

ИНГИБИТОРНАЯ СЕТЬ ПЕТРИ, ИСПОЛНЯЮЩАЯ ПРОИЗВОЛЬНУЮ ЗАДАННУЮ МАШИНУ ТЬЮРИНГА

Д.А. ЗАЙЦЕВ

Построена ингибиторная сеть Петри с фиксированной структурой, исполняющая произвольную заданную машину Тьюринга. Лента машины Тьюринга, ее программа и состояния зашифрованы маркировкой 10-ти выделенных позиций сети Петри. Правила работы машины Тьюринга закодированы одиночным потоком управления в сети Петри, скомпонованной из операторов последовательности, ветвления, цикла. Используются подсети, реализующие операции арифметики, сравнения, копирования.

ВВЕДЕНИЕ

Известно, что ингибиторная сеть Петри представляет собой универсальную алгоритмическую систему [1, 2]. Доказательство было получено на основе моделирования счетчиковой машины Минского ингибиторной сетью Петри [2] и предполагает индивидуальное кодирование программы каждой заданной машины Минского графом ингибиторной сети Петри.

Цель работы — построение ингибиторной сети с фиксированной структурой (графом), исполняющая произвольную заданную машину Тьюринга [3, 4] на основе шифрования машины Тьюринга маркировкой фиксированного числа позиций сети Петри.

Помимо указанной цели, при наличии обоих: IPNTM (Inhibitor Petri Net Turing Machine — ингибиторная сеть Петри, исполняющая машину Тьюринга) и TMIPN (Turing Machine Inhibitor Petri Net — машина Тьюринга, исполняющая ингибиторную сеть Петри) — их композиция дает новый способ построения универсальной ингибиторной сети Петри [5], а также универсальной машины Тьюринга [6]. Действительно, IPNTM (TMIPN) является универсальной ингибиторной сетью Петри: она принимает на вход шифр заданной ингибиторной сети Петри для машины Тьюринга TMIPN и затем исполняет машину Тьюринга TMIPN, зашифрованную для IPNTM. И наоборот, TMIPN (IPNTM) является универсальной машиной Тьюринга: она принимает на вход шифр заданной машины Тьюринга для ингибиторной сети Петри IPNTM и затем исполняет ингибиторную сеть Петри IPNTM, зашифрованную для TMIPN.

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Ингибиторная сеть Петри

Граф ингибиторной сети Петри [1, 2] является взвешенным двудольным ориентированным графом, представленным четверкой $G = (P, T, B, D)$, где $P = \{p_1, \dots, p_m\}$ — конечное число вершин именуемых позициями; $T = \{t_1, \dots, t_n\}$ — конечное число вершин именуемых переходами, а отобра-

жения $B: P \times T \rightarrow \mathbb{N} \cup \{-1\}$ и $D: T \times P \rightarrow \mathbb{N}$ задают входные и выходные дуги переходов вместе с их кратностью; \mathbb{N} — множество целых неотрицательных чисел; нулевое значение отображений — B, D — обозначает отсутствие дуги; ненулевое — кратность дуги, специальное значение -1 задает ингибиторную дугу. Отображения могут быть представлены соответствующими матрицами: $B = \|b_{i,j}\|, b_{i,j} = B(p_j, t_i)$ и $D = \|d_{i,j}\|, d_{i,j} = D(t_i, p_j)$.

Состояние сети именуется маркировкой и представлено отображением $Q: P \rightarrow \mathbb{N}$, которое задает количество динамических элементов — фишек внутри позиций сети. Ингибиторная сеть Петри (ИСП) [1, 2] — это пара $N = (G, Q_0)$, где G — граф сети, а Q_0 — ее начальная маркировка. Маркировка может быть представлена соответствующим вектором: $Q = \|q_j\|, q_j = Q(p_j)$. Таким образом, ингибиторная сеть Петри задана парой чисел, парой матриц и вектором $N = (m, n, B, D, Q_0)$. Динамика ингибиторной сети представляет собой пошаговый процесс изменения ее маркировки в результате срабатывания переходов и может быть описана уравнением состояний [5].

Позиции изображают окружностями с фишками в виде точек, размещенными внутри них, переходы — прямоугольниками. Для графического представления ингибиторной дуги используют полую окружность на конце дуги. Дуга с заполненной окружностью на конце обозначает пару дуг противоположного направления и равной кратности, она используется для проверки маркировки позиции (без ее изменения).

Машина Тьюринга

Машина Тьюринга (МТ) [3] — это шестерка $M = (X, Q, V, P, q_s, q_f)$, где X — конечный алфавит символов ленты, содержащий пустой символ λ ; Q — конечный алфавит внутренних состояний; q_s — начальное состояние; q_f — конечное (заключительное) состояние; $V = \{l, s, r\}$ — алфавит перемещений соответственно (влево, стоять, вправо); P — функция переходов (программа), представленная как:

- отображение для детерминированной машины Тьюринга

$$P: X \times Q \rightarrow X \times Q \times V;$$

- отношение для недетерминированной машины Тьюринга

$$P \subseteq (X \times Q) \times (X \times Q \times V).$$

Конструкция машины Тьюринга следующая:

1) бесконечная в обоих направлениях лента разделена на ячейки, содержащие символ алфавита X . Изначально все ячейки заполнены пустым символом λ ;

2) управляющая головка движется вдоль ленты, на текущем шаге обзревает одну текущую ячейку и находится во внутреннем состоянии q ;

3) программа (функция переходов) P задает переход машины Тьюринга на следующий шаг.

Правила перехода на следующий шаг:

а) останавливается, если $q = q_j$,

б) управляющая головка считывает символ x из текущей ячейки,

в) находит соответствующую команду $(q, x) \rightarrow (q', x', v)$ в P ,

г) записывает символ x' в текущую ячейку,

д) перемещается на одну ячейку влево, вправо, либо остается на месте в зависимости от значения v ,

е) переключается в следующее состояние $q := q'$ и продолжает по правилу (а).

Выбор команды в соответствии с правилом (в) уникален для детерминированной машины Тьюринга и множественен для недетерминированной машины Тьюринга [4]. Без ограничения общности рассмотрено только одно заключительное состояние. Определение машины Тьюринга с множеством заключительных состояний может быть преобразовано путем добавления команд перехода в единственное заключительное состояние без изменения текущего символа и перемещения головки.

Минимальную часть ленты, заполненную не пустыми символами, называют рабочей зоной. Обычно предполагают, что машина начинает (и завершает) свою работу в положении головки над крайней левой ячейкой рабочей зоны. В качестве начальной рабочей зоны слово α в алфавите X записано на ленте. После останова машины слово β , полученное в рабочей зоне, рассматривается как результат.

ШИФРОВАНИЕ МАШИНЫ ТЬЮРИНГА

В настоящем разделе представлено шифрование машины Тьюринга, включая ленту с позицией управляющей головки, текущее и заключительное состояния, программу, в форме маркировок десяти выделенных позиций сети IPNTM: $sL, x, sR, rX, q, qf, rQ, sP, rP, rV$. Шесть дополнительных позиций использованы для хранения текущей команды и ее компонентов: sI, Iq, Ix, Jq, Jx, Iv . Пример шифрования машины описан далее в подразделе «Пример шифрования машины Тьюринга».

Шифрование алфавитов и выделенных состояний МТ

Зашифруем символы алфавитов X, Q, V целыми неотрицательными числами диапазона от нуля до $|X| - 1, |Q| - 1, |V| - 1$ соответственно. При шифровании алфавита X будем предполагать, что пустой символ зашифрован нулем. Шифрование алфавита V следующее: $l \leftrightarrow 1, s \leftrightarrow 0, r \rightarrow 2$. Остальное шифрование произвольное.

Для работы с шифрами алфавитов использованы три позиции TPNTM с именами rX , rQ , rV содержащие следующие значения: $rX = |X|$, $rQ = |Q|$, $rV = |V|$.

Определение МТ содержит два выделенных состояния q_s, q_f , и во время работы МТ рассматривается текущее состояние q , совпадающее с начальным состоянием при запуске машины $q = q_s$. Выделенные состояния представлены двумя позициями IPNTM с именами q, qf , которые содержат шифры текущего и заключительного состояния соответственно; предполагается, что шифр q_s изначально загружен в позицию q .

Шифрование программы

Использовано рекуррентное шифрование и дешифрование векторов неотрицательных целых [5]:

$$s_j = s_{j-1}r + a_{m-1-j}, \quad s_0 = a_{m-1}, \quad j = \overline{1, m-1}, \quad (1)$$

$$a_j = s_{m-1-j} \bmod r, \quad s_{m-1-(j+1)} = s_{m-1-j} \operatorname{div} r, \quad s_{m-1} = s, \quad j = \overline{0, m-1}, \quad (2)$$

где шифр вектора A длины m равен $s = s_{m-1}$, а основание шифрования $r = \max_j a_j + 1$.

Для шифрования программа (функция переходов) P рассматривается как множество команд (инструкций) $P = \{I\}$ в форме $I = (q, x, q', x', v)$, компонентами которых являются шифры состояний, символов текущей ячейки, движения головки. Заметим, что недетерминированная МТ может содержать несколько команд с одинаковой парой (q, x) .

Команда зашифрована как вектор (1) с переменной величиной основания $r : (rV, rQ, rX, rQ, rX)$. Заметим, что rV реально не используется, так как вычисляется ее нулевая степень. Выраженное в явной форме шифрование имеет вид:

$$sI = q + xrQ + q'(rQrX) + x'(rQrXrQ) + v(rQrXrQrX). \quad (3)$$

Выполненные построения дают неоднозначную интерпретацию нуля как шифра инструкции $(q_0, x_0, q_0, x_0, v_0)$, что может быть неудобным при неизвестной длине программы. Либо указанная фиктивная команда должна быть исключена из рассмотрения, либо некоторые шифры (например V) должны начинаться с единицы.

Множество команд занумеровано в произвольном порядке, начиная с нуля до $k = |P| - 1$, для использования простого алгоритма последовательного поиска. Таким образом, программа представлена как вектор $P = (I_0, I_1, \dots, I_{k-1})$. Для ее шифрования как вектора (1) следует выбрать значение требуемого основания. Максимальный возможный шифр команды следующий:

$$sI_{\max} = (rQ - 1) + (rX - 1)rQ + (rQ - 1)(rQrX) + (rX - 1)(rQrXrQ) + \\ + (rV - 1)(rQrXrQrX).$$

Таким образом, основание для шифрования программы имеет значение

$$rP = sI_{\max} + 1. \tag{4}$$

Итак, программа МТ представлена двумя позициями IPNTM с именами sP, rP , которые содержат значения sP, rP соответственно. Дополнительные позиции использованы для хранения текущей команды $I = (q, x, q', x', v)$ и ее компонентов sI, Iq, Ix, Jq, Jx, Iv соответственно (для последовательного поиска подходящей команды).

Шифрование ленты

Шифрование рабочей зоны ленты может быть выполнено также, как и шифрование программы, но в этом случае необходима дополнительная информация о текущей позиции управляющей головки.

Заметим, что рекуррентное шифрование (1) и дешифрование (2) задают дисциплину стека (LIFO). Начнем с нулевого значения $s = 0$, тогда операции

$$\text{push}(x, s) = \text{MUL_ADD}(s, r, x) :: s := s \cdot r + x,$$

$$\text{pop}(s) = \text{MOD_DIV}(s, r, x) :: x := s \bmod r, s := s \text{ div } r$$

обеспечивают дисциплину стека, и достижение дна стека распознается как условие $s = 0$. Неоднозначность шифрования в случае, когда нулевой шифр представляет пустой символ λ , является полезной. Последовательное дешифрование нуля дает неограниченное количество символов λ на дне стека. «Бездонное дно» дает возможность представить две части ленты по обеим сторонам текущей ячейки как два стека.

Символы рабочей зоны ленты занумерованы следующим образом:

$$\dots, \lambda, L_{m-1}, L_{m-2}, \dots, L_0, x, R_0, R_1, \dots, R_{n-1}, \lambda, \dots,$$

где L_i — символы левой части, а R_j — символы правой части ленты относительно символа текущей ячейки x . Обе части ленты зашифрованы как векторы L, R в соответствии с (1). Полученные шифры обозначены как sL, sR соответственно.

Лента и текущая позиция управляющей головки зашифрованы как три неотрицательных целых числа $sT = (sL, x, sR)$, где sL — шифр левой части рабочей зоны, sR — шифр правой части рабочей зоны, x — шифр символа текущей ячейки. Основание для шифрования соответствует алфавиту ленты $r = rX$.

Описанное шифрование обеспечивает следующую реализацию перемещений управляющей головки:

$$\text{left}(sL, x, sR) :: \text{push}(x, sR), x := \text{pop}(sL),$$

$$\begin{aligned} \text{stay}(sL, x, sR) &:: \langle \text{без изменений} \rangle, \\ \text{right}(sL, x, sR) &:: \text{push}(x, sL), x := \text{pop}(sR). \end{aligned} \quad (5)$$

Итак, лента и позиция управляющей головки представлены тремя позициями IPNTM с именами sL, x, sR , которые содержат значения sL, x, sR соответственно.

Пример шифрования машины Тьюринга

Построим машину Тьюринга MTI , которая переводит числа из унарной в бинарную систему счисления: $X = \{\lambda, a, 0, 1\}$, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $q_s = q_0$, $q_f = q_5$; в унарной системе счисления символ a использован для представления числа.

Шифры алфавитов, программы и процесса работы MTI представлены в табл. 1, 2, 3 соответственно; программа не является полностью определенной: для краткости опущены команды для некорректных конфигураций.

Таблица 1. Шифрование алфавитов MTI

Алфавит X				Алфавит Q						rX	rQ	rP
λ	a	0	1	q_0	q_1	q_2	q_3	q_4	q_5			
0	1	2	3	0	1	2	3	4	5	4	6	1728

Таблица 2. Шифрование программы MTI

№	Команда	sI	sP
0	0 1 0 1 2	1302	1302
1	0 2 0 2 2	1452	2251308
2	0 3 0 3 2	1602	3890261826
3	0 0 1 0 1	600	6722372435928
4	1 1 2 0 1	631	11616259569284215
5	1 2 4 2 1	973	20072896535723124493
6	1 3 4 3 1	1123	34685965213729559125027
7	2 1 2 1 1	776	59937347889324678168047432
8	2 2 0 3 2	1598	103571737152753043874385964094
9	2 0 0 3 2	1586	178971961799957259814938945956018
10	2 3 3 2 1	956	309263549990326144960214498612000060
11	3 3 3 2 1	957	534407414383283578491250653601536104637
12	3 2 0 3 2	1599	923456012054314023632881129423454388814335
13	3 0 0 3 2	1587	1595731988829854632837618591643729183871172467
14	4 2 4 2 1	976	2757424876697988805543404926360364029729386023952
15	4 3 4 3 1	1126	4764830186934124655979003712750709043372379049390182
16	4 0 5 0 2	1276	8233626563022167405531718415633225226947470997346235772

Таблица 3. Шифрование процесса работы МТИ

Шаг	Лента	Шифр ленты			Сост. <i>q</i>	№ ком.	Шаг	Лента	Шифр ленты			Сост. <i>q</i>	№ ком.
		<i>sL</i>	<i>x</i>	<i>sR</i>					<i>sL</i>	<i>x</i>	<i>sR</i>		
0	aaaa ^	0	1	21	0	0	20	10aa ^	sL	x	sR	0	0
1	aaaa ^	1	1	5	0	0	21	10aa ^	57	1	0	0	0
2	aaaa ^	5	1	1	0	0	22	10aaλ ^	229	0	0	0	3
3	aaaa ^	21	1	0	0	0	23	10aa ^	57	1	0	1	4
4	aaaaλ ^	85	0	0	0	3	24	10aλ ^	14	1	0	2	7
5	aaaa ^	21	1	0	1	4	25	10a ^	3	2	1	2	8
6	aaaλ ^	5	1	0	2	7	26	11a ^	15	1	0	0	0
7	aaa ^	1	1	1	2	7	27	11aλ ^	61	0	0	0	3
8	aaa ^	0	1	5	2	7	28	11a ^	15	1	0	1	4
9	λaaa ^	0	0	21	2	9	29	11λ ^	3	3	0	2	10
10	1aaa ^	3	1	5	0	0	30	10 ^	0	3	2	3	11
11	1aaa ^	13	1	1	0	0	31	λ00 ^	0	0	10	3	13
12	1aaa ^	53	1	0	0	0	32	100 ^	3	2	2	0	1
13	1aaaλ ^	213	0	0	0	3	33	100 ^	14	2	0	0	1
14	1aaa ^	53	1	0	1	4	34	100λ ^	58	0	0	0	3
15	1aaλ ^	13	1	0	2	7	35	100 ^	14	2	0	1	5
16	1aa ^	3	1	1	2	7	36	100 ^	3	2	2	4	14
17	1aa ^	0	3	5	2	10	37	100 ^	0	3	10	4	15
18	λ0aa ^	0	0	22	3	13	38	λ100 ^	0	0	43	4	16
19	10aa ^	3	2	5	0	1	39	100 ^	0	3	10	5	

КОДИРОВАНИЕ АЛГОРИТМОВ ИНГИБИТОРНОЙ СЕТЬЮ ПЕТРИ

Сети Петри известны как форма представления параллельных (concurrent) алгоритмов и вычислений, управляемых потоками данных (data flow) [2, 7, 8]. В большинстве случаев, алгоритмы моделируются сетями Петри, что означает определенный акт абстрагирования, приводящий к утрате некоторых особенностей. Такое абстрагирование обычно оправдано общей целью исследования, например, поиском взаимных блокировок (тупиков), верификацией протокола и т.п.

В настоящей работе внимание сконцентрировано на точной спецификации одного потока управления, который описывает правила работы МТ. Для этих целей конструкции, рассмотренные в [1, 2], специфицированы более строго с точным разделением элементов сети Петри на две категории: для описания переменных и для описания одного потока управления. Указанное разделение выполнено до определенных границ: выбраны подсети, которые представляют множество базисных операций, необходимых для выполнения МТ. Для этих подсетей представлено доказательство их корректности в явном виде.

Каждый оператор (операция, процедура) представлен подсетью вида, изображенного на рис. 1. Для передачи параметров использованы контактные позиции, которые разделены на входные и выходные. Две выделенные позиции s и f представляют поток управления. Для обеспечения повторного прохождения потока управления через операторы, примем следующие соглашения: все внутренние позиции имеют нулевую маркировку; перед запуском оператора все входные переменные копируются во входные позиции оператора; работа оператора запускается попаданием фишки в позицию s ; оператор завершает свою работу при попадании фишки в позицию f ; при завершении работы оператора все его позиции пусты за исключением выходных позиций, которые содержат результат.

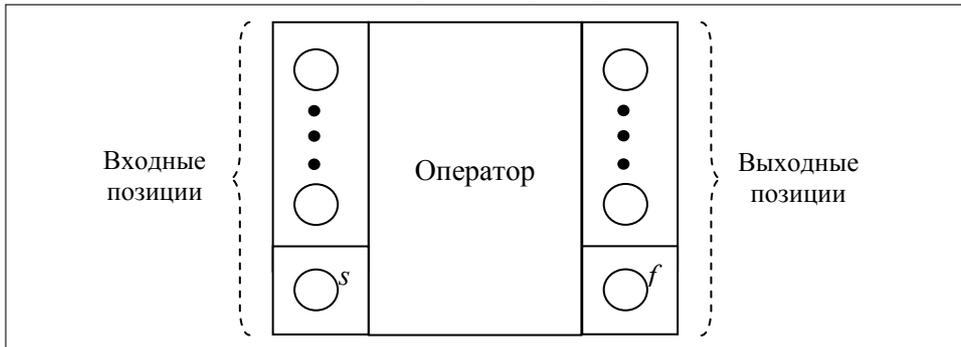


Рис. 1. Представление оператора

Рассматривается два вида переменных: глобальные статические и параметры операторов. Композиция статических переменных и параметров выполняется как совмещение (объединение) соответствующих позиций. Композиция (синхронизация [1]) потока управления и переменных обеспечивается парой выделенных позиций s и f , которые совмещаются при суперпозиции операторов — f предыдущего совмещается с s следующего.



Рис. 2. Представление алгоритма

Каждая статическая глобальная переменная представлена соответствующей позицией сети Петри (рис. 2). Поток управления моделируется

трасой прохождения одной фишки из начальной позиции $start(s)$ в завершающую позицию $finish(f)$.

Введены пунктирные дуги [5], обозначающие следующие дополнительные правила формирования значений входных и выходных позиций: при запуске содержимое переменной копируется во входную переменную оператора; после завершения переменная очищается, и в нее перемещается значение выходной позиции оператора. В случае нескольких переменных строятся цепочки *COPY* для последовательного копирования входных переменных и цепочки *CLEAN, MOVE* для перемещения значений выходных переменных. Последовательность *CLEAN, MOVE* обозначена как *ASSIGN*.

Лемма 1. Алгоритмические управляющие структуры могут быть закодированы ингибиторной сетью Петри в форме, представленной на рис. 3.

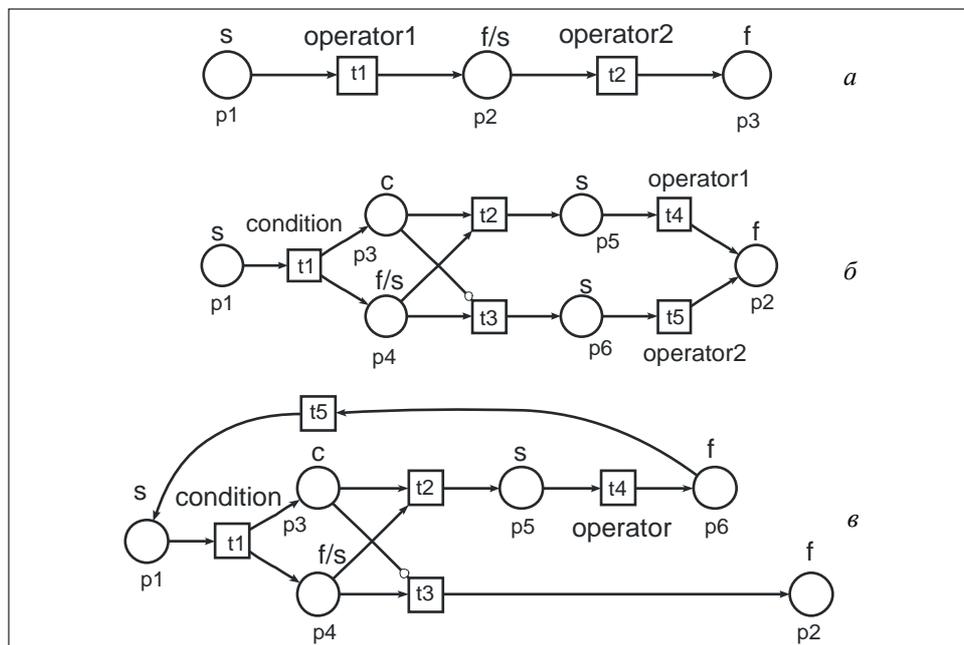


Рис. 3. Кодирование алгоритмических управляющих структур, где a — последовательность, b — ветвление, c — цикл (while)

Доказательство. Представим доказательство для структуры ветвления (рис. 3, б). В начальной маркировке $p_1 = 1$ разрешен только переход t_1 , который срабатывает и помещает значение условия в позицию p_3 и фишку потока управления в позицию p_4 . Допустим, что истинное значение представлено единичной маркировкой, а ложное — нулевой маркировкой. Поэтому только один из переходов t_2, t_3 разрешен, а именно: t_2 в единичной маркировке p_3 и t_3 в нулевой маркировке. При срабатывании t_2 фишка потока управления перемещается в позицию p_5 , что запускает переход t_4 , а при срабатывании t_3 фишка потока управления перемещается в позицию p_6 , что запускает переход t_5 . В итоге фишка потока управления попадает в позицию p_2 , а все другие позиции пусты. Следовательно, разрешены

только две последовательности срабатывания $t_1 t_2 t_4$ или $t_1 t_3 t_5$, что означает выбор операторов *operator1* или *operator2* после вычисления условия *condition*; *operator1* выбран, когда значение условия истинно, а *operator2* — когда ложно, что соответствует семантике структуры ветвления. □

Заметим, что в соответствии с рис. 3, а, суперпозиция операторов при кодировании алгоритма осуществляется совмещением выходной позиции f первого оператора с входной позицией s второго оператора.

Известны примеры представления основных алгебраических и логических операций сетями Петри [2, 9, 10]. В некоторых случаях удобно непосредственное представление наиболее часто выполняемых действий, таких как, например, *MOD_DIV* и *MUL_ADD* для дешифрования и шифрования МТ. В построениях IPNTM использованы вспомогательные сети, изученные в [5].

КОМПОЗИЦИЯ IPNTM

Закодируем правила работы МТ рассмотренные в подразделе «Машина Тьюринга» ингибиторной сетью Петри в соответствии с принципами, описанными в разделе «Кодирование алгоритмов ингибиторной сетью Петри» и шифрованием МТ, описанным в разделе «Шифрование машины Тьюринга». Заметим, что леммы 1, 2 перечисляют все требуемые управляющие структуры и операции. Получена сеть IPNTM, представленная на рис. 4.

При построении IPNTM выполнено последовательное соединение подсетей, реализующих правила (а)–(е) работы МТ, и организован основной цикл повторения шага. Подсеть EQ проверяет достижение заключительного состояния (правило (а)); подсеть FIND_I выполняет поиск подходящей команды (правила (б)–(в)); подсеть EXEC_I реализует выполнение найденной команды (г)–(е); ветвление после FIND_I реализует аварийный останов в случае отсутствия подходящей команды, подсеть FIND_I представляет собой суперпозицию цикла, двух ветвлений и последовательностей; подсеть EXEC_I представляет собой суперпозицию двух ветвлений и последовательностей для выбора одного из трех вариантов перемещения головки.

Для представления переменных использованы совмещенные позиции: все позиции с одинаковыми именами логически являются одной и той же позицией; совмещенные позиции облегчают графическое представление сети. Предполагаем, что перед запуском сети IPNTM шифр целевой (исполняемой) МТ загружен в выделенные позиции, а после останова сети IPNTM шифр полученной МТ (ее ленты) считан из соответствующих позиций.

Пунктирные линии обозначают рассмотренные в разделе «Кодирование алгоритмов ингибиторной сетью Петри» соглашения по копированию входных и выходных переменных. Двухнаправленные дуги использованы для работы с переменными, которые являются как входными, так и выходными. В этом случае копирование может быть оптимизировано двукратным применением *MOVE* без очистки. В некоторых случаях для копирования входной переменной вместе с ее очисткой целесообразно использовать *MOVE* вместо *COPY*. В качестве соответствующего обозначения использована линия с точечным пунктиром. Подстановка перехода подразумевает копиро-

вание соответствующей подсети с совмещением контактных позиций. В общем случае подстановка перехода требует указание отображения входных и выходных позиций. В приведенных сетях отображение задано неявно контекстом использованных операций и не указано дополнительно.

Теорема 1. Сеть IPNTM исполняет произвольную заданную детерминированную машину Тьюринга.

Доказательство. Докажем, что работа IPNTM полностью соответствует правилам работы МТ, описанным в подразделе «Машина Тьюринга» относительно шифрования МТ, описанного в разделе «Шифрование машины Тьюринга». Сеть IPNTM использует три подсети $FIND_I$, DA_I , $EXEC_I$ для нахождения подходящей команды, разбора (дизассемблирования) команды, исполнения команды соответственно, а также операции, представленные в [5] сетями Петри.

Во-первых, сеть IPNTM (рис. 4, а) реализует требуемую последовательность правил (а)–(е): в цикле (начинающемся с p_3) она сравнивает (EQ) текущее состояние q с заключительным состоянием qf и выходит из цикла (t_7) в случае совпадения в соответствии с правилом (а); чтение текущего символа по правилу (б) не требуется, поскольку символ текущей ячейки ленты (его шифр) выделен в позиции x в соответствии с шифрованием МТ. Затем она находит подходящую команду ($FIND_I$) и исполняет ее ($EXEC_I$). Для проверки возможных ошибок добавлены дополнительные позиции, названные $found$ и noi . В случае отсутствия подходящей команды подсеть $FIND_I$ не помещает фишку в позицию $found$ и IPNTM выходит из цикла с фишкой в позиции noi (нет команды).

Требуется доказать, что сеть $FIND_I$ находит подходящую команду в соответствии с правилом (в), и сеть $EXEC_I$ исполняет найденную команду в соответствии с правилами (г)–(е).

Сеть $FIND_I$ копирует шифр программы во вспомогательную позицию $sP1$ и обрабатывает $sP1$ в цикле (начинающемся с p_3). Имеется два возможных выхода из цикла: позитивный (t_{11}) — найдена подходящая команда, негативный (t_2) — все команды были проанализированы и подходящая команда не найдена ($sP1 == 0$). В цикле, операция MOD_DIV извлекает текущую команду из шифра программы $sP1$, разбирает текущую команду, используя подсеть DA_I , и сравнивает последовательно текущее состояние и текущий символ команды с текущим состоянием головки q и текущим символом ленты x . В случае любого несовпадения ($Iq \neq q$ или $Ix \neq x$) она возвращается к началу цикла (t_9 или t_{12}). Когда и состояние, и символ команды совпадают с текущими значениями в соответствии с правилом (с), она выходит из цикла (t_{11}) с размещением фишки в позиции $found$. Заметим, что разобранные компоненты найденной команды сохраняются в позициях Iq, Ix, Jq, Jx, Iv .

Сеть DA_I разбирает команду в соответствии с принципами, описанными в подразделе «Шифрование ленты»: она применяет четыре последовательных операции MOD_DIV с различными значениями основания r . Заметим, что в соответствии с (4) не требуется применять MOD_DIV пять раз, так как четвертое деление дает Iv , а остаток — Jx .

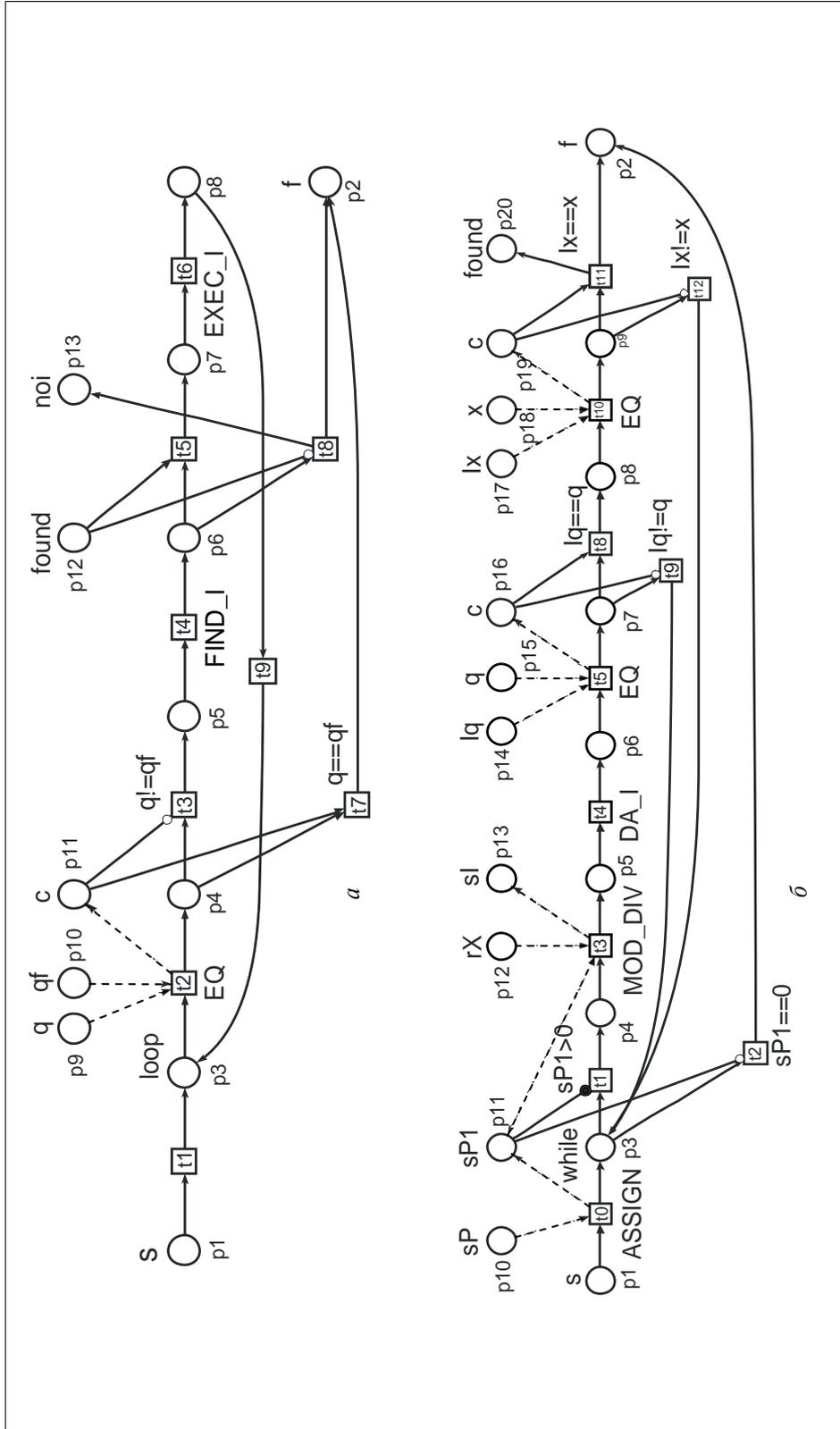


Рис. 4. Ингибиторная сеть Петри IPNTM, исполняющая машину Тьюринга, где a — IPNTM, b — FIND_I

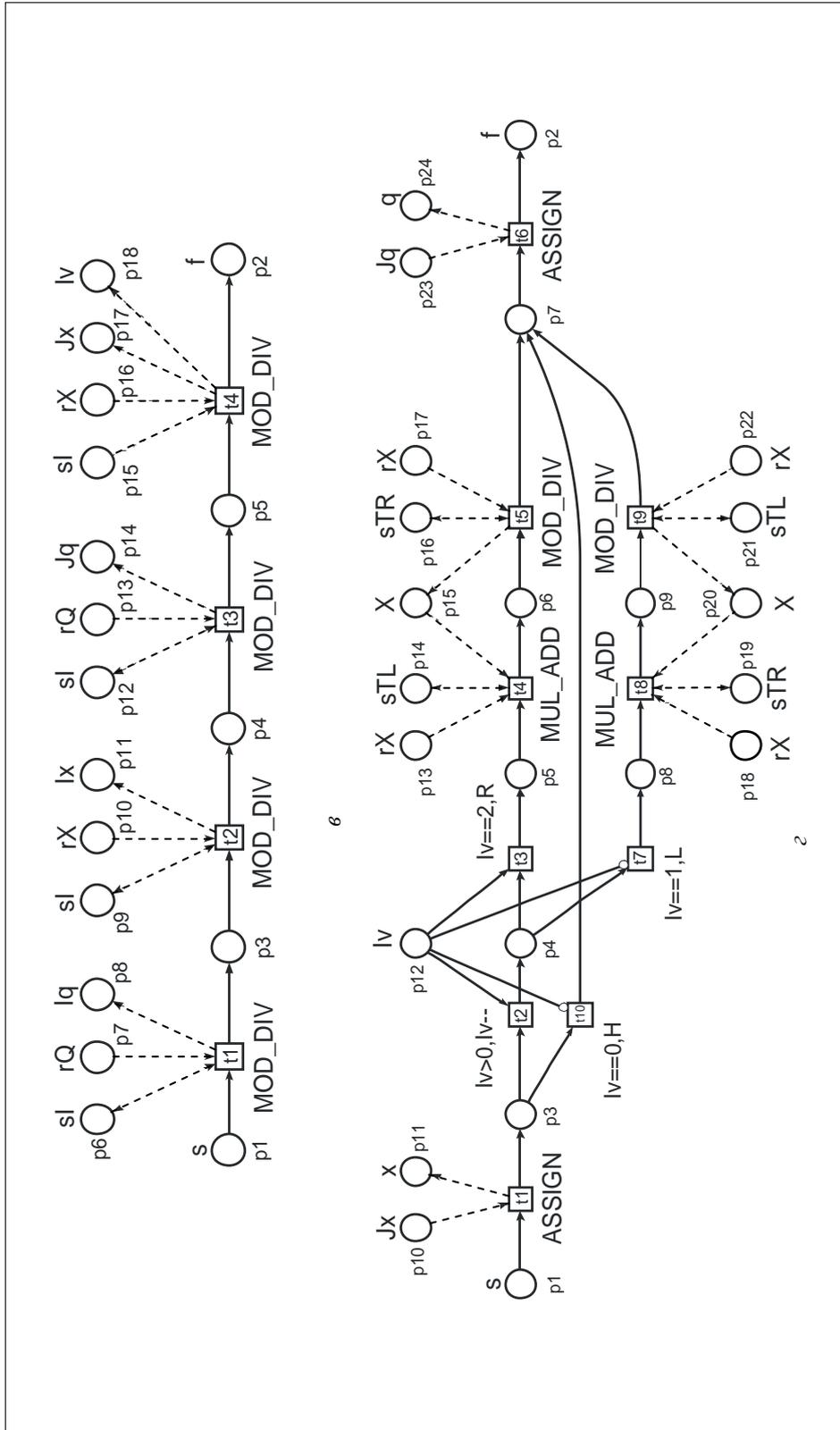


Рис. 4. Ингибиторная сеть Петри IPNTM, исполняющая машину Тьюринга, где ϵ — DA_1, z — EXEC_1 (Продолжение рис. 4)

Сеть $EXEC_I$ реализует последовательно правила (г)–(е): она присваивает новое значение текущему символу ленты $x = Jx$; осуществляет перемещение lv головки в соответствии с принципами шифрования ленты (5) посредством операций MUL_ADD и MOD_DIV в требуемой последовательности и не изменяет значения в случае $lv = 0$. Затем она переключается в следующее состояние $q = Jq$.

Итак, работа IPNTM полностью соответствует правилам работы МТ и способу кодирования МТ ингибиторной сетью Петри. Когда IPNTM останавливается корректно ($noi = 0$), шифр результата (рабочей зоны ленты) содержится в позициях sL , x , sR . В соответствии с обычными соглашениями машина останавливается в позиции управляющей головки над крайним левым символом рабочей зоны, следовательно, $sL = 0$ и результат считывается из позиций x , sR . □

В описанном способе шифрования МТ имеется лишь незначительное отличие между детерминированной и недетерминированной МТ, которое состоит в том, что недетерминированная МТ может содержать несколько команд с совпадающей парой (q, x) . При нахождении подходящей команды следует применить недетерминированный выбор команды из множества команд с совпадающей парой (q, x) . Для этой цели сеть $FIND_I$ преобразована в сеть $FIND_I_ND$. Сеть IPNTMND получена из сети IPNTM подстановкой подсети $FIND_I_ND$ вместо $FIND_I$ (рис. 5).

Теорема 2. Сеть IPNTMND исполняет произвольную заданную недетерминированную машину Тьюринга.

Доказательство. Так как различие между детерминированной и недетерминированной МТ состоит в выборе подходящей команды, который реализован подсетью $FIND_I$ и вследствие теоремы 1, следует доказать, что использование подсети $FIND_I_ND$ в сети IPNTMND вместо подсети $FIND_I$ обеспечивает недетерминированный выбор подходящей команды.

В соответствии со способом шифрования программы (подраздел «Шифрование ленты») отношение P рассматривается как множество команд, представленное для шифрования как вектор в произвольном порядке. Таким образом, следует найти все команды, содержащие одинаковую пару (q, x) и выполнить из них недетерминированный выбор.

Дальнейший процесс доказательства аналогичен доказательству теоремы 1. □

Заметим, что сеть IPNTMND является более общей, так как она может исполнять как недетерминированную, так и детерминированную МТ, что зависит от уникальности пар (q, x) множества команд программы. Построенные сети IPNTM и IPNTMND представлены покомпонентно в соответствии с использованными операциями и принципами работы с переменными. Определенный интерес представляет компоновка этих сетей в виде единой ИСП и ее исполнение в среде некоторой моделирующей системы, которая имитирует динамику срабатывания переходов.

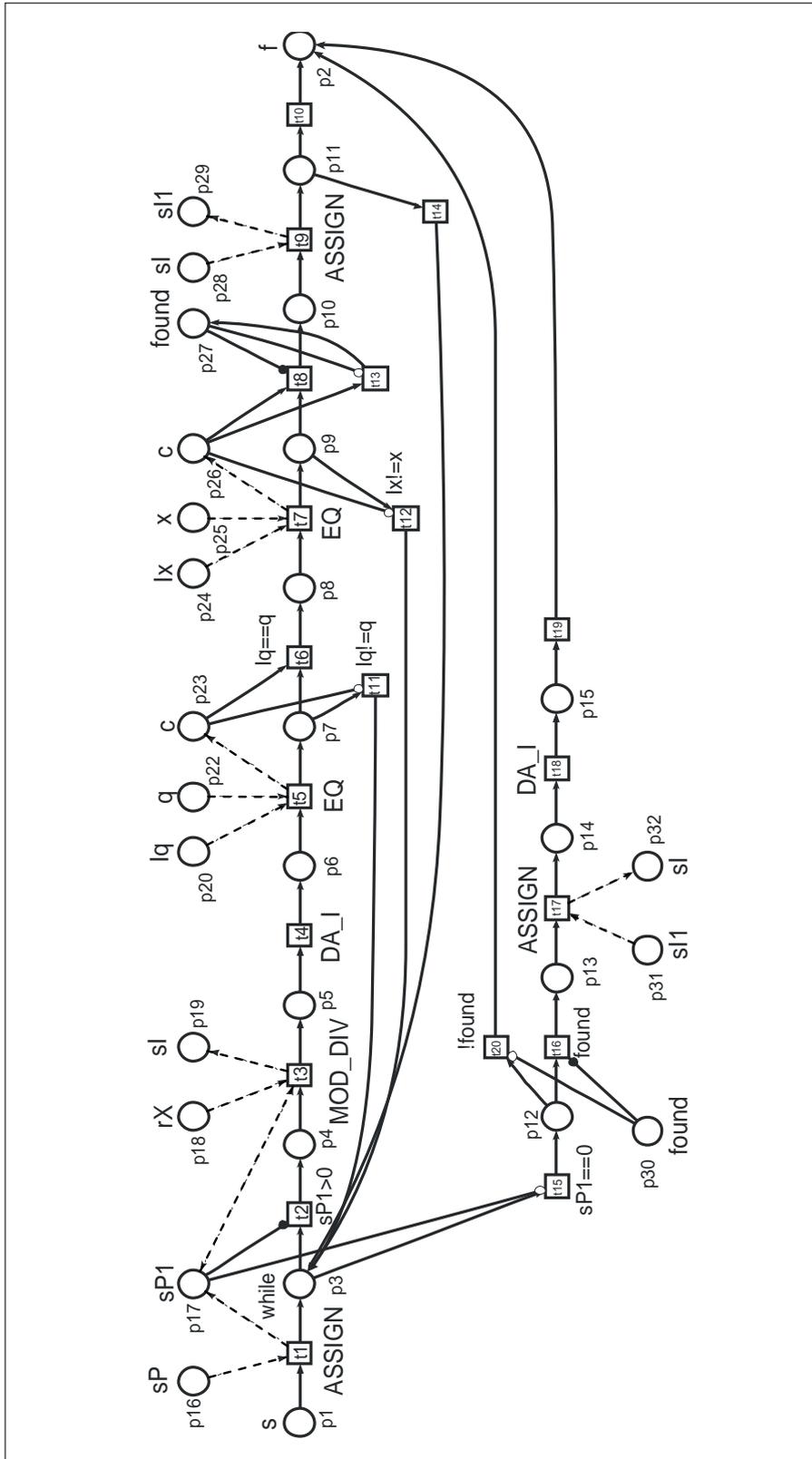


Рис. 5. Недетерминированный выбор подходящей команды FIND_I_ND

ВЫВОДЫ

В настоящей работе построена ингибиторная сеть Петри с фиксированной структурой, которая исполняет произвольную заданную машину Тьюринга как детерминированную, так и недетерминированную.

Композиция ингибиторной сети, исполняющей машину Тьюринга, и машины Тьюринга, исполняющей ингибиторную сеть Петри, дает новый способ представления универсальной ингибиторной сети [5], которая исполняет произвольную заданную ингибиторную сеть Петри.

Возможны построения аналогичных сетей в других классах сетей Петри, которые являются универсальной алгоритмической системой [2]: приоритетных, синхронных, временных сетях Петри.

Известны примеры построения универсальной машины Тьюринга с минимальным числом использованных символов/состояний [11, 12]. В этой связи определенный интерес представляет построение сети с минимальным количеством позиций (переходов), минимальным значением маркировки.

ЛИТЕРАТУРА

1. *Agerwala T.A.* Complete Model for Representing the Coordination of Asynchronous Processes. — Baltimore: John Hopkins University, Hopkins Computer Science Program, Res. Rep., July — № 32. — 1974. — 87 p.
2. *Котов В.Е.* Сети Петри. — М.: Наука, 1984. — 160 с.
3. *Turing A.M.* On Computable Numbers with an Application to the Entscheidungsproblem // Proceedings of the London Mathematical Society. — 1936. — 42. — P. 230–265.
4. *Hopcroft J.E., Motwani R., Ullman J.D.* Introduction to automata theory, languages, and computation, second edition. — NY: Addison-Wesley, 2001. — 498 p.
5. *Zaitsev D.A.* Universal Inhibitor Petri Net // Proceedings of the 17-th German Workshop on Algorithms and Tools for Petri Nets, 7–8 October. — Cottbus, Germany. — 2010. — P. 1–15. — <http://ceur-ws.org/Vol-643/>.
6. *The Universal Turing Machine. A Half-Century Survey* / Rolf Herken (ed.). — Springer-Verlag, Wien New York. — 1994. — 609 p.
7. *Best E., Devillers R., Koutny M.* Process Algebra and Concurrent Programming Languages // Lecture Notes in Computer Science. — **1492**: Lectures on Petri Nets II: Applications / Reisig W.; Rozenberg G. (eds.). — 1998. — P. 157–198.
8. *Goltz U.* On Representing CCS Programs by Finite Petri Nets // Lecture Notes in Computer Science: Proc. MFCS. — 1988. — **324**. — P. 339–350.
9. *Слепцов А.И.* Уравнения состояний и эквивалентные преобразования нагруженных сетей Петри (алгебраический подход) // Формальные модели параллельных вычислений: докл. и сообщ. Всесоюз. конф. — Новосибирск. — 1988. — С. 151–158.
10. *Слепцов А.И., Юрасов А.А.* Автоматизация проектирования управляющих систем гибких автоматизированных производств / Под ред. Б.Н. Малиновского. — К.: Техніка, 1986. — 160 с.
11. *Minsky M.* Size and structure of universal Turing machines using tag systems // Recursive Function Theory, Symposium in Pure Mathematics, AMS. — 1962. — **5**. — P. 229–238.
12. *Rogozhin Y.* Small universal Turing machines // Theoretical Computer Science. — 1996. — 168, № 2. — P. 215–240.

Поступила 29.07.2010