МАТЕМАТИЧЕСКОЕ И АРХИТЕКТУРНОЕ ОБЕСПЕЧЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ
(Вычислительные системы)
1985 год
Выпуск 109

УДК 681.3.323

РАСПОЗНАВАНИЕ НЕЗАВЕРШАЕМОСТИ СИНХРОННЫХ ИНТЕРПРЕТАЦИЙ ПАРАЛЛЕЛЬНЫХ МИКРОПРОГРАММ

С.М.Ачасова

В синхронных вычислениях по параллельной микропрограмме может возникнуть ситуация, в которой некоторое подмножество слов, характеризующих состояния обрабатываемых данных, повторяется циклически и образует контур в графе вычислений. Такие вычисления называются незавершающимися. Явление незавершаемости в асинхронных вычислениях по параллельным микропрограммам изучалось в [I], где был дан метод распознавания незавершаемости. В данной статье этот метод модифицирован для распознавания незавершаемости в синхронном случае. Показано, что для подкласса устойчивых параллельных микропрограмм задача распознавания незавершаемости синхронных интерпретаций сводится к задаче распознавания незавершаемости в асинхронном случае.

I. Используемые понятия

В этом разделе будет приведен краткий перечень необходимых для изложения материала понятий, связанных с параллельными микропрограммами и их интерпретациями [I-4].

Параллельная микропрограмма обрабатывает массив данных, который представляется в виде множества пар $\langle a, m \rangle$, где a - данное из алфавита A, m - место данного в массиве. Место представляется вектором координат узла целочисленной в общем случае n-мерной решетки, $m = x_1, \ldots, x_n$, $\{m\} = M$. Пара $\langle a, m \rangle$ называется клеткой, имеющей состояние a и имя m. Конечное множество $\{\langle a_i, m_i \rangle\} \subset A \times M$, в котором нет двух клеток с одинаковыми именами, называется словом и обозначается W.

Микрокоманда параллельной микропрограммы выполняет операцию подстановки $S_1 * S_2 \to S_3$. В подстановке справа и слева находятся слова. Слово в левой части разделено звездочкой на два подслова, S_1 и S_3 равномощны и содержат клетки с одним и тем же набором имен. Выполнение подстановки в слове W происходит следующим образом. Если $S_1 * S_2 \subseteq W$, т.е. подстановка применима к слову W, то из W удаляется S_1 и к оставшейся части добавляется S_3, т.е. образуется новое слово $W' = (W \setminus S_1) \cup S_3$.

Чтобы можно было по единой формуле производить однотипные подстановки, операция подстановки обобщается до микрокоманды и записывается θ: $S_1(m) * S_2(m) \to S_3(m)$, где

$$S_1(m) = \{\langle a_0, \varphi_0(m)\rangle, \langle a_1, \varphi_1(m)\rangle, \ldots, \langle a_p, \varphi_p(m)\rangle\},$$

$$S_2(m) = \{\langle b_0, \psi_0(m)\rangle, \langle b_1, \psi_1(m)\rangle, \ldots, \langle b_q, \psi_q(m)\rangle\},$$

$$S_3(m) = \{\langle c_0, \varphi_0(m)\rangle, \langle c_1, \varphi_1(m)\rangle, \ldots, \langle c_p, \varphi_p(m)\rangle\}.$$

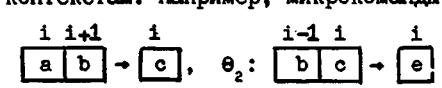
Все функции φ_i и ψ_j попарно различны и являются функциями сдвига на константу.

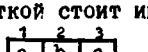
Выполнение микрокоманды θ для конкретного значения $m = m_t$ называется микрооперацией, которая записывается

$$\theta(m_t): S_1(m_t) * S_2(m_t) \to S_3(m_t).$$

Левая часть $S_1(m) * S_2(m)$ микрокоманды состоит из базы $S_1(m)$ и контекста $S_2(m)$. Клетки, составляющие базу, изменяют свои состояния в результате выполнения микрооперации. Контекст служит для определения применимости микрокоманды, состояния клеток контекста не изменяются при выполнении микрооперации.

Две микрокоманды пересекаются, если их левые части могут иметь общие клетки. Если общие клетки принадлежат контексту одной микрокоманды и контексту другой, то говорим, что микрокоманды пересекаются по контекстам. Например, микрокоманды



(внутри клетки записано состояние, над клеткой стоит имя) пересекаются по контекстам. Действительно, в слове 

клетка с именем 2 принадлежит левым частям микрокоманд θ_1, θ_2 и для той и другой микрокоманд является контекстной.

Множество микрокоманд $\{\theta_j\}$, $j = \overline{1,u}$, является параллельной микропрограммой и обозначается Ψ. Параллельная микропрограмма мо-

жет интерпретироваться в синхронном и асинхронном режимах. При этом шаг i синхронной интерпретации состоит в преобразовании слова W^{i-1} в слово W^{i} путем выполнения всех применимых к W^{i-1} микрокоманд (всех микроопераций для каждой микрокоманды). На каждом шаге асинхронной интерпретации выполняется только одна микрооперация из множества применимых к W^{i-1}; если это множество содержит более одного элемента, то результат шага i асинхронной интерпретации оказывается недетерминированным.

Множество слов, которые получаются из исходного слова при синхронной (асинхронной) интерпретации параллельной микропрограммы, вместе с отношением следования слов называется синхронным (асинхронным) вычислением. Граф отношения следования слов называется графом вычислений.

2. Незавершаемость синхронных интерпретаций

С и н х р о н н а я и н т е р п р е т а ц и я параллельной микропрограммы называется н е з а в е р ш а ю щ е й с я, если найдется слово $W \subset A \times M$ (A,M - конечные множества), для которого синхронное вычисление по этой микропрограмме окажется незавершающимся (в графе такого вычисления имеется контур).

ПРИМЕР I. Дана микропрограмма

$$W_1 = \left\{ \begin{array}{l} \theta_1: \begin{array}{|c|c|}\hline a & b \\ \hline \end{array} \rightarrow \begin{array}{|c|c|}\hline e & d \\ \hline \end{array}, \\ \theta_2: \begin{array}{|c|c|}\hline b & c \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline d \\ \hline \end{array}, \\ \theta_3: \begin{array}{|c|c|c|}\hline e & d & e \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline c \\ \hline \end{array}, \\ \theta_4: \begin{array}{|c|c|c|}\hline d & c & d \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline e \\ \hline \end{array}. \end{array} \right.$$

Граф вычислений по этой микропрограмме для исходного слова abcd изображен на рис.I. В графе имеется контур, соответствующий вычислению $(edcd \rightarrow eded)^*$ (звездочка - знак бесконечного повторения выражения в скобках).

* * *

Рис.I

Задача распознавания незавершаемости синхронной интерпретации параллельной микропрограммы Ψ для любого исходного слова размера не более N (размер N слова определяется величиной n-мерной решетки, в которой размещен массив обрабатываемых данных) ставится следующим образом. Выяснить, существует ли хотя бы одно слово размера не более N_i для которого микропрограмма Ψ порождает незавершающееся синхронное вычисление. Для отыскания такого слова будем использовать метод стимуляции микрокоманд [I], несколько изменив его для синхронного случая по сравнению с асинхронным.

Назовем з а ц е п л е н и е м некоторого подмножества микрокоманд $\{\theta_1,\ldots,\theta_k\} \subset \Psi$ слово, в котором микрокоманды пересекаются таким образом, что граф отношения пересечения микрокоманд является связным. Одна микрокоманда (точнее, ее левая часть) может иметь зацепления с различными подмножествами микрокоманд. Будем рассматривать все возможные зацепления для каждой микрокоманды $\theta_i \in \Psi$, т.е. зацепления θ_i с одиночными микрокомандами, с парами микрокоманд, с тройками и т.д. Множество всех возможных зацеплений для микрокоманды θ_i будем обозначать $H(\theta_i)$. Величина зацепления ограничивается размером N. Можно говорить о зацеплении некоторого слова W с микрокомандами микропрограммы Ψ.

ПРИМЕР 2. Пусть имеется микропрограмма

$$\Psi_2 = \left\{ \begin{array}{ll} \theta_1: & \begin{array}{|c|c|}\hline i & i+1 \\ \hline a & b \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline i \\ \hline c \\ \hline \end{array}, \\ \theta_2 & \begin{array}{|c|c|}\hline i-1 & i \\ \hline b & d \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline i \\ \hline g \\ \hline \end{array}, \\ \theta_3: & \begin{array}{|c|c|}\hline i-1 & i \\ \hline b & e \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline i \\ \hline h \\ \hline \end{array}, \\ \theta_4: & \begin{array}{|c|c|}\hline i & i+1 \\ \hline d & c \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline i \\ \hline g \\ \hline \end{array}. \end{array} \right.$$

На рис.2 множество зацеплений $H(\theta_1)$ изображено в виде графа. Слово ab считается тривиальным зацеплением для самого себя. На рис.3 изображено множество зацеплений слова dab с микрокомандами $\theta_i \in \Psi_2$.

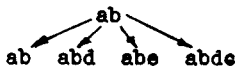


Рис.2

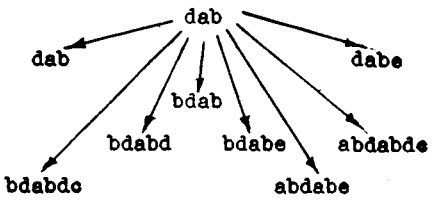


Рис. 3

Опишем процесс стимуляции микрокоманд для решения задачи распознавания незавершаемости в синхронном случае. Как и в [I],в качестве исходного слова берем левую часть $S_{i1} * S_{i2}$ микрокоманды θ_i. Для $S_{i1} * S_{i2}$ заводим вершину графа стимуляции $\Gamma(\theta_i)$.Через $\Gamma(\theta_i)$ будем обозначать граф стимуляций в синхронном случае в отличие от $G(\theta_i)$ - в асинхронном случае. Поскольку в синхронном случае все применимые микрокоманды выполняются на одном шаге вычисления, то нужно, кроме $S_{i1} * S_{i2}$, в качестве исходных слов взять еще все зацепления $H(\theta_i)$. Иначе говоря, нужно вместе с θ_i стимулировать все микрокоманды, которые могут оказаться применимыми в одном слове с θ_i.Заведем для каждого зацепления из $H(\theta_i)$ (и в том числе для тривиального зацепления $S_{i1} * S_{i2}$) вершину графа $\Gamma(\theta_i)$. Граф, построенный на этом первом шаге стимуляций для микрокоманды $\theta_i \in \Psi_2$, точно такой же, как и граф, изображенный на рис.2. Следующий шаг процесса стимуляций - один шаг синхронного вычисления для каждого зацепления, затем построение всех зацеплений для каждого слова - результата вычисления и т.д. Для каждого вновь полученного слова в графе заводится вершина. В процессе построения в графе может появиться контур. Это происходит в том случае, если для какого-либо слова W_j (результат шага вычисления) среди полученных ранее нашлось слово W_r такое, что оно,будучи дополнено клетками, послужившими для построения зацеплений, лежащих на пути от W_r к W_j (если такой путь имеется), оказалось равным W_j.

Процесс стимуляций микрокоманд заканчивается, если дальнейшее построение зацеплений невозможно вообще или невозможно без увеличения размера зацепления за пределы N, бесмысленно, если зацепления строятся путем добавления одних и тех же клеток, т.е.

возникает повторение ситуации, и наконец, в том случае, если в графе стимуляций появился контур.

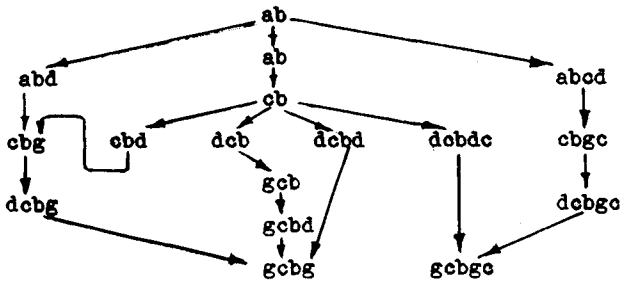


Рис. 4

Рассмотрим микропрограмму Ψ_2', которая получается из Ψ_2 (см. пример 2) путем удаления θ_3, таким образом, $\Psi_2'=\{\theta_1,\theta_2,\theta_4\}$, где $\theta_1,\theta_2,\theta_4 \in \Psi_2$. Граф стимуляций $\Gamma(\theta_1)$, изображенный на рис.4, дает полную картину поиска такого слова, для которого микропрограмма Ψ_2' порождает незавершающееся синхронное вычисление. Видно, что дальнейшие стимуляции невозможны, т.е. процесс закончился. Таким же образом строятся графы $\Gamma(\theta_i)$ для i=2,4. Микропрограмма Ψ_2' такова, что ни в одном графе $\Gamma(\theta_i)$ для всех $\theta_i \in \Psi_2'$ нет контуров. А вот микропрограмма Ψ_1 (см. пример I) такова, что в каждом графе $\Gamma(\theta_i)$, $i = \overline{1,4}$, $\theta_i \in \Psi_1$, есть контур. Граф $\Gamma(\theta_1)$, $\theta_1 \in \Psi_1$, изображен на рис.5.

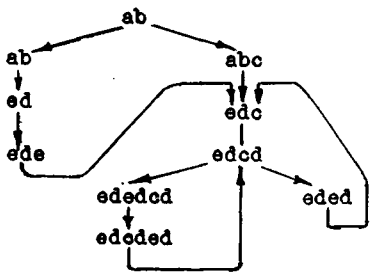


Рис. 5

ТЕОРЕМА I. Синхронная интерпретация параллельной микропрограммы Ψ является завершающейся для любого слова размера не более N, если и только если ни в одном графе $\Gamma(\theta_i)$, $\theta_i \in \Psi$, $i = \overline{1, |\Psi|}$, нет ни одного контура.

ДОКАЗАТЕЛЬСТВО. Необходимость очевидна. Достаточность следует из самого процесса стимуляций микрокоманд, при котором строятся все возможные зацепления для всех слов-результатов синхронного вычисления, начиная с левой части каждой микрокоманды. Это значит, что нет слов, кроме представленных в графах $\Gamma(\theta_i)$, $i = \overline{1, |\Psi|}$, для которых микропрограмма Ψ могла бы породить незавершающиеся синхронные вычисления.

3. Распознавание незавершаемости синхронных интерпретаций устойчивых микропрограмм

Покажем, что для распознавания незавершаемости синхронных интерпретаций устойчивых микропрограмм вместо громоздкого прямого метода, описанного в предыдущем разделе, можно использовать косвенный метод, который состоит в том, что задача распознавания незавершаемости в синхронном случае сводится к той же задача в асинхронном случае. Сведение синхронной задачи к асинхронной дает возможность уменьшить трудоемкость распознавания незавершаемости. Под трудоемкостью будем понимать количество шагов, выполняемых в процессе стимуляции микрокоманд. При этом будем считать равноценными шаги синхронного и асинхронного вычислений и им же равноценным каждый шаг расширения слова с целью стимуляций микрокоманд.

Напомним, что процесс стимуляции микрокоманд в асинхронном случае организован таким образом, что на одном шаге его стимулируется только одна микрокоманда и поэтому слово - источник для стимуляций - имеет столько расширений (а каждое расширение - это вершина в графе стимуляций $G(\theta_i)$), сколько микрокоманд могут пересечься с этим словом. В синхронном случае слово-источник расширяется для стимуляций всех возможных подмножеств микрокоманд, а таких расширений, т.е. зацеплений, существенно больше, чем в асинхронном случае. Поэтому процесс стимуляций в асинхронном случае существенно менее громоздок, чем в синхронном, т.е. граф $G(\theta_i)$ имеет существенно меньше вершин, чем граф $\Gamma(\theta_i)$.

Устойчивой называется параллельная микропрограмма, в которой микрокоманды пересекаются по контекстам. В такой микропрограмме выполнение одной из двух применимых и пересекающихся микрокоманд не нарушает условие применимости другой. А это значит, что в каком бы порядке ни выполнялись применимые микрокоманды по одной, любыми подмножествами или все сразу, результат вычисления будет один и тот же. Следовательно, синхронное вычисление по устойчивой микро-

программе для данного исходного слова эквивалентно асинхронному вычислению для того же исходного слова.

ТЕОРЕМА 2. Синхронная интерпретация устойчивой параллельной микропрограммы является завершающейся, если является завершающейся асинхронная интерпретация этой микропрограммы.

ДОКАЗАТЕЛЬСТВО. В графе стимуляций $G(\theta_i)$ (асинхронный случай) есть все те же преобразования слов, что и в графе $\Gamma(\theta_i)$ (синхронный случай), поскольку все микрокоманды устойчивой микропрограммы работают, не мешая друг другу, и после выполнения некоторой микрокоманды возможна стимуляция той же микрокоманды, которая пересекалась с первой до выполнения ее. Это дает возможность в результате получить все возможные зацепления микрокоманд, проводя процесс стимуляции в асинхронном режиме.

Приведем примеры процессов стимуляции для асинхронного и синхронного случаев для одной и той же устойчивой микропрограммы. Примеры убеждают в том, что процесс стимуляции в асинхронном случае гораздо менее громоздок, чем в синхронном.

ПРИМЕР 3. Дана устойчивая микропрограмма

$\Psi_3 =$

	i-1	i	i+1		i
θ_1:	a	b	c	→	a
θ_2:	c	d	e	→	a
θ_3:	e	a	d	→	d
θ_4:	c	a	e	→	h

Непосредственной проверкой можно убедиться в том, что микрокоманды в Ψ_3 пересекаются только по контекстам. На рис.6 изображен граф стимуляций $G(\theta_1)$, на рис. 7 - граф $\Gamma(\theta_1)$. Граф $\Gamma(\theta_1)$ содержит в два раза больше вершин, чем граф $G(\theta_1)$.

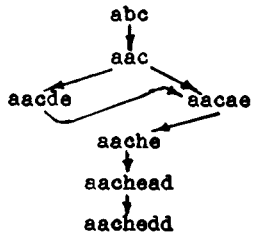


Рис. 6

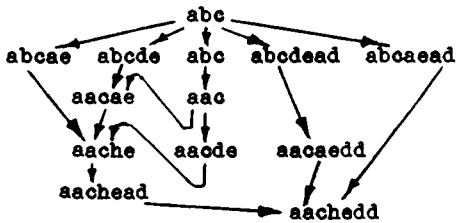


Рис. 7

Литература

I. АЧАСОВА С.М. Распознавание незавершаемости асинхронной интерпретации параллельной микропрограммы. - В кн.: Архитектура и математическое обеспечение вычислительных систем (Вычислительные системы, вып.104). Новосибирск, 1984,с.47-60.

2. БАНДМАН О.Л. Асинхронная интерпретация параллельных микропрограмм.- Кибернетика, 1984,№2,с.14-20.

3. Методы параллельного микропрограммирования /Под ред.Бандман О.Л. - Новосибирск: Наука, 1981. - 181 с.

4. АЧАСОВА С.М. Анализ асинхронной интерпретации параллельных микропрограмм. - В кн.: Однородные вычислительные системы из микро-ЭВМ (Вычислительные системы,вып.97).Новосибирск,1983,с.28-52.

Поступила в ред.-изд.отд.
28 июня 1985 года