Computer Science Journal of Moldova, vol.**10**, no.3(30), 2002

Nine Universal Circular Post Machines *

Artiom Alhazov Manfred Kudlek Yurii Rogozhin

Abstract

We consider a new kind of computational device like Turing machine, so-called circular Post machines with a circular tape and moving in one direction only, introduced recently by the second and the third authors. Using 2-tag systems we construct new nine small universal machines of this kind.

1 Introduction

We consider (deterministic) Circular Post machines (*CPM*) introduced recently by the second and the third authors of the paper [4, 5]. These are similar to those presented in [1], with the difference that the head can move only in one direction on the circular tape. It is also possible to erase a cell or to insert a new one. We introduce 5 variants of such machines, distinguished by the way a new cell is inserted. In [4, 5] it has been shown that all variants are equivalent to each other, and also to Turing machines, and that for all variants there exist equivalent Circular Post machines with 2 symbols, as well as with 2 states.

In 1956 Shannon [18] introduced the problem of constructing very small universal (deterministic) Turing machines. The underlying model of Turing machines is defined by instructions in form of quintuples $(\mathbf{p}, x, y, m, \mathbf{q})$ with the meaning that the machine is in state $\mathbf{p}$, reads symbol $x \in \Sigma$, overwrites it by y, moves by $m \in \{-1, 0, 1\}$, and goes into state $\mathbf{q}$.

Let *UTM*(m, n) be a class of universal Turing machine with m states and n symbols. It was known that there exist universal Turing

*Work supported by NATO project PST.CLG.976912 and project IST-2001-32008 "MolCoNet"

machines in the following classes: *UTM(19,2), UTM(10,3), UTM(7,4), UTM(5,5), UTM(4,6), UTM(3,9), UTM(2,18)* [16, 17, 2, 6], and the classes *UTM(2,2)* [3], *UTM(2,3)*, and *UTM(3,2)* [11] are empty. So there are 45 classes *UTM*(m,n) with an unsettled emptiness problem, i.e if *UTM*(m,n) is empty (for closely related problems see [7]).

To construct small universal Circular Post machines we use a method first presented in [9] (see also [15, 16]). This method uses tag systems [8] which are special cases of monogenic Post Normal systems [14], namely of the form $s_i vu \to u\alpha_i$ with $v \in \Sigma^{k-1}$ and $k > 1$ a constant. In [9, 10] it is also shown that 2-tag systems (i.e. $k = 2$) suffice to simulate all Turing machines, with halting only when encountering a special symbol s_H.

Since circular Post machines are also monogenic Post Normal systems we expect to get a more natural simulation of tag systems and perhaps smaller universal machines.

Circular Post machines may be useful for studying the formal models of biocomputing as well as membrane computing where the DNA molecules are present in the form of a circular sequence [12, 13].

Let *UPMi*(m,n) be a class of universal circular Post machines of variant i with m states and n symbols. In previous articles [4, 5] it was shown that such machines can simulate all Turing machines, and some small universal circular Post machines have been constructed, namely in the classes: *UPM0(4,18)*, *UPM0(5,11)*, *UPM0(6,8)*, *UPM0(7,7)*, *UPM0(8,6)*, *UPM0(9,5)*, *UPM0(12,4)* and *UPM0(18,3)*.

In this article we improve and complete all previous results on circular Post machines and present nine universal machines, namely in the classes: *UPM0(2,46)*, *UPM0(3,22)*, *UPM0(4,11)*, *UPM0(5,8)*, *UPM0(6,6)*, *UPM0(8,5)*, *UPM0(11,4)*, *UPM0(16,3)* and *UPM0(34,2)*.

Note that all nine universal Circular Post machines presented here were checked by the Circular Post machines SIMULATOR program designed by the first author.

2 Definitions

Definition 1: (Circular Post machine (*CPM0*)) *A Circular Post ma-*

chine is a quintuple $(\Sigma, Q, \mathbf{q}_0, \mathbf{q}_f, P)$ *with a finite alphabet* Σ *where* 0 *is the blank, a finite set of states* Q*, an initial state* $\mathbf{q}_0 \in Q$*, a terminal state* $\mathbf{q}_f \in Q$*, and a finite set of instructions of the forms*

$\mathbf{p}x \to \mathbf{q}$ *(erasing of the symbol read)*

$\mathbf{p}x \to y\mathbf{q}$ *(overwriting and moving to the right)*

$\mathbf{p}0 \to y\mathbf{q}0$ *(overwriting and creation of a blank)*

The storage of such a machine is a circular tape, the read and write head moving only in one direction (to the right), and with the possibility to cut off a cell or to create and insert a new cell with a blank.

This version is called variant 0 (*CPM0*).

Note that by erasing symbols the circular tape might become empty. This can be interpreted that the machine, still in some state, stops. However, in the universal machines constructed later, this case will not occur.

In this article it will be assumed that all machines are deterministic.

There are variants equivalent to such machines [4, 5].

Variant *CPM1* : The instructions are of the form

$\mathbf{p}x \to \mathbf{q}$ $\quad$ $\mathbf{p}x \to y\mathbf{q}$ $\quad$ $\mathbf{p}x \to x\mathbf{q}0$ (0 blank)

Variant *CPM2* : The instructions are of the form

$\mathbf{p}x \to \mathbf{q}$ $\quad$ $\mathbf{p}x \to y\mathbf{q}$ $\quad$ $\mathbf{p}x \to y\mathbf{q}0$ (0 blank)

Variant *CPM3* : The instructions are of the form

$\mathbf{p}x \to \mathbf{q}$ $\quad$ $\mathbf{p}x \to y\mathbf{q}$ $\quad$ $\mathbf{p}x \to yz\mathbf{q}$.

Variant *CPM4* : The instructions are of the form

$\mathbf{p}x \to \mathbf{q}$ $\quad$ $\mathbf{p}x \to y\mathbf{q}$ $\quad$ $\mathbf{p}x \to yx\mathbf{q}$.

Note that there are no universal circular Post machines with either only one state or one symbol, respectively [5].

Let us now define tag-systems. For a given positive integer m and a given alphabet $\Omega = \{s_1, \ldots, s_n, s_{n+1}\}$, a m-tag-system on Ω transforms the word w on Ω as follows: we delete the first m letters of w and we append to the right of the result a word that depends on the first letter of w. This process is iterated until m letters cannot be deleted or the first letter is s_{n+1}, and then stops. Formally, we have the following definitions.

A *tag-system* is a triple $\mathcal{T} = (m, \Omega, P)$, where m is a positive integer, $\Omega = \{s_1, \ldots, s_n,\ s_{n+1}\}$ is a finite alphabet, and P maps $\{s_1, \ldots, s_n\}$ into the set Ω^* of finite words on the alphabet Ω and s_{n+1} to $STOP$.

A tag-system $\mathcal{T} = (m, \Omega, P)$ is called m-tag-system when it is necessary to stress on number m. Words $P_i = P(s_i) \in \Omega^*$ are called the *productions* of the tag-system $\mathcal{T}$. The letter s_{n+1} is the *halting symbol*. Productions are often displayed as follows:

$$\begin{array}{rcl} s_i & \rightarrow & P_i, \quad i = 1, \ldots, n, \\ s_{n+1} & \rightarrow & STOP. \end{array}$$

A *computation* of the tag-system $\mathcal{T} = (m, \Omega, P)$ on a word $w \in \Omega^*$ is a sequence $w = v_0, v_1, \ldots,$ of words on Ω such that, for all nonnegative integer k, *a current word* v_k is transformed into v_{k+1} by deleting the first m letters of v_k and appending the word P_i to the result if the first letter of v_k is a_i. The computation stops in k steps if the length of v_k is less than m or the first letter of v_k is s_{n+1}. In that latter case, the result is v_k.

It was proved that there are universal 2-tag-systems ([9]), and therefore, we will deal only with 2-tag-systems. Due to the proof in [9, 10]) (see also [15]), we can restrict our attention to tag-systems which also have the following properties:

1. The computation of a tag-system stops only on a word beginning with the halting symbol s_{n+1}.

2. The productions $P_i,\ i = 1, \ldots, n$, are not empty.

3. The current word $v_k,\ k \geq 0$ contains at least 3 letters.

Henceforth, the tag-systems will be 2-tag-systems satisfying the above conditions.

3 Universal Circular Post Machines

In this part we present nine small universal machines of variant 0, with the halting state not included but represented by **H** in the program.

In the tables $y\mathbf{q}$ stands for $\mathbf{p}x \to y\mathbf{q}$, y for $\mathbf{p}x \to y\mathbf{p}$, $\mathbf{q}$ for $\mathbf{p}x \to \mathbf{q}$, $y\mathbf{q}x$ for $\mathbf{p}x \to y\mathbf{q}x$, and $\mathbf{H}$ for the unique halting state.

The machines are constructed by simulation of tag systems. Let the alphabet be $\Omega = \{s_1, \cdots, s_{n+1}\}$ with $s_H = s_{n+1}$. A symbol s_i is encoded in unary form by some number N_i, together with a separator. In a 2-tag instruction $s_i \to \alpha_i$ $(P_i = \alpha_i)$ with $\alpha_i = s_{i1} \cdots s_{im(i)}$ the symbols s_{ij} are encoded in the same way, with other separators.

UPM0(2,46)

	i	i_2	i_3	i_4	i_5	i_6	i_7
1	$i_2\mathbf{2}$	$\mathbf{2}$	i_4	$\mathbf{1}$	i_5	i_7	i_5
2	i_3	$a_4\mathbf{1}$	i	i_5	i_6		i

	c	c_2	c_3	c_4	c_5	c_6	c_7
1	$c_2\mathbf{2}$	$\mathbf{1}$	c_4	$\mathbf{2}$	c_5	c_7	c_5
2	c_3	$a_5\mathbf{1}$	c	c_5	c_6		c

	e	e_2	e_3	e_4	e_5	e_6	e_7	e_8
1	e_6	$e_3\mathbf{2}e_2$	i_5	$e_2\mathbf{2}$	c_5	$e_7\mathbf{2}$	$e_2\mathbf{2}$	e_5
2	e	e_4	e_8	e_6	c	e_2	$e\mathbf{1}$	

	a	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	a_2	a_5	a_4	a	a_6	a_5	a_8	a_5
2	a_3	a	a_3	a_4	$i_2\mathbf{1}$	a_7	$c_2\mathbf{1}$	a_4

	b	b_2	b_3	b_4	b_5	b_6	b_7	b_8
1	b_3	b_5	b_6	b_5	b	b_8	$d_5\mathbf{2}$	b_6
2	$b_2\mathbf{1}$	b_4	b	b_4	b_5	$b_7\mathbf{1}$	b_5	d_5

	d	d_2	d_3	d_4	d_5	d_6	d_7	d_8
1	d_2	d_3	d_3	d_3	d_6	b_6	d	d_4
2	$\mathbf{H}$	$d\mathbf{1}$	d_8	d_7	$d_6\mathbf{1}$	b_5		

$N_1 = 1$; $N_{k+1} = N_k + m_k + 2$ $(1 \leq k \leq n)$
Blank symbol: e_2
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$
Encoding of α_i : $ba^{N_{i1}}b \cdots ba^{N_{im(i)}}bb$

Separators : c, b, d

The initial configuration is
$bba^{N_{11}}b\cdots ba^{N_{1m(1)}}bbb\cdots bbba^{N_{n1}}b\cdots a^{N_{nm(n)}}bbd\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t}\cdots ci^{N_w}ce$

In the first stage i^{N_r} is read, N_r separators b's are changed into b_5's (actually we consider the first string of two b's as one b, and strings of three b's as strings consisting only of two separators b), and $i^{N_r}ci^{N_s}c$ is erased, giving

$b_5b_5a_4^{N_{11}}b_5a_4^{N_{12}}b_5\cdots b_5a_4^{N_{1m(1)}}b_5b_5b_5\cdots b_5b_5b_5a_4^{N_{(r-1)1}}b_5\cdots$
$b_5a_4^{N_{(r-1)m(r-1)}}b_5b_5b_5\ \mathbf{2}a_5^{N_{r1}}b_6\cdots a_5^{N_{rm(r)}}b_6b_6b_6\cdots b_6b_6b_6a_5^{N_{nm(n)}}b_6b_6$
$d_3i_5^{N_t}c_5\cdots c_5i_5^{N_w}c_5e_2$

In the second stage, starting with **2**, the part $a_5^{N_{r1}}b_6\cdots a_5^{N_{rm(r)}}b_6b_6$ is copied to the end of $i_5^{N_t}c_5\cdots i_5^{N_w}c_5e_2$ as $i_5^{N_{r1}}c_5\cdots c_5i_5^{N_{rm(r)}}c_5$ and the last b_6b_6 implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **2** encounters d.

The result is immediately after string dc_2 and before symbol e, and it is a string over $\{i_3, c_3\}$ where i_3 corresponds to i and c_3 corresponds to c.

UPM0(3,22)

	i	i_2	i_3	c	c_2	c_3	e	e_2	e_3	d	d_2	d_3
1	**2**	**1**	i_3	**3**	**3**	c_3		$e_3\mathbf{3}e_2$	i_3	d	d_2	d_2
2	i		i_3	c	$d\mathbf{1}$	c_3	e	**2**	$c_3\mathbf{1}$	**H**	d_3	$c_2\mathbf{3}$
3	i_2	i_3	i	c_2	c_3	c	e_2	e_2	$e\mathbf{2}$	$d_2\mathbf{1}$		

	a	a_2	a_3	a_4	a_5	b	b_2	b_3	b_4	b_5
1	a		a_3		a_3	b		a_3		b_3
2	a_2	a_2	a_5	a	a_4	$b_2\mathbf{1}$	b_2	b_5	b	b_4
3	a_3	a_4	$a_4\mathbf{1}$	a_4	$a_4\mathbf{1}$	b_3	b_4	$b_4\mathbf{2}$	b_4	$b_4\mathbf{2}$

$N_1 = 1$; $N_{k+1} = N_k + m_k + 1$ $(1 \leq k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: e_2
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$

Encoding of α_i : $a^{N_{i1}}b\cdots ba^{N_{im(i)}}bb$
Separators : c, b, d

The initial configuration is
$baa^{N_{11}}b\cdots ba^{N_{1m(1)}}bb\cdots bba^{N_{n1}}b\cdots a^{N_{nm(n)}}bbd\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t}\cdots ci^{N_w}ce$

In the first stage i^{N_r} is read, N_r separators b's are changed into b_4's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$b_4a_4a_4^{N_{11}}b_4a_4^{N_{12}}b_4\cdots b_4a_4^{N_{1m(1)}}b_4b_4\cdots b_4b_4a_4^{N_{(r-1)1}}b_4\cdots$
$b_4a_4^{N_{(r-1)m(r-1)}}b_4b_4\ \mathbf{3}a_3^{N_{r1}}b_3\cdots a_3^{N_{rm(r)}}b_3b_3\cdots b_3b_3a^{N_{nm(n)}}b_3b_3$
$d_2i_3^{N_t}c_3\cdots c_3i_3^{N_w}c_3e_2$

In the second stage, starting with $\mathbf{3}$, the part $a_3^{N_{r1}}b_3\cdots a_3^{N_{rm(r)}}b_3b_3$ is copied to the end of $i_3^{N_t}c_3\cdots i_3^{N_w}c_3e_2$ as $i_3^{N_{r1}}c_3\cdots c_3i_3^{N_{rm(r)}}c_3$ and the last b_3b_3 implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage $\mathbf{2}$ encounters d.

The result is immediately after string dc and before string ce.

UPM0(4,11)

	i	i_2	c	c_2	e	e_2	e_3	d	d_2	a	b
1	**2**	i_2	**3**	c_2		$i_2\mathbf{2}e_2$	c_2	d	d	a	b
2	i	i	c	c	e	$e_2\mathbf{3}$	4	**H**		i	$c\mathbf{1}$
3	**3**	i_2	**4**	c_2		$e_3\mathbf{2}e_2$				$i_2\mathbf{1}$	$c_2\mathbf{4}$
4	i_2	a	c_2	b	e_2	e	$d\mathbf{1}$	$d_2\mathbf{3}$	$e_3\mathbf{2}$	a	b

$N_1 = 1$; $N_{k+1} = N_k + m_k + 1$ $(1 \le k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: e_2
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$
Encoding of α_i : $a^{N_{i1}}b\cdots ba^{N_{im(i)}}bb$
Separators : c, b, d

The initial configuration is
$ba^{N_{11}}b\cdots ba^{N_{1m(1)}}bb\cdots bba^{N_{n1}}b\cdots a^{N_{nm(n)}}bbd\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t}\cdots ci^{N_w}e$

In the first stage i^{N_r} is read, N_r separators b's are changed into c_2's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$$c_2i_2^{N_{11}}c_2i_2^{N_{12}}c_2\cdots c_2i_2^{N_{1m(1)}}c_2c_2\cdots c_2c_2i_2^{N_{(r-1)1}}c_2\cdots$$
$$c_2i_2^{N_{(r-1)m(r-1)}}c_2c_2\ \mathbf{3}a^{N_{r1}}b\cdots a^{N_{rm(r)}}bb\cdots bba^{N_{nm(n)}}bb$$
$$d_2i_2^{N_t}c_2\cdots c_2i_2^{N_w}e_3e_2$$

In the second stage, starting with **3**, the part $a^{N_{r1}}b\cdots a^{N_{rm(r)}}bb$ is copied to the end of $i_2^{N_t}c_2\cdots i_2^{N_w}e_3e_2$ as $i_2^{N_{r1}}c_2\cdots c_2i_2^{N_{rm(r)}}$ and the last bb implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **2** encounters d.

The result is immediately after string dc and before symbol e.

UPM0(5,8)

	i	c	e	e_2	e_3	d	a	b
1	**2**	**3**		e_2**4**		d**2**	a	b
2	i	c	e	i**1**e_2	c	**H**	i	c**5**
3	**3**	**4**				d**4**	a	b
4	i	c	e_2	e_3**1**e_2	c**5**		i**1**	c**3**
5	a	b		e		d**1**	a	b

$N_1 = 1$; $N_{k+1} = N_k + m_k + 1$ $(1 \le k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: e_2
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$
Encoding of α_i : $a^{N_{i1}}b\cdots ba^{N_{im(i)}}bb$
Separators : c, b, d

The initial configuration is
$$ba^{N_{11}}b\cdots ba^{N_{1m(1)}}bb\cdots bba^{N_{n1}}b\cdots a^{N_{nm(n)}}bbd\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t}\cdots ci^{N_w}ce$$

In the first stage i^{N_r} is read, N_r separators b's are changed into c's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$$ci^{N_{11}}ci^{N_{12}}c\cdots ci^{N_{1m(1)}}cc\cdots cci^{N_{(r-1)1}}c\cdots$$
$$ci^{N_{(r-1)m(r-1)}}cc\ \mathbf{4}a^{N_{r1}}b\cdots a^{N_{rm(r)}}bb\cdots bba^{N_{nm(n)}}bb$$
$$di^{N_t}c\cdots ci^{N_w}ce_2$$

In the second stage, starting with **4**, the part $a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb$ is copied to the end of $i^{N_t}c \cdots i^{N_w}ce_2$ as $i^{N_{r1}}c \cdots ci^{N_{rm(r)}}c$ and the last bb implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **2** encounters d.

The result is immediately after string dc and before string ce.

UPM0(6,6)

	i	c	e	d	a	b
1	**2**	**3**		d**5**	a	b
2	i	c	e	**H**	i	c**6**
3	**3**	**4**	e**4**	d**4**	a	b
4	i	c	d**3**e	**6**	i**1**	c**3**
5	i	c	i**3**e	c		
6	a	b	e	d**1**	a	b

$N_1 = 1$; $N_{k+1} = N_k + m_k + 1$ $(1 \leq k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: e
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$
Encoding of α_i : $a^{N_{i1}}b \cdots ba^{N_{im(i)}}bb$
Separators : c, b, d

The initial configuration is
$ba^{N_{11}}b \cdots ba^{N_{1m(1)}}bb \cdots bba^{N_{n1}}b \cdots a^{N_{nm(n)}}bbd\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t} \cdots ci^{N_w}e$

In the first stage i^{N_r} is read, N_r separators b's are changed into c's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$ci^{N_{11}}ci^{N_{12}}c \cdots ci^{N_{1m(1)}}cc \cdots cci^{N_{(r-1)1}}c \cdots$
$ci^{N_{(r-1)m(r-1)}}cc\ \mathbf{4}a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb \cdots bba^{N_{nm(n)}}bb$
$di^{N_t}c \cdots ci^{N_w}de$

In the second stage, starting with **4**, the part $a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb$ is copied to the end of $i^{N_t}c \cdots i^{N_w}de$ as $i^{N_{r1}}c \cdots ci^{N_{rm(r)}}$ and the last bb implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **2** encounters d.

The result is immediately after string dc and before symbol e.

UPM0(8,5)

	i	c	e	a	b
1	**2**	**8**	e**2**	a	b
2	i	c	e**3**	i**5**	c**6**
3	i	c	**H**	i**3**	c**1**
4	i	c	c**1**e	a	b
5	i	c	i**1**e	a	b
6	a	b	b**1**	a**4**	e**7**
7	i	c	e**6**	a	b
8	**8**	**4**	e**2**		

$N_1 = 1$; $N_{k+1} = N_k + m_k + 1$ $(1 \leq k < n)$; $N_{n+1} = N_n + m_n + 2$

Blank symbol: e

Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$

Encoding of α_i : $a^{N_{i1}}b \cdots ba^{N_{im(i)}}bb$

Separators : c, b

The initial configuration is

$ba^{N_{11}}b \cdots ba^{N_{1m(1)}}bb \cdots bba^{N_{n1}}b \cdots a^{N_{nm(n)}}bb\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t} \cdots ci^{N_w}e$

In the first stage i^{N_r} is read, N_r separators b's are changed into c's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$ci^{N_{11}}ci^{N_{12}}c \cdots ci^{N_{1m(1)}}cc \cdots cci^{N_{(r-1)1}}c \cdots$
$ci^{N_{(r-1)m(r-1)}}cc\ \mathbf{2}a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb \cdots bba^{N_{nm(n)}}bb$
$i^{N_t}c \cdots ci^{N_w}ce$

In the second stage, starting with **2**, the part $a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb$ is copied to the end of $i^{N_t}c \cdots i^{N_w}ce$ as $i^{N_{r1}}c \cdots ci^{N_{rm(r)}}$ and the last bb implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **3** encounters e.

The result is immediately after string ccc and before symbol e.

UPM0(11,4)

	i	c	a	b
1	**2**	**B**	a	b
2	i	c	a**3**	
3	i	c	i	c**4**
4		**H**	a**1**	b**1**
5	i**9**		a	b
6	i**7**		a	b
7	i	c	b**B**a	**8**
8	a	b	a	b**1**
9	i	c	i**B**a	c**9**
A	i	c	i**5**	c**6**
B	**B**	**7**	a**A**	

$N_1 = 1$; $N_{k+1} = N_k + m_k + 1$ $(1 \leq k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: a
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}b$
Encoding of α_i : $a^{N_{i1}}b \cdots ba^{N_{im(i)}}bb$
Separators : c, b

The initial configuration is

$ba^{N_{11}}b \cdots ba^{N_{1m(1)}}bb \cdots bba^{N_{n1}}b \cdots a^{N_{nm(n)}}bbb\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t} \cdots ci^{N_w}a$

In the first stage i^{N_r} is read, N_r separators b's are changed into c's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$ci^{N_{11}}ci^{N_{12}}c \cdots ci^{N_{1m(1)}}cc \cdots cci^{N_{(r-1)1}}c \cdots$
$ci^{N_{(r-1)m(r-1)}}cc$ $\mathbf{A}$ $a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb \cdots bba^{N_{nm(n)}}bbb$
$i^{N_t}c \cdots ci^{N_w}ba$

In the second stage, starting with **A**, the part $a^{N_{r1}}b \cdots a^{N_{rm(r)}}bb$ is copied to the end of $i^{N_t}c \cdots i^{N_w}ba$ as $i^{N_{r1}}c \cdots ci^{N_{rm(r)}}$ and the last bb implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **4** encounters c.

The result is immediately after string $cccc$ and before symbol a.

UPM0(16,3)

	i	c	a
1	**2**	c	a
2	i**3**	**4**	a**8**
3	i	c	a**5**
4	**4**	**7**	
5	i	c**6**	i
6	i**5**	**H**	i**1**
7	i	c	c**2**a
8	i	c**B**	i**9**
9	i**A**	c	a
A	i	c	i**2**a
B	i		i**C**
C		c**E**	a**D**
D	i**7**	c	a
E	i**F**	c	a
F	i	c	a**G**
G	a	c	a**1**

$N_1 = 2$; $N_{k+1} = N_k + m_k + 1$ $(1 \leq k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: a
Encoding of symbol s_i: $i^{N_i}c$ and $a^{N_i}ca$
Encoding of α_i : $a^{N_{i1}}ca \cdots caa^{N_{im(i)}}caca$
Separators : ca, c

The initial configuration is

$caa^{N_{11}}ca \cdots caa^{N_{1m(1)}}caca \cdots cacaa^{N_{n1}}ca \cdots a^{N_{nm(n)}}cacc$
$\mathbf{1}i^{N_r}ci^{N_s}ci^{N_t} \cdots ci^{N_w}a$

In the first stage i^{N_r} is read, N_r separators ca's are changed into ci's, and $i^{N_r}ci^{N_s}c$ is erased, giving

$cii^{N_{11}}cii^{N_{12}}ci \cdots cii^{N_{1m(1)}}cici \cdots cicii^{N_{(r-1)1}}ci \cdots$
$cii^{N_{(r-1)m(r-1)}}cici\ \mathbf{8}a^{N_{r1}}ca \cdots a^{N_{rm(r)}}caca \cdots cacaa^{N_{nm(n)}}cacc$
$i^{N_t}c \cdots ci^{N_w}ca$

In the second stage, starting with **8**, the part $a^{N_{r1}}ca\cdots a^{N_{rm(r)}}caca$ is copied to the end of $i^{N_t}c\cdots i^{N_w}ca$ as $i^{N_{r1}}c\cdots ci^{N_{rm(r)}}$ and the last *caca* implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **6** encounters c.

The result is immediately after string *ccic* and before symbol a.

UPM0(34,2)

UPM0(34,2) models *UPM0(11,4)*. Symbols of *UPM0(34,2)* are $0, 1$ and states are letters. String 00 of *UPM0(34,2)* corresponds to symbol c of *UPM0(11,4)*, string 01 corresponds to symbol i, string 11 corresponds to symbol b, and string 10 corresponds to symbol a.

	A	**B**	**C**	**D**	**E**	**F**	**G**	**I**	**J**	**K**	**L**	**M**
0	**B**	**K**	0**D**	0**C**	0**F**	0**E**	1**F**	**H**	0**A**	**L**	**M**	0**N**
1	1**J**	**C**	1**E**	1**C**	1**F**	0**G**	0**I**	1**J**	1**A**		**K**	0**O**

	N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**
0	0**M**	0**P**0	1**Q**0	0**R**	0**Q**	1**T**	0**W**	0**T**	0**W**	0**V**	1**Y**0
1	1**M**		1**d**	1**R**	0**S**	0**P**	1**U**	1**T**	0**X**	1**V**	

	Y	**d**	**e**	**f**	**g**	**j**	**k**	**l**	**m**	**n**	**o**
0	1**Q**0	0**e**	0**N**	0**e**	0**k**	0**g**	0**n**	1**m**	1**l**	0**k**	0**l**
1		1**g**	1**f**	1**e**	1**j**	1**g**	1**n**	1**J**	0**l**	1**o**	

$N_1 = 2$; $N_{k+1} = N_k + m_k + 1$ $(1 \leq k < n)$; $N_{n+1} = N_n + m_n + 2$
Blank symbol: 0
Encoding of symbol s_i: $(01)^{N_i}00$ and $(10)^{N_i}11$
Encoding of α_i : $(10)^{N_{i1}}11\cdots 11(10)^{N_{im(i)}}1111$
Separators : 00, 11

The initial configuration is

$$11(10)^{N_{11}}11\cdots 11(10)^{N_{1m(1)}}1111\cdots 1111(10)^{N_{n1}}11\cdots$$
$$(10)^{N_{nm(n)}}111111\mathbf{A}(01)^{N_r}00(01)^{N_s}00(01)^{N_t}\cdots 00(01)^{N_w}10$$

In the first stage $(01)^{N_r}$ is read, N_r separators 11's are changed into 00's, and $(01)^{N_r}00(01)^{N_s}c$ is erased, giving

$00(01)^{N_{11}}00(01)^{N_{12}}00\cdots00(01)^{N_{1m(1)}}0000\cdots0000(01)^{N_{(r-1)1}}00\cdots$
$00(01)^{N_{(r-1)m(r-1)}}0000\ \mathbf{R}(10)^{N_{r1}}11\cdots11(10)^{N_{rm(r)}}1111\cdots$
$11(10)^{N_{nm(n)}}111111(01)^{N_t}00\cdots00(01)^{N_w}0010$

In the second stage, starting with **R**, the part

$$(10)^{N_{r1}}11\cdots(10)^{N_{rm(r)}}1111$$

is copied to the end of

$$(01)^{N_t}00\cdots(01)^{N_w}10$$

as

$$(01)^{N_{r1}}00\cdots00(01)^{N_{rm(r)}}$$

and the last 1111 implies a restoration of the tape, and a new cycle may start.

The machine stops if in the first stage **I** encounters 00.

The result is immediately after string **H**00 (where **H** is a halting state) and before string 10.

References

[1] Arbib, M. A. : *Theories of Abstract Automata.* Prentice Hall, Englewood Cliffs, 1969.

[2] Baiocchi, C. : *Three Small Universal Turing Machines.* Lecture Notes in Computer Science (LNCS), Springer, **2055** (2001) 1-10.

[3] Kudlek, M. : *Small deterministic Turing machines.* Theoretical Computer Science (TCS), Elsevier, **168-2** (1996) 241–255.

[4] Kudlek, M., Rogozhin, Yu. : *Small Universal Circular Post Machines.* Computer Science Journal of Moldova, **9**, no. 1(25) (2001), pp.34–52.

[5] Kudlek, M., Rogozhin, Yu. : *New Small Universal Circular Post Machines.* LNCS **2138** (2001) 217 – 227.

[6] Kudlek, M., Rogozhin, Yu. : *A Universal Turing Machine with 3 States and 9 Symbols.* Developments in Language Theory, LNCS **2295** (2002) 311–318

[7] Margenstern, M. : *Frontier between decidability and undecidability: a survey.* TCS **231-2** (2000) 217–251.

[8] Minsky, M. L. : *Recursive Unsolvability of Posts Problem of "tag" and Other Topics in the Theory of Turing Machines.* Annals of Math. **74** (1961) 437–454.

[9] Minsky, M. L. : *Size and Structure of universal Turing Machines Using Tag Systems.* In *Recursive Function Theory, Symposia in Pure Mathematics, AMS* **5** (1962) 229–238.

[10] Minsky, M. L. : *Computation: Finite and Infinite Machines.* Prentice Hall International, London, 1972.

[11] Pavlotskaya, L. : *Sufficient conditions for halting problem decidability of Turing machines.* Avtomaty i mashiny (Problemi kibernetiki), Moskva, Nauka **33** (1978) 91–118 (in Russian).

[12] Păun, G., Rozenberg, G., Salomaa, A. : *DNA Computing: New Computing Paradigms.* Springer, 1998.

[13] Păun, G. : *Membrane Computing. An Introduction.* Springer, 2002.

[14] Post, E. : *Formal Reduction of the General Combinatorial Decision Problem.* Amer. Journ. Math. **65** (1943) 197–215.

[15] Robinson, R. M. : *Minsky's Small Universal Turing Machine.* Intern. Journ. of Math. **2** no. 5 (1991) 551–562.

[16] Rogozhin, Yu. V. : *Small Universal Turing Machines.* TCS **168-2** (1996) 215–240.

[17] Rogozhin, Yu. : *A Universal Turing Machine with 22 States and 2 Symbols.* Romanian Journal of Information Science and Technology **1** no. 3 (1998) 259–265.

[18] Shannon, C. E. : *A Universal Turing Machine with Two Internal States.* In *Automata Studies*, Ann. Math. Stud. **3**4, Princeton Uni. Press (1956) 157–165.

A.Alhazov, M.Kudlek, Y.Rogozhin, Received October 4, 2002

Artiom Alhazov
Institute of Mathematics and Computer Science
of Moldovan Academy of Sciences,
str. Academiei 5, Chishinev,
MD-2028, Moldova.
E–mail: *artiom@math.md*

Manfred Kudlek
Fachbereich Informatik,
Universität Hamburg
E–mail: *kudlek@informatik.uni-hamburg.de*

Yurii Rogozhin
Institute of Mathematics and Computer Science
of Moldovan Academy of Sciences,
str. Academiei 5, Chishinev,
MD-2028, Moldova.
E–mail: *rogozhin@math.md*