С. С. АГАЯН, С. Р. ВАРТАНОВ

ПРЕДСТАВЛЕНИЕ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В СИСТЕМАХ КЛЕТОЧНЫХ ПРЕОБРАЗОВАНИЙ

1. Введение

Представленная работа посвящена задаче приведения описаний алгоритмов в технологии последовательного программирования к системам клеточных преобразований, функционирующих на параллельной основе.

Предлагаемое здесь решение этой задачи основано на принципах распараллеливания схем программ. В целом проблематика параллельных вычислений и распараллеливания схем программ в существенной степени была охвачена работами Карпа и Миллера [1], Котова и Нариньяни [2], Келлера [3]. В то же время определенная фон Нейманом [4] и развитая в последующих работах Мура [5] и других авторов [6,7] клеточная логика (КЛ), опиравшаяся на прикладные задачи моделирования в биологии, получила широчайшее развитие в вопросах управления многопроцессорными ассоциациями (примером служит ЭВМ типа ILLIAC-IV) [8,9] и обработки изображений [10, 11], что предопределилось эффективностью адаптации исходного объекта—изображения—к конфигурации клеток—процессоров, информационных элементов и т. д. [11].

Однако определенным недостатком, как нам представляется, является значительная трудоемкость прямого описания алгоритмов в виде систем параллельных подстановок (СПП) [8] , в связи с чем и была поставлена решаемая здесь задача приведения схем программ к СПП.

В работе исследуются вопросы, связанные с условиями возникновения противоречивости в СПП и показано, что стационарные клеточные преобразования над двоичным алфавитом никогда не противоречивы (2); в 3 описывается алгоритм приведения выбранного класса схем программ к СПП и приводится пример построения по схеме, соответствующей программе быстрого усеченного ортогонального преобразования в базисе функций Виленкина—Уолша [13], моделирующую ее СПП; в 4 рассмотрены некоторые условия максимального распараллеливания схем.

2. Системы параллельных подстановок и их свойства

В данном параграфе вводятся необходимые понятия систем клеточных преобразований на основе [8] и продемонстрированы некоторые свойства, в основном, стационарных СПП.

Пусть А—конечный алфавит, М—множество имен с мощностью не более чем счетной. Пара $(a, m) \in A x M$ называется клеткой.

Определение I. Пусть $f=(f_0, f_1, \dots, f_n)$ и $\varphi=(\varphi_0, \varphi_1, \dots, \varphi_n)$—упорядоченные некоторым образом совокупности функций f_i: $A^{n_i} \to A$ и φ_i: $M \to M$, причем для любого $x \in M$ в совокупности φ

$$\varphi_0(x) \neq \varphi_1(x) \neq \dots \neq \varphi_n(x) \tag{1}$$

Конечное множество пар $S=\{(f_0, \varphi_0)(f_1, \varphi_1) \dots (f_n, \varphi_n)\}$ называется конфигурацией.

Конфигурация, у которой все f_i и φ_i, $i=\overline{1, n}$), —константы, называется словом (элементом конфигурации) и обозначается $W \equiv \bar{S} \equiv \bar{S}^x \equiv S(x)$. Векторы f и φ называются соответственно первой и второй проекциями конфигурации (слова) и обозначаются $Pr_1(S)=f$, $Pr_2(S)=\varphi$. К(А, М)—множество всех слов над A и M.

Определение 2. Произведением [8] двух конфигураций $S_1=\{(a_0, \varphi_0)(a_1, \varphi_1) \dots (a_n, \varphi_n)\}$ и $S_2=\{(b_0, \psi_0)(b_1, \psi_1) \dots (b_m, \psi_m)\}$ называется конфигурация $S_3=\{(a_0, \varphi_0) \dots (a_n, \varphi_n)(b_0, \psi_0) \dots (b_m, \psi_m)\}$, а суммой S_1 и S_2 —S_3, полученное по правилам объединения множеств (обозначение операций: $S_3=S_1 * S_2$ и $S_3=S_1 U S_2$). Операция называется недопустимой, если результат ее—не конфигурация.

Выражение вида P: $S_1 * S_2 \to S_3$, где S_1, S_2, S_3— конфигурации, называется подстановкой (правилом), причем $S_1 * S_2$— левая часть, S_2 —контекст, S_3—правая часть. Подстановка Р называется применимой к слову w (выполнимой), если существует хотя бы один $x \in M$ такой, что $\overline{S_1 * S_2^x} \leqslant w$. Применение подстановки P описывается формулой

$$W'=P(W)=\Big(W \setminus \bigcup_{l=1}^{m} \bar{S}_1^{x_l}\Big) U \Big(\bigcup_{l=1}^{m} \bar{S}_3^{x_l}\Big), \tag{2}$$

где $x_1, x_2, \dots, x_m$—набор значений, на отором имеет место выполнимость P.

Конечное множество правил, записанных в произвольном порядке, называется системой параллельных подстановок

$$\Phi: \begin{cases} P_1 : S_{11} * S_{12} \to S_{13} \\ P_2 : S_{21} * S_{22} \to S_{23} \\ \quad \dots \\ P_n : S_{n1} * S_{n2} \to S_{n3} \end{cases}, \tag{3}$$

задающей преобразование слова w итерационной процедурой (w^{l-1} —результат $l-1$ шага итерации):

1) если ни одна подстановка P_j, $j=\overline{1,n}$), не применима к W^{l-1}, то $\Phi(W)=W^{l-1}$ является конечным результатом применения Φ к W;

2) если существует k правил $P_j \in \Phi$, применимых к W^{l-1}, то при очевидном условии, что $\bigcup_{j=1}^{k} \bigcup_{l=1}^{m_j} S_{i1} * S_{i2}$ (x_l)—конфигурация

$$W^l=\Big(W^{l-1} \setminus \bigcup_{j=1}^{k} \bigcup_{l=1}^{m_j} S_{j1}(x_l)\Big) U \Big(\bigcup_{j=1}^{k} \bigcup_{l=1}^{m_j} S_{j3}(x_l)\Big) \tag{4}$$

Правило Р (алгоритм Ф) называется непротиворечивым относительно слова W, если результат его применения на этом слове является словом, и называется непротиворечивым, если результат его применения на любом слове непротиворечив.

Правило Р называется стационарной подстановкой, если $Pr_2(S_1)=Pr_2(S_3)$, и СПП называется стационарной, если каждая ее подстановка стационарна

На основе результатов [14] справедлива

Теорема 1. Пусть правило Р выполнимо и W — входное слово. Р приводит к противоречивости, если и только если при этом существует $\overline{W}\subseteq W$ и $\overline{W}\cap\overline{S}_1=\varnothing$, что $Pr_1(\overline{W})\neq Pr_1(\overline{S}_3')$ и $Pr_2(\overline{W})=Pr_2(\overline{S}_3')$, где $\overline{S}_3'\subseteq\overline{S}_3$.

Следствие. Пусть для правила Р операция $S_2\ U\ S_3$ недопустима. Тогда существует хотя бы одно слово W, к которому применимо Р и на котором оно противоречиво.

Доказательство. Существование слова W очевидно следует из того, что таковым может быть элемент конфигурации $\overline{S_1*S_2}^*$, полученный фиксированием некоторого $x\in M$, для которого $\overline{S}^x=\overline{S_2US_3^x}$ не есть слово. По [14] существуют $W_1\subset\overline{S}$ и $W_2\subset\overline{S}$ такие, что $W_1\cap(\overline{S}_2^x\cap\overline{S}_3^x)=\varnothing$, $W_2\cap(\overline{S}_2^x\cap\overline{S}_3^x)=\varnothing$ и $Pr_1(W_1)\neq Pr_1(W_2)$, $Pr_2(W_1)=Pr_2(W_2)$, причем $W_1\subset\overline{S}_2$ и $W_2\subset\overline{S}_3$. Но так как $\overline{S_1*S_2^x}\subseteq W$, то $W_1\subset W$. Обозначив $W_1\equiv\overline{W}$ и $W_2\equiv\overline{S}_3'$, приходим к выполнению условия достаточности теоремы I.

Остановимся на некоторых свойствах правил и СПП.

Свойство 1. Правило Р называется подстановкой с тождественным преобразованием, если в S_1 существует $\overline{S}_1'\subseteq\overline{S}_1$, а в $S_3-\overline{S}_3'\subseteq\overline{S}_3$ такие, что $Pr_1(\overline{S}_1')=Pr_1(\overline{S}_3')$ и $Pr_2(\overline{S}_1')=Pr_2(\overline{S}_3')$, т. е. $\overline{S}_1'=\overline{S}_3'$. Такое правило приводимо к виду $P': S_1''*S_2''\to S_3''$, где $S_1''=S_1\setminus S_1'$, $S_2''=S_2\cup S_1'$, $S_3''=S_3\setminus S_3'$.

Свойство 2. Стационарная подстановка P_i перекрывает стационарную подстановку P_j, если $Pr_2(S_{i1})=Pr_2(S_{j1})$, $S_{j1}\subseteq S_{i1}$, а $S_{i2}\subseteq S_{j2}$ и $S_{j3}\subseteq S_{i3}$. Ясно, что P_j можно удалить из СПП.

Свойство 3. I) Любую подстановку $P: S_1*S_2\to S_3$ можно разбить на подстановки P_i, функционирующие на той же области определения и вкупе исполняющие функции P, по следующим правилам:

а) произведем разбиение множества S_1 на подмножества

$[S_1]=\{S_{i1}\}$, $i=(\overline{1,k})$, и $S_1=\bigcup\limits_{i=1}^{k}S_{i1}$, причем для любых i, j $S_{i1}\cap S_{j1}=\varnothing$;

б) записываем подстановку Р в виде системы

$$P:\begin{cases}P_1: S_{11}*S_{12}\to S_3\\ P_2: S_{21}*S_{22}\to S_3,\\ \quad\cdots\\ P_k: S_{k1}*S_{k2}\to S_3\end{cases}\quad S_{i2}=S_2*S_{11}*\ldots*S_{i-11}*S_{i+11}*\ldots*S_{k1}\qquad(5)$$

2) При выборе Р стационарной следует, что ее можно разбить по тому же принципу, что и в 1), но при одновременном разбиении конфигурации S_3 на подмножества—конфигурации S_{i3}, причем $Pr_2(S_{i1})=Pr_2(S_{i3})$. В этом случае система P принимает вид

$$P:\begin{cases}P_1: S_{11}*S_{12}\to S_{13}\\ P_2: S_{21}*S_{22}\to S_{23}\\ \quad\cdots\\ P_k: S_{k1}*S_{k2}\to S_{k3}\end{cases}\qquad(6)$$

Определение 3. Правила P_i и P_j находятся в отношении после-
довательности, если существуют x, $y \in M$ такие, что выполнено $\overline{S_{i1}*}$
$\overline{S^x_{i2}*S^y_{j3}}$ — не слово, а $\overline{S_{i1}*S_{i2}*}\cup\overline{S^y_{j3}}$ — слово.

Теорема 2. Пусть P и P′—стационарные, без тождественных преобразований и находятся в отношении последовательности. Тогда P и P′ не будут выполняться за один шаг итерации на тех $x \in M$, при которых они находятся в указанном отношении.

Доказательство. Предположим, что P и P′ выполнимы на текущем шаге итерации и по [14] $\overline{S_1*S^x_2}\cup\overline{S'_1*S'_2}^y$ — слово. Одновременно $\overline{S'_1*S'_2}^y*\overline{S^x_3}\notin K(A,M)$, а $\overline{S'_1*S'_2}^y\cup\overline{S^x_3}\in K(A,M)$, т. е. существуют $W'\subset\overline{S'_1}$ $*\overline{S'_2}^y$ и $W\subset\overline{S}^x_3$ такие, что $Pr_1(W')=Pr_1(W)$ и $Pr_2(W')=Pr_2(W)$. Но $Pr_2(W)=Pr_2(W'')$, где $W''\subset\overline{S}^x_1$, и по условию нетождественности подстановок P, P' $Pr_1(W)\ne Pr_1(W'')$; следовательно $Pr_1(W')\ne Pr_1(W)$, т. е. $\overline{S'_1*S'_2}^y\cup\overline{S_1*S^x_2}$ не есть слово, что противоречит условию выполнимости P и P' за один шаг итерации.

Следующей леммой определяются условия, при которых имеет место противоречивость стационарных подстановок и алгоритмов.

Лемма. Противоречивость результата алгоритма стационарных подстановок Ф может возникнуть только лишь при параллельной реализации правил из Ф в течение исполнения одного шага итерации (по формуле (4): либо $m_j>1$, либо $k>1$, либо то и другое).

Доказательство. Предположим обратное, т. е. пусть слово W^{k+1} получается из слова W^k применением одного правила P_j из Φ фиксированием единственного x. По теореме I это означает, что существует такое $W\subset W^k$, $W\cap S_{j1}=\varnothing$, что $Pr_1(W)\ne Pr_1(\overline{S'_{j3}})$ и $Pr_2(W)=$ $=Pr_2(\overline{S'_{j3}})$, где $\overline{S'_{j3}}\subseteq\overline{S_{j3}}$. Однако, т. к. $Pr_2(S_{j1})=Pr_2(S_{j3})$ и $W\cap\overline{S}_{j1}=\varnothing$ то существует $\overline{S'_{j1}}\subseteq\overline{S_{j1}}$ такое, что $Pr_2(\overline{S'_{j1}})=Pr_2(W)$ и $Pr_1(\overline{S'_{j1}})\ne Pr_1(W)$, т. е. приходим к противоречию с условием выполнимости подстановки P_j на слове W^k.

Теорема 3. Пусть $A=A$ и $|A|=2$ (например, $\overline{A}=\{0,1\}$), тогда алгоритм Φ стационарных подстановок, правила которого не обладают свойством тождественного преобразования, никогда не приводит к противоречивости.

Доказательство. По предыдущей лемме при предположении о существовании СПП из указанного в условии теоремы класса, приводящей к противоречивости, на некотором шаге итерации должны быть выполнены подстановки P_i и P_j такие, что $\overline{S_{i1}*S^x_{i2}}\cup\overline{S_{j1}*S_{j2}}^y$ — слово, а $\overline{S}=\overline{S}^x_{i3}\cup\overline{S}^y_{j3}$ — нет, что, в свою очередь, по теореме I означает наличие $W_1\subseteq\overline{S}^x_{i3}$ и $W_2\subseteq\overline{S}^y_{j3}$ при условии $Pr_1(W_1)\ne Pr_1(W_2)$ и $Pr_2(W_1)=$ $Pr_2(W_2)$. Так как $Pr_2(S_{i1})=Pr_2(S_{i3})$ и $Pr_2(S_{j1})=Pr_2(S_{j3})$, то в S_{i1} существует $\overline{S'}^x_{i1}\subseteq\overline{S}^x_{i1}$, а в $S_{j1}-\overline{S'}^y_{j1}\subseteq\overline{S_{j1}}^y$, что $Pr_2(S'^x_{i1})=Pr_2(W_1)=Pr_2(W_2)$ $=Pr_2(\overline{S'_{j1}}^y)$ и $Pr_1(\overline{S'_{i1}}^x)\ne Pr_1(W_1)$, $Pr_1(\overline{S'_{j1}}^y)\ne Pr_1(W_2)$. Следовательно, по условию нетождественности и двоичности алфавита, $Pr_1(\overline{S'_{i1}}^x)$ $\ne Pr_1(\overline{S'_{i1}}^x)\ne Pr_1(\overline{S'_{j1}}^y)$, т. е. $\overline{S_{i1}*S_{i2}}^x\cup\overline{S_{j1}*S_{j2}}^y$ не есть слово, что противоречит предположению.

3. СП-алгоритм

Ниже описывается алгоритм приведения определенного класса схем программ к классу конструкций, функционирующих на основе клеточных преобразований и являющихся продуктом распараллеливания этих схем на внутриблочной основе [16]. Выбор одного из простейших принципов распараллеливания обусловлен стремлением максимально упростить изложение СП-алгоритма. Следует отметить, что доопределением предлагаемой здесь методики возможно достижение больших степеней параллелизма (см., в частности, п. 4 данной статьи).

Опишем рассматриваемый класс схем программ, выбрав за основу определения стандартных схем и схем с массивами [12,15], введением следующего базиса B:

а) $X^I=\{I, I_1, I_2,\ldots, J, J_1, J_2,\ldots\}$—множество интерпретированных переменных, определенных на множестве целых неотрицательных чисел и имеющих начальное значение 0;

б) $R=\{a, a_1, a_2\ldots, b, b_1, b_2,\ldots\}$—множество символов—констант;

в) $\overline{X}=\{x, x_1\ x_2,\ldots, y, y_1, y_2,\ldots\}$—множество простых переменных;

г) $X^{(I)}=\{x(I), x_1(I),\ldots, \lambda(I_1), x(I_2),\ldots\}$—множество выражений, называемых переменными массивов, где $I, I_1, I_2\ldots\in X^I$;

д) $F=\{f_1, f_2,\ldots, g_1, g_2,\ldots, h_1, h_2\ldots\}$—множество функциональных символов;

е) $C=\{c_1, c_2,\ldots\}$—множество предикатных символов;

ж) {старт, стоп, (,). ,. :=}$=V$—множество специальных символов.

Помимо того в базис добавлены три интерпретированных оператора $I:=I+1$, $I:=I\dot{-}1$, $I:=0$, где

$$I\dot{-}1=\begin{cases}I-1, & I>0\\ I=0, & I\leqslant 0\end{cases} \qquad (7)$$

С каждым функциональным и предикатным символом связана местность—целое неотрицательное число (мы будем указывать его только в случаях необходимости).

Оператор присваивания над памятью $X=\overline{X}UX^{(I)}$ —это выражение вида $X:=f(y_1\ldots, z)$, где X—выходная переменная, а $(y,\ldots, z)$—вектор входных переменных длины n, n—местность f (считается, что $x, y,\ldots,z$—переменные обоих типов, определенных в (в и г)), $f\in F$. Условный оператор над памятью $X=\overline{X}UX^{(I)}$—это предикатный терм вида $c(x, y,\ldots,z)$, где $c\in C$, $(x, y,\ldots,z)$—вектор длины n входных переменных (из $\overline{X}\cup x^{(I)}$), n—местность. С помощью специальных символов старт и стоп формируются операторы старт и стоп.

Различают две формы представления схем программ: графовую и линейную [12]. Что касается линейной формы, то конструирование основывается на привнесении в базис специальных символов, позволяющих формировать алголоподобное представление схем (такая форма используется в статье только в примере, а потому ее формальное описание, с которым можно ознакомиться в [12], опускается).

Графовая форма—это помеченный ориентированный граф, в котором имеется

а) единственная начальная вершина, из которой исходит одна дуга (непомеченная) и которой поставлен в соответствие оператор старт;

б) одна заключительная вершина, из которой не исходит ни одна дуга и которой поставлен в соответствие оператор стоп;

в) прочие вершины, каждая из которых имеет либо одну выходную дугу, и тогда ей поставлен в соответствие функциональный оператор (вершина—преобразователь), либо две, помеченные символами 0 и I, дуги, и такой вершине сопоставлен условный оператор (вершина—распознаватель).

Интерпретацией схемы на базисе B и области интерпретации D является функция I, проводящая следующие сопоставления:

а) каждой константе $a \in R$ сопоставляется элемент $I(a)=d \in D$;

б) каждому $x \in \overline{X}$—$I(x)=d \in D$;

в) каждой переменной массива $x(I) \in X^{(I)}$—$(d_1, d_2, \ldots, d_l, \ldots)$, где все $d_i \in D$;

г) каждому $f^{(n)} \in F$, $n \geq 1$,—всюду определенную функцию $F^{(n)}=$ $=I(f^{(n)}) : D^n \to D$;

д) каждому $c^{(n)} \in C$, $n \geq 1$,—всюду определенный предикат $F^n_c=$ $=I(c^{(n)}) : D^{(n)} \to \{0,1\}$;

Исполнение интерпретированной схемы (программы) осуществляется последовательным продвижением от вершины старт в соответствии с результатами преобразований и распознаваний в вершинах к вершине стоп.

Определение 4. Цепочка функциональных операторов, заключенная с обеих сторон операторами, не являющимися функциональными, называется блоком.

Очевидно, что каждая схема обладает конечным числом блоков.

Определение 5. Простой схемой с массивами (ПСМ) называется приведенная относительно конкуренционной зависимости [16] схема в базисе B, причем для различных $f_i^{(n)}$, $f_j^{(n)}$ и $c_k^{(n)}$, $c_e^{(n)}$ могут соответствовать одна $F^{(n)}$ и одна F^n_c.

Далее определяется алфавит А и множество М, на которых задается класс формальных систем параллельных подстановок (ФСПП).

Множество А клеточных состояний ФСПП определяется объединением множеств $A=\overline{A} \cup T \cup \overline{F}$, где $\overline{\mathbf{A}}=\{f_1, f_2, \ldots, x^0_1, x^0_2, \ldots, d^{01}, d^{02}, \ldots,$ $d^{0p}, \ldots\} \cup R \cup \{0,1\}$ (d^{0l}—символ начального состояния массива), $T=\{t_1,$ $t_2, \ldots\}$—множество символов переменных, определенных на $\overline{A}$.

Определение 6. Пусть $\bar{\tau}(x)=\xi_k\ \xi_{k-1} \ldots \xi_0$—последовательность состояний клетки с именем $x \in M$, полученная за k преобразований (возможно, тождественных) над ней применением ФСПП, причем $\bar{\xi}_0=$ $=x^0$—начальное состояние клетки из $\overline{A}$. 0—историей клетки x называется такой префикс $\bar{\tau}^0(x)=\bar{\xi}_k\ \bar{\xi}_{k-1} \ldots \bar{\xi}_l$, $0 \leqslant l \leqslant k$, что $\bar{\xi}_l \in R \cup \{x^0_1, x^0_2, \ldots,$ $d^{0p}, \ldots\}$, а при $j < l$ $\bar{\xi}_j \in \{f_1, f_2, \ldots, g_1, g_2, \ldots, h_1, h_2, \ldots\} \subset \overline{A}$.

Пусть T_0—множество всевозможных 0—историй. Тогда множество $\overline{F}$ образуют функции $f^{n_l}_{c_l} : T^{n_l}_0 \to \{0,1\}$.

Если W^t—слово, полученное на t—ом шаге итерации при исполнении некоторой ФСПП, то текущее множество $\overline{T}_0$ 0—историй образуют 0—истории клеток, входящих в W^t. Преобразование состояния клетки с именем X и 0—историей $\tau^0(x)$ влечет появление новой 0-истории, полученной добавлением нового состояния к $\tau^0(x)$ слева и выделением соответствующего префикса. Если клетка удаляется из текущего для ФСПП слова, то из $\overline{T}_0$ удаляется соответствующая ей 0—история.

$M=N\cup\overline{X}\cup R\cup P_f\cup\overline{C}$, где N—натуральный ряд, $P_f=\{P_i\}$, $\overline{C}=\{c_i\}$ —конечные множества.

Текущее слово W согласовано с произвольной интерпретацией базиса β, если, при наличии в нем клетки с именем $c_i, I(c_i^{(n)})$ равно значению функции $f_{c_i}^n$.

Теорема 4. Для любой ПСМ можно построить моделирующую ее ФСПП.

Доказательство. Первым этапом определим взаимно однозначное соответствие I_S, на котором основаны все дальнейшие преобразования:

а) каждой переменной $x\in\overline{X}$ и константе $a\in R$ ставится в соответствие единственное $m\in M$;

б) каждой переменной массива $x(I)$ ставится в соответствие функция $\varphi_{x(I)}: M\rightarrow M$;

в) каждой счетчиковой переменной ставятся в соответствие функции $\psi_I: M\rightarrow M$ и $\overline{\psi}_I: M\rightarrow M$, для которых выполняется одно из двух: либо $\psi_I(m)=\overline{\psi}_I(m+1)$, либо $\overline{\psi}_I(m)=\psi_I(m+1)$, $m\in N$;

г) каждому функциональному символу $f_i\in F$ ставится в соответствие $f_i\in\overline{A}$;

д) каждому предикатному символу $c_i\in C$ ставится в соответствие $f_{c_i}\in\overline{F}$.

Вторым этапом всем конструкциям вида $x:=f_i(x_1, x_2, \ldots, x_k)$ из ПСМ S сопоставляются подстановки вида

$$P_{f_j}: \{(t_x^j, x)(1, P_j)\}*S_{j2}\rightarrow\{(f_j, x)(0, P_{f_j})\}, \tag{8}$$

где $f_j\in\overline{A}$, x, $P_j\in M$, $S_{j2}=\{(t_{x_1}^j, x_1)(t_{x_2}^j, x_2)\ldots(t_{x_k}^j, x_k)\}$

либо $S_{j2}=\{(t_{x_1}^j, x_1)(t_{x_2}^j, x_2)\ldots(t_{x_{l-1}}^j, x_{l-1})(t_{x_{l+1}}^j, x_l)\ldots(t_{x_k}^j, x_k)\}$ (в случае, если существует $x_i=x$, $1\leqslant i\leqslant k$, из вектора входных переменных). Клетки с именами P_{f_j} исполняют роль явного управления подстановками (включение и выключение). Необходимость их следует из случаев построения подстановок с тождественным преобразованием, имеющих место для вершин схемы, лежащих внутри циклов.

Третьим этапом всем конструкциям вида $C_i(x_1, x_2, \ldots, x_k)$ по всем входным для каждого распознавателя блокам сопоставляется набор подстановок:

$$P_{c_i}^l: \{(0, P_{f_1})(0, P_{f_2})\ldots(0, P_{f_n})\}*\{(t_{x_1}^l, x_1)\ldots(t_{x_k}^l, x_k)\}\rightarrow\{(f_{c_i}, c_i)\}, \tag{9}$$

где $l=(\overline{1,\lambda})$, $\bar{\lambda}$ — количество входных для распознавателя блоков. Оче-
видно, что при таком построении подстановок клетки управления с
именами P_{f_1}, $P_{f_2}, \ldots, P_{f_n}$ в дальнейшем не нужны и, возникнув после
исполнения одной из подстановок (за исключением начального мо-
мента реализации ФСПП) включения блоков (см. (10) и (11)), они
уничтожаются. Пусть $h_1, h_2, \ldots, h_r$ и $g_1, g_2, \ldots, g_m$ — вершины — преобра-
зователи выходных блоков, лежащих на ветвях I и 0 распознавателя.

Подстановки включения блоков имеют вид:

$$P^{\text{и}}_{c_l} : \{(1,c_l)\} \to \{(1,P_{h_1})(1,P_{h_2})\ldots(1,P_{h_r})\} \tag{10}$$

$$P^{\text{л}}_{c_l} : \{(0,c_l)\} \to \{(1,P_{g_1})(1,P_{g_2})\ldots(1,P_{g_m})\} \tag{11}$$

Четвертым этапом по оператору старт формируется входное слово
W^0 для конструируемой ФСПП. $W^0=\{(x^0_1, x_1)(x^0_2, x_2)\ldots(d^0_1, \varphi x_1(I_1)^{(1)}\ldots$
$(d^0_\rho, \varphi_{x_\rho}(I_\rho)\ (1))(1,P_0)\ldots(1,\overline{P}_l)\ldots(1,P_{f_l})\ldots\}$, где $d^0_1\ d^0_2, \ldots, d^0_\rho$ — начальные
состояния переменных $x_1(I_1)$, $x_2\ (I_2), \ldots, x_\rho(I_\rho)$, представленных в W^0
стартовыми клетками $\varphi_{x_1(I_1)}(1), \ldots \varphi_{x_\rho(I_\rho)}(1)$. Клетки управления подста-
новками определяют включение подстановок, соответствующих функ-
циональным операторам блока, лежащего между вершиной старт и
ближайшим к ней распознавателем.

Пятым этапом сопоставляется оператору стоп конечный набор под-
становок

$$P^l_F : \{(0,P^l_{f_1})(0,P^l_{f_2})\ldots(0,P^l_{f_k})(1,P_0)(1,P_1)\ldots(1,P_p)\} \to \varnothing, \tag{12}$$

где правила $P^l_{f_1}, P^l_{f_2}, \ldots, P^l_{f_k}$ соответствуют функциональным операторам
каждого блока.

Каждому оператору вида $I := I+1$ сопоставляется подстановка

$$P_I : \{(I^0, \psi_I(m))(1,P_I)\} \to \{(I^0, \bar{\psi}_I(m))(0,P_\rho)\}, \tag{13}$$

где $\bar{\psi}_I(m+1)=\bar{\psi}_I(m)$, $m \in N$, $m>0$ при $I<0$ и $m=1$ при $I=0$. Каждому
оператору вида $I := I-1$ ставится в соответствие подстановка типа
(13), но $\bar{\psi}_I(m)=\bar{\psi}_I(m+1)$. Ко всем подстановкам вида (13), применяется
принцип включения и выключения так, как это было описано выше.

В контекст подстановки, в которой участвует клетка с именем, соот-
ветствующим переменной массива, включается клетка с именем, соот-
ветствующим индексу переменной этого массива. Такое расширение
контекста позволяет фиксировать функции пробега по массивам на
конкретных значениях. Этот подход определен выбором метода после-
довательного прохода циклов при распараллеливании.

Пример 1. Пусть в схеме имеются операторы 1: $x\ (I) := f_l(y(J),z)$;
2: $I := I+1$; 3: $J := J-1$; причем 2 и 3 обеспечивают счетчиковые
значения для I. Тогда

$$P_{f_l} : \{(t^{f_l}_{x(I)}, \varphi_{x(I)}(m))(1,P_{f_l})\} * \{(t^{f_l}_{y(J)}, \bar{\bar{\varphi}}_{y(J)}(m))(t^{f_l}_z, z)(I^0, \bar{\psi}_I(m))(J^0, \bar{\psi}_J(m))$$
$$(0,P_I)(0,P_J)\} \to \{(f_l, \varphi_{x(I)}\ (m))(0,P_{f_l})\}$$

$$P_I : \{(I^0,\psi_I(m))(1,P_I)\} \to \{(I^0,\bar{\psi}_I(m))(0,P_I)\}$$

$$P_J : \{(J^0,\psi_J(m))(1,P_J)\} \to \{(J^0,\bar{\psi}_J(m))(0,P_J)\}.$$

Если операторы, между которыми имеет место информационная зависимость, находятся в блоке, входящем в тело цикла, то между подстановками, соответствующими им, устанавливается отношение последовательности с помощью клетки управления подстановкой, встраиваемой в контекст того правила, которое ожидает исполнения другого. Из определения операторов изменения счетчиков следует их неприводимость и, следовательно, вышеуказанное проводится для соответствующих им подстановок.

Следующим этапом, в связи с тем, что в W^0 имеются лишь первые представители массивов, вводятся подстановки

$$P_0 : \{(I^0, \varphi_l(m))\}^*\{(1, P_0)\} \to \{(I^0, \bar{\varphi}_l(m))\}$$

$$P_i : {}^*\{(I^0, \bar{\varphi}_l(m))(1, P_i)\} \to \{(d_i^0, \varphi_{x_i(l_i)}(m))\} \qquad (14)$$

при условии, что в W^0 вносится клетка с именем $\bar{\varphi}_l$ (2) ($i=\overline{(1,p)}$, p — количество различных массивов). Таким образом, на каждом шаге итерации конструируемой ФСПП P_0 будет задавать очередное значение $m \in M$, а все P_i будут формировать клеточные подпулы в текущих W, соответствующие массивам x_i (l_i), причем эта процедура будет опережать использование подпулов подстановками преобразования массивов.

Необходимо отметить, что области значений $\varphi_{x_i(l_i)}$, $i=(\overline{1,\mu})$, не должны пересекаться.

Последним этапом проводится определение значений используемых переменных состояний клеток выявлением информационных зависимостей соответствующих подстановкам операторов.

Как было показано, описанный выше СП-алгоритм отображает любую ПСМ в представление, изначально несущее в себе свойство параллельного исполнения, что и приводит схему к распараллеливанию.

После приведенных построений очевидным образом следуют два утверждения.

Утверждение 1. Если алгоритм ФСПП останавливается, то результирующее слово не содержит клеток управления подстановками.

Утверждение 2. ПСМ пуста тогда и только тогда, когда пуста моделирующая ее ФСПП.

В истинности теоремы можно убедиться, проанализировав исполнение произвольного блока ПСМ и функционирование соответствующего фрагмента ФСПП.

В качестве примера демонстрации работы СП-алгоритма приводится ПСМ, соответствующая программе быстрого усеченного ортогонального преобразования в базисе функций Виленкина—Уолша (см. текст программы в [13]. Помимо того в примере показаны различные вариации в построениях подстановок относительно пустых блоков и засылок констант (такие вариации не показывались в описании алгоритма как не существенные).

Пример 2. Пусть задана схема

0: старт
1: $x_1 := a_N$
2: $l := l+1$
3: $y_1 := f_1(y_1)$
4: $x_2 := f_2(x_1)$

5: $K:=K+1$
6: $y_2:=f_3(y_2)$
7: $x_3:=f_4(x(k))$
8: $J:=J+1$
9: $y_3:=f_5(y_3)$
10: $x_4:=f_6(x(J))$
11: $x(J):=f_7(x_3,x_4)$
12: $x(k):=f_8(x_3,x_4)$
13: если $c_1(y_2)$ то 5.
14: $x_1:=f_9(x_2)$
15: если $c_2(y_1)$ то 2
16: $x_5:=a_1$
17: $x_6:=a_3$
18: $x_7:=a_1$
19: $L:=L+1$
20: $y_4:=f_{10}(y_4)$
21: $x_5:=f_{11}(x_5)$
22: $x_7:=f_{12}(x_7,x_8)$
23: $x_g:=f_{13}(x_7,x_5)$
24: $M=M+1$
25: $y_5:=f_{14}(y_5)$
26: $x(M):=f_{15}(x(M),x_7)$
27: если $c_3(y_5)$ то 24
28: $x_6:=f_{16}(x_g)$
29: если $c_4(y_4)$ то 19
30: если $c_5(x_{10})$ то 36
31: $x_{11}:=f_{17}(g_N)$
32: $T:=T+1$
33: $y_6:=f_{18}(y_6)$
34: $x(T):=f_{19}(x(T),\ x_{11})$
35: если $c_6(y_6)$ то 32
36: стоп

ФСПП.

$P_1:\{(t^1_{x_1},\ x_1)(1,P_1)\}\to\{(a_N,x_1)(0,P_1)\}$

$P_2:\{(I^0,\psi_I(m))(1,P_2)\}\to\{(I^0,\bar{\psi}_I(m))(0,P_2)\}$

$P_3:\{(t^3_{y_1},\ y_1)(1,P_3)\}\to\{(f_1,y_1)(0,P_3)\}$

$P_4:\{(t^4_{x_2},x_2)(1,P_4)\}^*\{(t^4_{x_1},x_1)\}\to\{(f_2,x_2)(0,P_4)\}$

$P_5:\{K^0,\bar{\psi}_k(m))(1,P_5)\}^*\{(0,P_2)\}\to\{(K^0,\bar{\psi}_k(m))(0,P_5)\}$

$P_6:\{(t^6_{y_2},\ y_2)(1,P_6)\}\to\{(f_3,y_2)(0,P_6)\}$

$P_7:\{(t^7_{x_3}\ x_3)(1,P_7)\}^*\{(t^7_{x(k)}\ \bar{\varphi}_{x(k)}),(m))\ (K^0,\ \bar{\psi}_k(m))\ (0,P_5)\}\to\{(f_4,\ x_3)$
$(0,P_7)\}$

$P_8:\{(J^0,\varphi_J(m))(1,P_8)\}^*\{(1,P_5)\}\to\{(J^0,\bar{\psi}_J(m))(0,P_8)\}$

$P_9:\{(t^9_{y_3},\ y_3)(1,P_9)\}\to\{(f_5,y_3)(0,P_9)\}$

$P_{10}:\{(t^{10}_{x_4},\ x_4)(1,P_{10})\}^*\{(t^{10}_{x(J)},\ \varphi_{x(J)}(m))(J^0,\bar{\psi}_J(m))(0,P_8)\}\to\{(f_6,x_4)$
$(0,P_{10})\}$

$P_{11}:\{(t^{11}_{x(J)},\ \bar{\varphi}_{x(J)}(m))\}^*\{(t^{11}_{x_3},\ x_3)(t^{11}_{x_4},\ x_4)(J^0,\bar{\psi}_J(m))(0,P_8)\}\to\{(f_7,\bar{\varphi}_{x(J)}$
$(m))0_1P_{11})\}$

$P_{12}:\{(t^{12}_{x(k)},\ \varphi_{x(k)})(m))(1,P_{12})\}^*\{(t^{12}_{x_3},x_3)(t^{12}_{x_4},x_4)(K^0,\psi_k(m))(0,P_{11})\}\to\{(f_8,$
$\varphi_{x(k)}(m))(0,P_{12})\}$

$P^1_{c_1}:\{(0,P_1)(0,P_2)\ldots(0,P_{12})\}^*\{(t^{1,1}_{y_2},\ y_2)\}\to\{(f_{c_1},\ c_1)\}$

$P^N_{c_1}:\{(1,c_1)\}\to\{(1,P_5)(1,P_6)\ \ldots\ (1,P_{12})\}$

$P^\wedge_{c_1}:\{(0,c_1)\}\to\{(1,P_{13})\}$

$P_{13}:\{(t^{13}_{x_1},\ x_1)(1,P_{13})\}^*\{(t^{13}_{x_2},x_2)\}\to\{(f_9,x_1)(0,P_{13})\}$

$P^1_{c_2}:\{(0,P_{13})\}^*\{(t^{2,1}_{y_1},y_1)\}\to\{(f_{c_2},\ c_2)\}$

$P^N_{c_2}:\{(1,c_2)\}\to\{(1,P_2)(1,P_3)\ldots(1,P_{12})\}$

$P^\wedge_{c_2}:\{(0,c_2)\}\to\{(1,P_{14})(1,P_{15})\ldots(1,P_{24})\}$

$P_{14}:\{(t^{14}_{x_5},\ x_5)(1,P_{14})\}\to\{(a_1,x_5)(0,P_{14})\}$

$P_{15}:\{(t^{15}_{x_6},\ x_6)(1,P_{15})\}\to\{(a_3,x_6)(0,P_{15})\}$

$P_{16}:\{(t^{16}_{x_7},\ x_7)(1,P_{16})\}\to\{(a_1,\ x_7)(0,P_{16})\}$

$P_{17}:\{(L^0,\psi_L(m))(1,P_{17})\}\to\{(L^0,\bar{\psi}_L(m))(0,P_{17}(\}$

$P_{18}:\{(t^{18}_{y_4},\ y_4)(1,P_{18})\}\to\{(f_{10},y_4)(0,P_{18})\}$

$P_{19}:\{(t^{19}_{x_5},\ x_5)(1,P_{19})\}\to\{(f_{11},x_5)(0,P_{19})\}$

$P_{20}:\{(t^{20}_{x_7},\ x_7)(1,P_{20})\}^*\{(t^{20}_{x_8},\ x_8)\}\to\{(f_{12},x_7)(0,P_{20})\}$

$P_{21}\{(t^{21}_{x_9},\ x_9)(1,P_{21})\}^*\{(t^{21}_{x_7},\ x_7)(t^{21}_{x_5},\ x_5)\}\to\{(f_{13},x_9)(0,P_{21})\}$

$P_{22}:\{(M^0,\psi_M(m))(1,P_{22})\}^*\{(0,P_{17})\}\to\{(M^0,\bar{\psi}_M(m))(0,P_{22})\}$

$P_{23}:\{(t^{23}_{y_5},\ y_5)(1,P_{23})\}\to\{(f_{14},y_5)(0,P_{23})\}$

$P_{24}:\{(t^{24}_{x(M)},\ \varphi_{x(M)}(m))(1,P_{24})\}^*\{(t^{24}_{x_7},x_7)\ (0,P_{22})(M^0,\bar{\psi}_M(m))\}\to\{(f_{15},\varphi_{x(M)}$
$(m))(0,P_{24})\}$

$P_{25};\{(t^{25}_{x_6},\ x_6)(1,P_{25})\}^*\{(t^{25}_{x_9},x_9)\}\to\{(f_{16},x_6)(0,P_{25})\}$

$P^1_{c_3}:\{(0,P_{14})(0,P_{15})\ldots(0,P_{24})\}^*\{(t^{3,1}_{y5},\ y_5)\}\to\{(f_{c_3},\ c_3)\}$

$P^N_{c_3}:\{(\mathrm{p},c_3)\}\to\{(1,P_{22})(1,P_{23})(1,P_{24})\}$

$P^\wedge_{c_3}:\{(0,c_3)\}\to\{(1,P_{25})\}$

$P^1_{c_4}:\{(1,P_{25})\}^*\{(t^{25}_{y_4},y_4)\}\to\{(f_{c_4},c_4)\}$

$P^N_{c_4}:\{(1,c_4)\}\to\{(1,P_{17})(1,P_{18})\ldots(1,P_{24})\}$

$P^1_{c_5}:\{(0,c_4)\}^*\{(t^{5,1}_{x_{10}})\}\to\{(f_{c_5},\ c_5)\}$

$P^N_{c_5}:\{(1,c_5)(1,P_0)(1,P_{30})\}\to\varnothing$

$P^\wedge_{c_5}:\{(0,c_5)\}\to\{(1,P_{26})\ldots(1,P_{29})\}$

$P_{26}:\{(t^{26}_{x_{11}},\ x_{11})(1,P_{26})\}^*\{(a_{\mathrm{N}},\ m_{a_{\mathrm{N}}})\}\to\{(f_{17},x_{11})(0,P_{26})\}$

$P_{27}:\{(T^0,\psi_T(m))(1,P_{27})\}\to\{(T^0,\bar{\psi}_T(m))(0,P_{27})\}$

$P_{28}:\{(t^{28}_{y_6},y_6)(1,P_{28})\}\to\{(f_{18},y_6)(0,P_{28})\}$

$P_{29}:\{(t^{29}_{x(T)},\ \varphi_{x(T)}(m))(1,P_{29})\}_*\{(t^{29}_{x_{11}},x_{11})(T^0,\bar{\psi}_T(m))(0,P_{27})\}\to\{(f_{19},\bar{\varphi}_{x(T)}$
$(m))(0,P_{29})\}$

$P_{30}:{}^*\{(Q^0,\bar{\psi}_Q(m))(1,P_{30})\}\to\{(d^0,\bar{\varphi}_{x(Q)}(m))\}$

$P_0:\{Q^0\psi_Q(m))\}^*\{(1,P_0)\}\to\{(Q^0,\bar{\psi}_Q(m))\}$

$P^1_{c_6} : \{(0,P_{26})\ldots(0,P_{29})\}^*\{(t^{6,1}_{y},y_6)\}\to\{(f_c ,c_6)\}$

$P^{N}_{c_6} : (1,c_6)\}\to\{(1,P_{27})(1,P_{28})(1,P_{29})\}$

$P^{\Lambda}_{c_6} : \{(0,c_6)(1,P_0)(1,P_{30})\}\to\varnothing$

$P^2_{c_1} : \{(0,P_{22})(0,P_{23})(0,P_{24})\}^*\{(t^{3,2}_{y5}, y_5)\}\to\{(\bar{f}_{c_3}, c_3)\}$

$P^3_{c_1} : \{(0,P_2)\ldots(0,P_{12})\}^*\{(t^{1,3}_{y_2},y_2)\}\to\{(f_{c_1}\ c_1)\}$

$P^2_{c_3} : \{(0,P_{22})(0,P_{23})(0,P_{24})^*\{(t^{3,3}_{y_5},y_5)\}\to\{(f_{c_3}, c_3)\}$

$P^2_{c_6} : \{(0,_{27})0,P_{28})(0,P_{29})\}^*\{(t^{6,2}_{y}, y_6)\}\to\{(fc_6, c_6)\}$

$W^0=\{Q^0,\psi_Q(2))(d^0, \varphi_{x(Q)}(1))(I^0,\psi_I(1))(J^0,\psi_J(1))(K^0,\psi_k(1))(L^0,\psi_L(1))$
$(M^0,\bar{\psi}_M(1))(T^0,\psi_T(1))(x^0_1,x_1)(x^0_2,x_2)(x^0_3,x_3)(x^0_4)(x_4)(x^0_5,x_5)(x^0_6,x_6)$
$(x^0_7,x_7)(x^0_8,x_8)(x^0_9,x_9)(x^0_{10},x_{10})(x^0_{11},x_{11})(y^0_1,y_1)(y^0_2,y_2)(y^0_3,y_3)(y^0_4,y_4)$
$(y^0_5,y_5)(y^0_6,y_6)(a_N,m_{a_N})(1,P_1)(1,P_2)\ldots(1,P_{12})(1,P_0)(1,P_{30})\}$

$t^1_{x_1}=\{x^0_1\}$

$t^3_{y_1}=\{y^0_1,f_1\}$

$t^4_{x_2}=\{x^0_2,f_2\}$

$t^4_{x_1}=\{a_N\}$

$t^6_{y_2}=\{y^0_2,f_3\}$

$t^7_{x_3}=\{x^0_3,f_4\}$

$t^7_{x(k)}=\{d^0,f_7,f_8\}$

$t^9_{y_3}=\{y^0_3,f_5\}$

$t^{10}_{x_4}=\{x_4,f_6\}$

$t^{10}_{x(J)}=\{d^0,f_7,f_8\}$

$t^{11}_{x(J)}=t^{12}_{x(k)}=t^{10}_{x(3)}$

$t^{11}_{x_3}=t^{12}_{x_3}=\{f_4\}$

$t^{1,1}_{y_2}=\{f_3\}$

$t^{11}_{x_4}=\{f_6\}$

$t^{13}_{x_1}=\{a_N,f_9\}$

$t^{13}_{x_2}=\{f_2,f_9\}$

$t^{2,1}_{y_1}=\{f_1\}$

В примере подстановки $P^1_{c_6}$ и $P^N_{c_6}$ соответствуют случаю пустых блоков; правила P_1, P_{14} и др. соответствуют засылкам. Помимо того в подстановках включения блоков в некоторых случаях опускается перечисление всех клеток управления, а значения переменных состояний приведены только для части клеток с именами—символами имен переменных схемы программы. Подстановки P_0 и P_{30} включаются входным словом W^0 и последовательно до момента остановки алгоритма ФСПП (если, конечно, такой момент наступит) формируют элементы массива $x(Q)$.

4. Максимальный параллелизм

Рассмотрим вопросы, связанные с максимальным параллелизмом схем программ и СПП.

Определение 7. Алгоритм Ф из класса СПП эквивалентен по историям алгоритму Ф′ из того же класса, если только Ф останавливается тогда и только тогда, когда останавливается Ф′; Ф противоречив тогда и только тогда, когда противоречив Ф′; $Pr_2(\Phi(W))=Pr_2(\Phi'(W))$, где W—общее для обоих входное слово, и история каждой клетки из $W'=\Phi(W')$ совпадает с историей той же клетки для $W'=\Phi'(W)$.

Из определения следует, что Φ „более параллелен" чем, Φ', если при любом входном слове W выходное слово W' получается за меньшее число шагов итерации при исполнении Φ относительно Φ'.

Утверждение 3. Для классов схем программ, поддающихся максимальному распараллеливанию (см. [17]), наличие средств управления подстановками в СПП не является необходимым.

Истинность утверждения следует из свойств тотальных схем (в частности, из возможности приведения тотальной схемы к ациклическому виду, что обуславливает отсутствие тождественных подстановок).

Утверждение 4. Для классов схем программ, поддающихся максимальному распараллеливанию (см. [17]), существует моделирующее их ФСПП с максимальным параллелизмом.

Истинность данного утверждения следует из всех предыдущих рассуждений.

В заключение следует отметить, что исследуемые в данной работе вопросы и предлагаемые решения ориентированы прежде всего на создание аппарата, посредством которого было бы возможно эффективное описание и исполнение вычислительного процесса на мультипроцессорных и мультимашинных ассоциациях. Предлагаемый в работе алгоритм позволяет решать задачи автоматического распараллеливания программ, причем класс ФСПП может выступать фундаментом при создании функций упреждения [3].

ЛИТЕРАТУРА

1 *Karp R. M., Miller R. E.* Parallel program Schemata.—J. Comput. Syst. Sci, 1969, vol. 3, №2, p. 147—195 (Русский перевод: *Карц Р. М., Миллер Р. Е.* Параллельные схемы программ. В кн. Кибернетический сборник (новая серия), вып. 13. М., Мир, 1976, с. 5—61).

2 *Котов В. Е., Нариньяни А. С.* Асинхронные вычислительные процессы над памятью.—Кибернетика, 1966 № 3, с. 64—71.

3 *Keller R. M.* Parallel program Schemata and maximal parallelism. I. Fundamental results.—Journal ACM, 1973, vol. 20, №3, p. 514—537.

4 *Фон Нейман Дж.* Теория самовоспроизводящихся автоматов.—М., Мир, 1971, 384 с.

5 *Мур Э. Ф.* Математические модели самовоспроизведения. В сб.: Математические проблемы в биологии. М., Мир, 1966, с. 36—62.

6 *Smith III A. R.* Cellular automata complexity trade-offs.—Inform. and Control, 1971, №18, p. 466—482

7 *Wachsmuth I.* L'ocally synchronous cellulair automata.—Acta Cybernetica, vol. 6, 1983, №1, p. 55—75.

8 *Бандман О. Л.* и др. Методы параллельного микропрограммирования.—Новосибирск, Наука, 1981, с. 180.

9 *Королев Л. Н.* Структуры ЭВМ и их математическое обеспечение.—М., Наука, 1978, 345 с.

10 *Престон К. и др.* Основы клеточной логики с приложениями к обработке изображений в медицине.—ТИИЭР, т. 67, № 5, 1979, с. 149—185.

11. *Resenfeld A., Wu A.* Cellular computens for parallel region-level Image processing.—Lect. Notes Comput. Sci., vol. 153, 1983. p. 333—348.

12 *Котов В. Е.* Введение в теорию схем программ.—Новосибирск, Наука, 1978, 258 с.

13 *Дагман Э. Е., Кухарев Г. А.* Быстрые дискретные ортогональные преобразования.—Новосибирск. Наука, 1983, 230 с.

14 *Сергеев С. Н.* Распознавание непротиворечивости алгоритмов стационарных подстановок.—В кн. Архитектура вычислительных систем с программируемой структурой.—Новосибирск, Наука, 1980, с. 18—25.

15 *Constabel R. L., Ciries D.* On classes of Program Schemata.— TR 71—105, Cornell University, 1971. (Русский перевод: *Констейбл Р. Л, Грис Д.* О классах схем программ. В кн.: Кибернетический сборник (новая серия), вып. 14., М., Мир, 1977, с. 122—178)

16 Алгоритмы, математическое обеспечение и архитектура многопроцессорных вычислительных систем.—М., Наука, 1982, 332 с.

17 *Вальковский В. А.* Распараллеливание операторных схем над массивами.—Программирование, 1977, № 1, с. 15—20.

2—306