**Proposition 3.4.** Let 
\[PN = (\{p_1, \ldots, p_n\}, \{a_1, \ldots, a_n\}, F, W, \mu_0), n \geq 2\] be a Petri net such that 
\[W(p_i, a_i) = \alpha_i, W(a_i, p_{i+1}) = \beta_i, 1 \leq i \leq n - 1, \text{ and } W(p_n, a_n) = \alpha_n, W(a_n, p_1) = \beta_n, \mu_0 = (\lambda_1, \ldots, \lambda_n), \mu_0(p_i) = \lambda_i, 1 \leq i \leq n.\] 
Furthermore let \(g_j = gcd(\beta_{j-1}, \alpha_j), 1 \leq j \leq n\). If \(\lambda_1/\alpha_1 > 1\) and \(\lambda_i \leq g_i\) for all \(i = 2, \ldots, n\), then \(D(PN)\) is \((1,1)\)-limited.

Let 
\[PN_2 = (\{p_1, p_2\}, \{a, b, c\}, F, W, \mu_0)\] be a Petri net such that 
\[W(a, p_1) = \alpha_1, W(p_1, b) = \alpha_2, W(b, p_2) = \beta_1, W(p_1, c) = \alpha_3, W(p_2, c) = \beta_2, \mu_0(p_1) = \lambda_1, \mu_0(p_2) = \lambda_2.\]

**Lemma 3.1.** Let \(PN_2\) be a Petri net mentioned above, and let \(\alpha = gcd(\alpha_1, \alpha_2, \alpha_3), \beta = gcd(\beta_1, \beta_2)\). Suppose that \(D(PN_2) \neq \emptyset\) and \(\lambda_1 \leq \alpha, \lambda_2 \leq \beta\). If \(d \in D(PN_2)\) and \(v\) is its proper suffix, then we have one of the following:
(1) \(\Delta(v)(p_1) \leq -\alpha, \Delta(v)(p_2) \leq -\beta\).
(2) \(\Delta(v)(p_1) = 0, \Delta(v)(p_2) \leq -\beta\).
(3) \(\Delta(v)(p_1) \leq -\alpha, \Delta(v)(p_2) \leq 0\).

**Proposition 3.5.** If \(D(PN_2) \neq \emptyset\) and \(\lambda_1 \leq \alpha, \lambda_2 \leq \beta\), then \(D(PN_2)\) is \((1,1)\)-limited.

Let 
\[PN_3 = (\{p, q\}, \{a, b, c\}, W, \mu_0)\] be a Petri net such that 
\[W(a, p) = \alpha, W(q, a) = \beta, W(p, b) = \]
\[ \alpha + \beta, W(b, q) = \alpha + \beta, W(c, p) = \beta, W(q, c) = \alpha, \mu_0(p) = \lambda_p, \mu_0(q) = \lambda_q. \]

**Lemma 3.2.** Let \( PN_3 \) be a Petri net mentioned above. Suppose that \( \beta < \lambda_p \leq \alpha + \beta \) and \( \beta < \lambda_q \leq \alpha \), then for any \( u \in PSeq(PN_3) \) we have one of the following.

1. \( \Delta(u) = \left( \begin{array}{c} \frac{k(\alpha - \beta)}{\alpha - \beta} \\ \frac{k(\alpha - \beta)}{\alpha - \beta} \end{array} \right), k \geq 0, \)
2. \( \Delta(u) = \left( \begin{array}{c} \frac{k(\alpha - \beta) + \lambda \alpha}{\alpha - \beta} \\ \frac{k(\alpha - \beta) - \lambda \beta}{\alpha - \beta} \end{array} \right), k \geq 0, l \geq 1, \)
3. \( \Delta(u) = \left( \begin{array}{c} \frac{k(\alpha - \beta) - \lambda \beta}{\alpha - \beta} \\ \frac{k(\alpha - \beta) + \lambda \alpha}{\alpha - \beta} \end{array} \right), k \geq 0, l \geq 1. \)

**Proposition 3.6.** Suppose that \( D(PN_3) \neq \emptyset \). If \( \beta < \lambda_p \leq \alpha + \beta \) and \( \beta < \lambda_q \leq \alpha \), then \( D(PN_3) \) is \((1,1)\)-limited.

**References**