From Post Systems to the Reachability Problems for Matrix Semigroups and Multicounter Automata

Igor Potapov *

Department of Computer Science,
University of Liverpool, Chadwick Building,
Peach St, Liverpool L69 7ZF, U.K.,
igor@csc.liv.ac.uk

Abstract. The main result of this paper is the reduction of PCP(n) to the vector reachability problem for a matrix semigroup generated by n 4×4 integral matrices. It follows that the vector reachability problem is undecidable for a semigroup generated by 7 integral matrices of dimension 4. The question whether the vector reachability problem is decidable for n=2 and n=3 remains open. Also we show that proposed technique can be applied to Post's tag-systems. As a result we define new classes of counter automata that lie on the border between decidability and undecidability.

1 Introduction

In this paper we show the connection between decision problems for Post systems and the reachability problems for matrix semigroups and counter automata.

We start from the vector reachability problem for a matrix semigroup, which is a generalisation of the orbit problem [12]. The vector reachability problem is formulated as follows: *"Let S be a given finitely generated semigroup of $n \times n$ matrices from $\mathbb{Q}^{n\times n}$ and vectors $\bar{x}$,$\bar{y}$ from $\mathbb{Q}^n$. Decide whether there is a matrix $M \in S$ such that $M \cdot \bar{x} = \bar{y}$."*

In fact, the problem has close relation to the membership problem in matrix semigroups [1, 7, 14] and the reachability in linear iterative maps. It is equivalent to the following reachability problem for non-deterministic linear maps: "Given two vectors u and v in n-dimensional vector space over $\mathbb{Q}$ and a set A of linear transformations. Determine whether exists a sequence of transformations from A such that maps v to u". In algebraic terms the vector reachability problem can be expressed as a problem of determining whether it is possible to get a vector u by an action of matrix semigroup on the initial vector v.

In case where a semigroup generated by one matrix the vector reachability is decidable in polynomial time [12]. It is shown in [14] that the vector reachability problem is decidable for the case of row-monomial matrix semigroups over

* Work partially supported by NAL/00684/G Nuffield Foundation grant.

semigroup $\mathcal{S}$, where $\mathcal{S}$ is an arbitrary finitely generated *commutative* matrix semigroup over an algebraic number field F. In particular the vector reachability problem is decidable for a case of row-monomial matrix semigroups over $\mathbb{Q}$ or $\mathbb{C}$. In general case the problem is also decidable for $n = 1$ since it can be reduced to the solution of the system of linear Diophantine equations. Since the membership problem for 3×3 matrix semigroup is undecidable it is also easy to see that the vector reachability problem is undecidable for any $n \geq 9$ by reduction of membership problem of dimension 3 to the vector reachability problem of dimension 9 [7].

As a main result of this paper we prove that the vector reachability problem is undecidable for any $n \geq 4$. In particular the undecidability result is based on the reduction the Post correspondence problem to the 4-dimensional vector reachability problem. The question whether the problem is decidable for n=2 and n=3 is still open.

As an extension of proposed reduction we apply a similar technique to the Post's tag-systems to show their connection to reachability problems for counter automata and to identify some classes of automata with decidable and undecidable reachability problems.

This paper is organised as follows. Next section contains preliminaries. In the Section 3 we show a matrix interpretation of Post correspondence problem and the main result. Another application of the developed technique is presented in Section 4, where we convert tag-systems to four counter automata. The paper ends with some conclusions and open problems.

2 Preliminaries

In what follows we use traditional denotations $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{Q}^+$ for the sets of naturals (non-negative integers), integers, rationals and non-negative rationals, respectively. A *semigroup* is a pair $(S, \cdot)$, where S is a set and $\cdot$ is an associative binary operation on S. A semigroup $(S, \cdot)$ is generated by a set A of its elements iff every element of S is a finite product $a_{i_1} \cdot a_{i_2} \cdot \ldots \cdot a_{i_k}$ where $a_{i_j} \in A$. The set of $n \times n$ matrices over rationals (integers) is denoted by $\mathbb{Q}^{n \times n}$ ($\mathbb{Z}^{n \times n}$). It is clear that the identity element for a semigroup $(\mathbb{Q}^{n \times n}, \cdot)$ or for $(\mathbb{Z}^{n \times n}, \cdot)$ is the identity matrix that we denote by E_n (or E).

We denote an empty word by ϵ. The concatenation of two strings w and v is a string obtained by appending the symbols of v to the right end of w, that is, if $w = a_1 a_2 \ldots a_n$ and $v = b_1 b_2 \ldots b_n$ then the concatenation of w and v, denoted by $w \cdot v$ or wv, is $a_1 a_2 \ldots a_n b_1 b_2 \ldots b_n$. The reverse of a string is obtained by writing the symbols in reverse order; if w is a string as shown above, then its reverse w^{-1} is $a_n \ldots a_2 a_1$.

2.1 The Stern-Brocot Tree

The Stern-Brocot tree is an elegant way for constructing the set of all nonnegative fractions $\frac{m}{n}$ where m and n are relatively prime [8].

Take two rational numbers, $\frac{a}{b}$ and $\frac{c}{d}$, and insert between them a third value, called the mediant, equal to $\frac{a+c}{b+d}$. Now with three numbers in hand, construct mediant between the first and second and between second and third, so that the next level of the tree has five numbers.

The canonical version of the Stern-Brocot tree starts with two irreducible fractions $\frac{0}{1}$, $\frac{1}{0}$ representing zero and infinity. Actually $\frac{0}{1}$ is a fraction while $\frac{1}{0}$ is not. However, we use them to describe a way to get all possible positive fractions arranged in a binary tree form as illustrated on Figure 1.

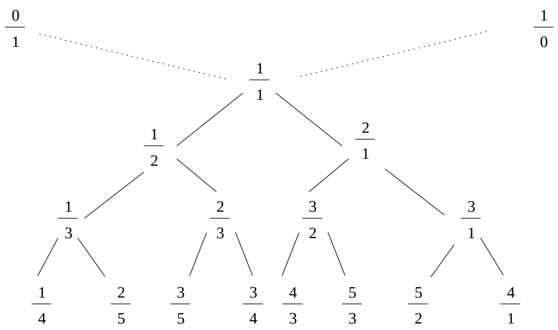

Fig. 1. The Stern-Brocot Tree.

So, for example, from $\frac{0}{1}$ and $\frac{1}{0}$ we get $\frac{1}{1}$. The mediant of $\frac{0}{1}$ and $\frac{1}{1}$ is $\frac{1}{2}$ while the mediant of $\frac{1}{1}$ and $\frac{1}{0}$ is $\frac{2}{1}$. On the next stage of the construction, we form four new fractions: $\frac{1}{3}$ from $\frac{0}{1}$ and $\frac{1}{2}$, $\frac{2}{3}$ from $\frac{1}{2}$ and $\frac{1}{1}$, $\frac{3}{2}$ from $\frac{1}{1}$ and $\frac{2}{1}$, and, finally, the mediant of $\frac{2}{1}$ and $\frac{1}{0}$ which is $\frac{3}{1}$. Continuing this way we get an infinite tree known as the Stern-Brocot tree [1].

If we specify the position of a fraction in the tree as a path consisting of L(eft) an R(ight) moves along the tree starting from the top (fraction), and also define matrices

$$M_L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, M_R = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

then product of the matrices corresponding to the path (or binary word) is matrix $\begin{pmatrix} n & n' \\ m & m' \end{pmatrix}$ whose entries are numerators and denominators of parent fractions $\frac{m}{n}$ and $\frac{m'}{n'}$. For example, the path leading to fraction $\frac{3}{5}$ is LRL. The corresponding matrix product is

$$M_L \times M_R \times M_L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix}$$

[1] It was discovered independently by the German mathematician Moriz Stern (1858) and by the French clock maker Achille Brocot (1860)

and the parents of $\frac{3}{5}$ are $\frac{1}{2}$ and $\frac{2}{3}$.

Let us consider a binary word $w{=}w_1 \cdot \ldots \cdot w_n$, that corresponds to some path in the Stern-Brocot tree. In case of matrix representation the empty word corresponds to the identity matrix and the binary word w corresponds to a matrix $M_w{=}M_{w_1} \times \ldots \times M_{w_n}$.

So let w be a binary word and M_w be its matrix representation. For example, in order to append character L to the word w (right concatenation) we have to multiply matrix M_w by M_L from the right, so the word $w' = w \cdot L$ corresponds to $M_w \times M_L$. Similarly we can add character R to the head of the word w (left concatenation), so the word $w' = R \cdot w$ corresponds to $M_R \times M_w$.

The left and the right contractions can be expressed using inverse matrices:

$$M_L^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, M_R^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}.$$

Let the head character of the word $w' = R \cdot w$ be R. We can delete a head character R if we multiply M_R^{-1} by M_w' then $M_R^{-1} \times M_{w'} = M_R^{-1} \times M_R \times M_w = M_w$. In order to delete the tail character L of the word $w' = w \cdot L$ we multiply $M_{w'}$ by M_L^{-1}, so $M_{w'} \times M_L^{-1} = M_w \times M_L \times M_L^{-1} = M_w$

2.2 Two Mappings Between Words and Matrices

Now we derive two mappings ψ and ϕ from the Stern-Brocot number system. First, let us consider the mapping ψ between $\{L, R\}^*$ and 2×2 matrices:

$$\psi : \epsilon \mapsto \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E \quad \psi : L \mapsto \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = M_L \quad \psi : R \mapsto \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = M_R$$

$$\psi : w_1 \cdot \ldots \cdot w_r \mapsto M_{w_1} \times \ldots \times M_{w_r}.$$

It follows from the properties of Stern-Brocot number system [8, 10] that the mapping ψ is an isomorphism between $\{L, R\}^*$ and elements of matrix semigroup generated by 2×2 matrices $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. Since for every matrix with nonzero determinant there is only one unique inverse matrix we can also define a similar mapping ϕ. It can be defined using inverse matrices of the semigroup generator. Mapping ϕ is also an isomorphism between $\{L, R\}^*$ and elements of matrix semigroup generated by 2×2 matrices $\{M_L^{-1}, M_R^{-1}\}$:

$$\phi : \epsilon \mapsto \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E \quad \phi : L \mapsto \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} = M_L^{-1} \quad \phi : R \mapsto \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} = M_R^{-1}$$

$$\phi : w_1 \cdot \ldots \cdot w_r \mapsto M_{w_1}^{-1} \times \ldots \times M_{w_r}^{-1}.$$

Note, that these two mappings from $\{L, R\}^*$ to matrices are injective, the mappings from $w \in \{L, R\}^+$ to $w^{-1} \in \{L, R\}^+$ and from $\psi(u)$ to $\phi(u^{-1})$ are bijective (see Figure 2). There are many interesting properties of mappings ϕ and ψ, but we state here only some of them that will be used in the paper.

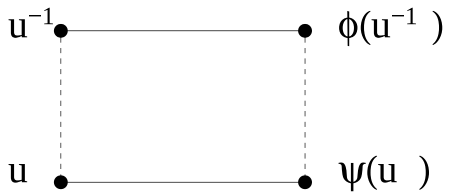

Fig. 2. Mappings between words and matrices.

Proposition 1. *Given a word $w \in \{L, R\}^+$ and $w' \in \{L, R\}^*$.*

- *Let* $\psi(w) = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$ *then* $e, h > 0$ *and* $f, g \geq 0$*;*
- *Let* $\phi(w) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *then* $a, d > 0$ *and* $b, c \leq 0$*;*
- *Let* $\phi(w' \cdot L) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *then* $a \leq |b|$*; Let* $\phi(w' \cdot R) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *then* $d \leq |c|$*;*
- *Let* $\psi(R \cdot w') = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$ *then* $e \leq g$*; Let* $\psi(L \cdot w') = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$ *then* $h \leq f$*.*

Proof. These properties hold for a case of four initial matrices $\phi(L)$, $\phi(R)$, $\psi(L)$, $\psi(R)$. The straightforward check of the matrix operations

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \phi(L), \begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \phi(R), \psi(L) \times \begin{pmatrix} e & f \\ g & h \end{pmatrix}, \psi(R) \times \begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

and easy induction on the length of words w and w' shows the above properties.

Lemma 1. *Given two words $u, v \in X^*$, $u = v$ iff $\phi(u^{-1}) = (\psi(v))^{-1}$.*

Proof. $\Rightarrow$ Let $u = v$ then $\phi(u^{-1}) \times \psi(v)$ corresponds to a product of matrices of even length of the form: $A_n \times \ldots \times A_1 \times B_1 \times \ldots \times B_n$ where $A_i \in \{M_L^{-1}, M_R^{-1}\}$, $B_j \in \{M_L, M_R\}$, $i, j \in 1..n$ and for each symmetric pair of matrices A_k,B_k we have A_k=B_k^{-1}. From it follows that $A_n \times \ldots \times A_1 \times B_1 \times \ldots \times B_n = A_n \times \ldots \times A_2 \times E \times B_2 \times \ldots \times B_n = \ldots = A_n \times E \times B_n = E$. The same idea of proof works for $\psi(v) \times \phi(u^{-1})$.

$\Leftarrow$ Let $\phi(u^{-1}) = (\psi(v))^{-1}$ then $\psi(v)$ is the unique inverse matrix for $\phi(u^{-1})$ that should be equal to $\psi(u)$, since $\phi(u^{-1}) = (\psi(u))^{-1}$. By injectivity of mapping from $\{L, R\}^*$ to matrices we have that $u = v$.

3 Post Correspondence Problem and its Matrix Interpretation

Post correspondence problem (in short, PCP) is formulated as follows: Given a finite alphabet X and a finite sequence of pairs of words in X^*: $(u_1, v_1), \ldots, (u_k, v_k)$.

Is there a finite sequence of indexes $\{i_j\}$ with $\{i_j \in \{1..k\}\}$, such that

$$u_{i_1} \cdot \ldots \cdot u_{i_n} = v_{i_1} \cdot \ldots \cdot v_{i_n}?$$

PCP(n) denotes the same problem with a sequence of n pairs. Without loss of generality we assume that the alphabet X is binary.

Lemma 2. *Given a finite binary alphabet X and a finite sequence of pairs of words in X^*:*

$$(u_1, v_1), \ldots, (u_k, v_k)$$

and a finite sequence of indexes $\{i_j\}$ with $\{i_j \in \{1..k\}\}$. The word $u = u_{i_1} \cdot \ldots \cdot u_{i_n}$ is equal to the word $v = v_{i_1} \cdot \ldots \cdot v_{i_n}$ if and only if

$$\phi(u^{-1}) \times \psi(v) = E.$$

Proof. The proposition flollows from Lemma 1 and the fact that matrices $\psi(w)$ and $\phi(w)$ have the inverse matrices (elements) for any word $w \in \{L, R\}^*$.

Theorem 1. *PCP(n) can be reduced to the vector reachability problem for a semigroup generated by n 4×4-matrices.*

Proof. Given a sequence of pairs of words in a binary alphabet $A^* = \{L, R\}$: $(u_1, v_1), \ldots, (u_n, v_n)$. Let us construct the sequence of pairs of 2×2 matrices using two mappings ϕ and ψ: $(\phi(u_1), \psi(v_1)), \ldots, (\phi(u_n), \psi(v_n))$.

Instead of equation $u = v$ we would like to consider a concatenation of two words $u^{-1} \cdot v$ that is a palindrome in case where $u = v$. Now we show a matrix interpretation of this concatenation. We associate 2×2 matrix C with a word w of the form $u^{-1} \cdot v$. Initially C is an identity matrix corresponding to an empty word. The extension of a word w by a new pair of words (u_r, v_r) (i.e. that gives us $w' = u_r^{-1} \cdot w \cdot v_r$) corresponds to the following matrix multiplication

$$C_{w'} = C_{u_r^{-1} \cdot w \cdot v_r} = \phi(u_r^{-1}) \times C_w \times \psi(v_r) \tag{1}$$

Let us rewrite the operation (1) in more details.

$$\begin{pmatrix} c_{w'}^{11} & c_{w'}^{12} \\ c_{w'}^{21} & c_{w'}^{22} \end{pmatrix} = \begin{pmatrix} u^{11} & u^{12} \\ u^{21} & u^{22} \end{pmatrix} \times \begin{pmatrix} c_w^{11} & c_w^{12} \\ c_w^{21} & c_w^{22} \end{pmatrix} \times \begin{pmatrix} v^{11} & v^{12} \\ v^{21} & v^{22} \end{pmatrix} \tag{1'}$$

According to the Lemma 2 $u = u_{i_1} \cdot \ldots \cdot u_{i_n} = v_{i_1} \cdot \ldots \cdot v_{i_n} = v$ for a finite sequence of indexes $\{i_j\}$ with $\{i_j \in \{1..k\}\}$ if and only if $\phi(u^{-1}) \times \psi(v)$ is equal to the identity matrix. So the question of the word equality can be reduced to the problem of finding a sequence of pairwise matrix multiplications that gives us the identity matrix.

Now we show that it is possible to avoid pairwise matrix multiplications by increasing the dimension from 2 to 4. Actually we represent matrices C_w and

$C_{w'}$ from (1') as 4×1 vectors and we unite every pair of matrices $\phi(u_r^{-1})$ and $\psi(v_r)$ into 4×4 joint matrix $M_{u_r^{-1},v_r}$ in the following way:

$$\begin{pmatrix} c_{w'}^{11} \\ c_{w'}^{12} \\ c_{w'}^{21} \\ c_{w'}^{22} \end{pmatrix} = \underbrace{\begin{pmatrix} u^{11}\cdot v^{11} & u^{11}\cdot v^{21} & u^{12}\cdot v^{11} & u^{12}\cdot v^{21} \\ u^{11}\cdot v^{12} & u^{11}\cdot v^{22} & u^{12}\cdot v^{12} & u^{12}\cdot v^{22} \\ u^{21}\cdot v^{11} & u^{21}\cdot v^{21} & u^{21}\cdot v^{11} & u^{22}\cdot v^{21} \\ u^{21}\cdot v^{12} & u^{21}\cdot v^{22} & u^{21}\cdot v^{12} & u^{22}\cdot v^{22} \end{pmatrix}}_{M_{u_r^{-1},v_r}} \cdot \begin{pmatrix} c_{w}^{11} \\ c_{w}^{12} \\ c_{w}^{21} \\ c_{w}^{22} \end{pmatrix} \qquad (2)$$

Note that the equation (1') is equivalent to the equation (2) in sense that the expression for computing values of $c_{w'}^{11}$, $c_{w'}^{12}$, $c_{w'}^{21}$ and $c_{w'}^{22}$ in (2) coincide with the corresponding values in (1').

Thus for every pair of words (u_r, v_r) we construct the matrix $M_{u_r^{-1},v_r}$. Now PCP(n) can be reduced to the following vector reachability problem: Given a matrix semigroup S generated by set of matrices $\{M_{u_1^{-1},v_1}, \ldots, M_{u_n^{-1},v_n}\}$. Decide whether there is a matrix $M \in S$ such that

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = M \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

It follows that if we can solve the above problem then we can solve the PCP(n).

Matiyasevich and Senizergues proved in [16] that the PCP(7) is undecidable. Thus the following corollary of the Theorem 1 holds.

Corollary 1. *The vector reachability problem is undecidable for a semigroup generated by* 7 *matrices of dimension* ≥ 4.

4 Post's Tag-systems

In this section we consider tag-systems, which is another interesting family of systems proposed by Post. We plan to show that problems about tag systems can be converted to the reachability problems in counter automata using the techniques and two mappings ψ and ϕ presented in Sections 2 and 3.

A tag system is set of rules that specifies a fixed number of elements to be removed from the beginning of a sequence and a set of elements to be appended ("tagged" onto the end) based on the elements that were removed from the beginning. Tag systems have a Turing machine-like halting problem for deciding based on an arbitrarily given initial sequence whether repeated application of the rules leads to a word of length smaller than the number of elements removed from the beginning [17]. Wang [19] also considered a sort of opposite to a tag system that he dubbed a lag system. Lag systems allow dependence on more than just the first element, but remove only the first element. The tag and lag systems can be represented as a communicating finite state machine [6] that interacts with one FIFO channel.

4.1 CFSM with FIFO Channel

Definition 1. *A Communication Finite State Machine (CFSM) is a finite transition system given by a 4-tuple $C = (Q, q_0, \Sigma, \delta)$ where : Q is a finite set of states, $q_0 \in Q$ is the initial state, Σ is a finite alphabet, and δ is a transition function such that $\delta \subseteq Q \times (\{+, -\} \times \Sigma) \times Q$.*

Note that C sees $(+, a)$ and $(-, a)$ as single symbols, which we henceforth write as $+a$ and $-a$ respectively. The label $-a$ denotes the emission of a in the channel, and $+a$ denotes the reception of a from the channel (Figure 3).

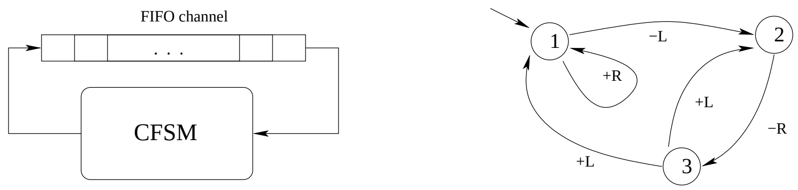

Fig. 3. The communicating finite state machine model.

The FIFO channel F, that contains the word w, can be changed only by two modes: either to send another message a to the channel and then F will store the word $w \cdot a$, or if F contains a word $w' = a \cdot w$, to receive a message a, then F will store a word w.

Now for any CFSM with a finite alphabet $\{L, R\}$ we can construct an equivalent model of a matrix transition system T with the same set of states and transitions but with the different transition function and the set of labels. The configuration of T will be represented by pair (M, s), where M is an integral matrix $\begin{pmatrix} n & n' \\ m & m' \end{pmatrix}$ and s is a state such that $s \in Q$.

Note, that the operation of concatenation can be performed on any word. However the contraction is a conditional operation since we need to know the head or the tail character of a word in the FIFO channel. The Proposition 1 states that we can identify the head or the tail symbol of a word by checking linear inequalities on matrix elements.

The transition system T can be constructed by changing the labels of CFSM in compliance with the following rules:

- $-\ L$: to substitute all labels $-L$ by label $\times M_L$
- $-\ R$: to substitute all labels $-L$ by label $\times M_R$
- $+\ L$: to substitute all labels $+L$ by guard $(m < n)$ and label $M_L^{-1} \times$
- $+\ R$: to substitute all labels $+R$ by guard $(m \geq n)$ and label $M_R^{-1} \times$

Here the label $\times M_a$ denotes the right multiplication and corresponds to the emission of a symbol a in the channel, and $M_a^{-1}\times$ denotes the left multiplication and corresponds to the reception of a from the channel.

The dynamics of the matrix transition system can be defined as follows. Let the current configuration of T be a pair $[M, s]$. The transition system T can reach the configuration $[M \times M_L, s']$ from (M, s) by a transition $\times M_L$ or the configuration $[M \times M_R, s']$ by a transition $\times M_R$ form state s to state s'.

Also the transition system T could have transition with pre and post conditions. For example the configuration $[M_L^{-1} \times M, s']$ can be reached from (M, s) by a transition $n > m | M_L^{-1}\times$ from state s to state s' if the precondition $n > m$ is true.

It is easy to see that we can go from the transition system T to counters automaton by introducing the individual counters for each matrix element. Let us rewrite matrix multiplications as a set of equations with four counters c_1, c_2, c_3, c_4 to keep track of values n, n', m and m', respectively .

$\times M_R$ corresponds to $\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} \times \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} c_1 + c_2 & c_2 \\ c_3 + c_4 & c_4 \end{pmatrix}$

$\times M_L$ corresponds to $\begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} c_1 & c_1 + c_2 \\ c_3 & c_3 + c_4 \end{pmatrix}$

$M_R^{-1}\times$ corresponds to $\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \times \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ c_3 - c_1 & c_4 - c_2 \end{pmatrix}$

$M_L^{-1}\times$ corresponds to $\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} = \begin{pmatrix} c_1 - c_3 & c_2 - c_4 \\ c_3 & c_4 \end{pmatrix}$

Now we have finished reduction of CFSM with one FIFO into a special form of counter automaton. Since the system of CFSM with FIFO can model a Turing Machine [19, 9] the following theorem holds:

Theorem 2. *Let A be a 4-counter automata where each transition is from the set of unconditional jumps:*

$$\begin{cases} c_1 := c_1 + c_2 \\ c_3 := c_3 + c_4 \end{cases}; \begin{cases} c_2 := c_1 + c_2 \\ c_4 := c_3 + c_4 \end{cases}$$

or from the set conditional jumps:

$$\text{if } (c_1 > c_3) \text{ then } \begin{cases} c_1 := c_1 - c_3 \\ c_2 := c_2 - c_4 \end{cases};$$
$$\text{if } (c_1 \leq c_3) \text{ then } \begin{cases} c_3 := c_3 - c_1 \\ c_4 := c_4 - c_2 \end{cases}.$$

Then A can simulate a Turing Machine.

The most popular universal model of computations is a model of two register (counter) machine or so called Minsky machine. In that model we can independently increment or decrement each counter and check them for zero. There are many other models of 2 and 3 counter machines that are universal as well but that have different basic operations such as *reset* ($c_i := 0$), *transfer* ($c_i := c_i + c_j$), different tests like checking the equality or inequality between counters ($c_i = c_j$ or $c_i < c_j$).

For example the reachability problem for a model of two counter automata with four operations such as $\{+1, -1, reset, transfer\}$ is decidable. The similar weak automaton (without testing for zero) with three counters have undecidable reachability and boundedness problems [4]. The model of two counter machine with only one test for zero and one reset (or transfer operation) has effectively computable semilinear reachability set [5]. So the idea balancing between the size of dimension and power of basic operations gives a number of classes on the border between decidability and undecidability.

The class of automata from Theorem 2 is another class of such system, where in dimension 4 we use *transfer* operation, *reverse transfer* operation ($c_i := c_i - c_j$) and comparison between only two counters. Moreover, as it will be shown in the next subsection, there is a resembling class of automata with decidable reachability properties.

4.2 CFSM with LIFO Channel

The CFSM with a LIFO channel or stack is a pushdown automata. The only difference from tag system is that we write or read to/from one side of the channel. In such case the LIFO channel can be represented as a product of matrices that could be extended only from one side. So we use the following labels $M_R\times$, $M_L\times$, $M_R^{-1}\times$ and $M_L^{-1}\times$ for the matrix transition system and other guards from Proposition 1 to check the first symbol in the stack.

The LIFO channel L, that contains the word w, can be changed only by two modes: either to send another message a to the channel and then L will store the word $a \cdot w$, or if L contains a word $w' = a \cdot w$, to receive a message a, then L will store a word w. The label $M_a\times$ denotes the right matrix multiplication and corresponds to the push a into the stack or LIFO channel and $M_a^{-1}\times$ corresponds to the pop an a off the top of the stack. We use the same method that converts a CFSM into the counter automaton. again. Note that the 4-counter automata from Theorem 3 is very similar to the automata from Theorem 2, but the status of reachability problem is opposite.

Theorem 3. *Let A is a 4-counter automata where each transition is from the set of unconditional jumps:*

$$\begin{cases} c_1 := c_1 + c_3 \\ c_2 := c_2 + c_4 \end{cases} ; \begin{cases} c_3 := c_1 + c_3 \\ c_4 := c_2 + c_4 \end{cases}$$

or from the set conditional jumps:

$$\textit{if } (c_1 > c_3) \textit{ then } \begin{cases} c_1 := c_1 - c_3 \\ c_2 := c_2 - c_4 \end{cases} ;$$

$$\textit{if } (c_1 \leq c_3) \textit{ then } \begin{cases} c_3 := c_3 - c_1 \\ c_4 := c_4 - c_2 \end{cases} .$$

The counters automata A is equivalent to a pushdown automata and therefore has decidable reachability problem.

It is possible to apply the same methods to different tag-systems [19, 9, 13] to derive a variety of multicounter automata with different structural constraints that have decidable or undecidable properties.

5 Conclusion

The main result of this paper is the reduction of PCP(n) to the vector reachability problem with 4×4 matrix semigroup with n generators. We leave the question whether the problem is decidable or undecidable for n=2 and n=3 as an open problem. We have also shown that this technique can be applied for other Post-systems and can be used for further investigation about the border between decidability and undecidability in the class of multicounter automata.

Author is very grateful to anonymous referees for their careful review and helpful suggestions on an earlier version of the paper and also to Alexei Lisitsa for many discussions and valuable comments about this work.

References

1. J.Bestel and Karhumaki. Combinatorics on Words - A Tutorial, Bulletin of the EATCS, February 2003, 178 - 228
2. V. Blondel and J. Tsitsiklis. A survey of computational complexity results in systems and control. Automatica, 36, 2000, 1249-1274
3. H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger arithmetic, Research Report LSV-98-1, Mar.,1998.
4. C. Dudourd, A.Finkel and Ph. Schnoebelen. Resets nets between decidability and undecidability. In Proc. 25th Int. Coll. Automata, Languages and Programming, LNCS 1443, 1998, 103-115
5. A. Finkel and G. Sutre. Decidability of reachability problems for classes of two counters automata. 17th Ann. Symp. Theoretical Aspects of Computer Science (STACS'2000), Lille, France, Feb. 2000, volume 1770 of Lecture Notes in Computer Science, 2000, 346-357
6. Alain Finkel, S. Purushothaman Iyer, Grgoire Sutre. Well-abstracted transition systems: application to FIFO automata. Inf. Comput. 181(1), 2003, 1-31
7. S. Gaubert and R. Katz. Reachability problems for products of matrices in semirings. arXiv:math.OC/0310028, v1. 2003, 1-21
8. R.Graham, D.Knuth, O.Patashnik. Concrete Mathematics. 2nd edition, Addison-Wesley, 1994
9. M. Gouda, E. Manning, Y. Yu. On the progress of communication between two finite state machines. Information and control, 63, 1984, 200-216
10. Brian Hayes. On the Teeth of Wheels. American Scientist, Volume 88, Number 4, July-August, 2000, 296-300
11. O. H. Ibarra, J. Su, T. Bultan, Z. Dang, and R. A. Kemmerer. Counter Machines: Decidable Properties and Applications to Verification Problems. MFCS, LNCS 1893, 2000, 426-435

12. R. Kannan and R. J. Lipton. Polynomial-time algorithm for the orbit problem, Journal of the ACM (JACM), Volume 33 , Issue 4, 1986, 808 - 821
13. O. Kurganskyy, I. Potapov. On the bound of algorithmic resolvability of correctness problems of automaton interaction through communication channels. *Cybernetics and System Analysis*, 3, 1999, 49-57
14. A. Lisitsa, I. Potapov. Membership and reachability problems for row-monomial transformations. MFCS, 2004, 623-634
15. Maurice Mergenstern. Frontier between decidability and undecidability: a survey. Theoretical Computer Science, v.231, 2000, 217-251
16. Yu. V. Matiyasevich, G. Senizergues, Decision problems for semi-Thue systems with a few rules, Proc. LICS'96, 1996, 523-531
17. M. Minsky. Computation: Finite and Infinite Machines. Englewood Cliffs, N.J.: Prentice-Hall, 1967
18. C.Moore. Unpredictability and Undecidability in Dynamical Systems, *Physical Review Letters*, Vol.64, N. 20, 1990, 2354-2357
19. Wang, H. "Tag Systems and Lag Systems." Math. Ann. 152, 1963, 65-74