ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Сборник трудов
1965 г. Института математики СО АН СССР Выпуск I7

МАТРИЧНЫЙ ρ -ЯЗЫК ДЛЯ ОПИСАНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Э.В. Евреинов, Ю.Г. Косарев

Как указывалось в работах [I],[2], параллельные алгоритмы существенно отличаются от известных алгоритмов [3] одновременностью выполнения большого числа операций. Формально отличие состоит в том, что вместо однокоординатной записи (например, в виде строчки) надо применять двухкоординатную, где одной координатой служит последовательность операций во времени в каждой из ветвей вычислений, а другой - распределение операций в каждый момент времени между ветвями вычислений. Заметим, что имеющиеся языки описания алгоритмов [4] -[6] не пригодны для ρ -алгоритмов и применение указанных языков оказывается по той же причине неудобным и при анализе некоторых свойств современных ЭВМ. Так, например, в работе [7] при оценке производительности ЭВМ авторы пришли к необходимости ввести обобщенную операцию, отражающую одновременность выполнения некоторого числа операций различными блоками ЭВМ.

Для описания схемы ρ -алгоритмов введем матричный язык. Для краткости будем его называть ρ - языком.

В ρ -языке используются в качестве элементов как простые (см.[2]), так и обобщенные операторы. Под последними будем понимать последовательность нескольких простых операторов,если: I) один и только один из входящих в него простых операторов имеет внешний вход, 2) в каждый момент времени выполняется только один простой оператор, 3) за конечное число шагов после выполнения оператора, имеющего внешний вход, будут выполнены все простые операторы.

ρ -операторы назовем о б о б щ е н н ы м и ρ -операторами, если их компонентами являются обобщенные операторы. Обобщенные ρ -операторы будем обозначать либо буквой Q_j, либо $\dot{Q}_j$, либо в виде столбца

$$\begin{pmatrix} Q_{1j} \\ Q_{2j} \\ \vdots \\ Q_{nj} \end{pmatrix}$$

где Q_{ij} $(i=1,2,\ldots,n)$ - обобщенный оператор, принадлежащий i -ой ветви вычислений.

Простые ρ -операторы будем обозначать буквами латинского и русского алфавитов, написанными жирным шрифтом, либо с точкой сверху, либо в виде столбца из входящих в него простых операторов.

Для некоторых часто встречающихся стандартных операторов будем употреблять установившиеся обозначения:

A - арифметический оператор, производящий вычисления по формулам;

P - оператор условного перехода;

$F(i)$ - оператор изменения параметра i, изменяющий адреса либо содержимое определенных ячеек памяти, зависящих от значения параметра;

$И$ - оператор начала, определяющий первую команду программы;

$Я$ - оператор конца, определяющий останов машины;

$Ф$ - оператор формирования, осуществляющий изменение других операторов;

$В$ - оператор восстановления первоначального значения параметра либо содержимого ячейки памяти;

E - всякий простой оператор, отличный от перечисленных выше;

Λ - пустой оператор, пропускающий выполнение одного шага вычислений.

Аналогично, некоторые ρ -операторы, все составляющие которых одинаковы и совпадают с одним из стандартных операторов, будем называть стандартными ρ -операторами и обозначать теми же буквами, что и для его составляющих, написанных жирным шрифтом или снабженных точкой сверху. К стандартным ρ - операторам мы будем относить также операторы, специфичные для вычислительной системы [8]:

$\dot{H}-\rho$ - оператор настройки, производящий изменение состояний коммутаторов элементарных машин либо параметров, управляющих структурой элементарных машин (ЭМ).

$\dot{O}-\rho$ - оператор обмена информацией между ЭМ, который указывает для каждой ЭМ, какой код (содержащийся в ячейке оперативной памяти, арифметическом устройстве или устройстве управления) должен быть послан во внешний канал связи либо принят из канала связи и направлен в один из блоков ЭМ либо то и другое одновременно.

$\dot{P}-\rho$ - оператор обобщенного условного перехода, устанавливающий изменение хода вычислительного процесса в системе в зависимости от выполнения некоторого условия (или условий) в одной или всех ЭМ.

Введем также по аналогии с [5] левую и правую ρ-полу - скобки: $\underset{k}{\llcorner}$ и $\underset{k}{\lrcorner}$, соответственно. Левая полускобка $\underset{k}{\llcorner}$ будет всегда следовать за $\dot{P}\rho$-оператором обобщенного условного перехода. Правая полускобка $\underset{k}{\lrcorner}$ предшествует ρ -оператору, к выполнению которого необходимо перейти в случае невыполнения условия, определяемого $\dot{P}$. В случае выполнения условия осуществляется переход к ρ-оператору, следующему за $\dot{P}$.

Логической схемой ρ -алгоритма будем называть матрицу

$$\begin{bmatrix} \Omega_{11} & \Omega_{12} \dots & \Omega_{1j} & \dots & \Omega_{1k} \\ \Omega_{21} & \Omega_{22} \dots & \Omega_{2j} & \dots & \Omega_{2k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \Omega_{i1} & \Omega_{i2} \dots & \Omega_{ij} & \dots & \Omega_{ik} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \Omega_{L1} & \Omega_{L2} \dots & \Omega_{Lj} & \dots & \Omega_{Lk} \end{bmatrix}$$

элементами j -го столбца ($j=1,2,\dots,k$) которой являются либо операторы простые или обобщенные, либо операторы условного перехода с левой полускобкой, либо операторы с правой полускобкой; элементами i -ой строки ($i=1,2,\dots,L$) являются операторы,образующие кортеж,соответствующий i -ой ветви вычислений,причем в i -ой строке для каждой левой(правой)полускобки имеется одна и только одна правая (левая) полускобка с тем же индексом k . Все столбцы нумеруются слева направо,а строки сверху вниз.

Будем также использовать векторную форму записи схемы ρ-алгоритма, представляющую собой строку,составленную из символов ρ -операторов, левой и правой полускобок.

В некоторых случаях, по аналогии с [4], будем вместо левой полускобки использовать стрелку, исходящую из данного оператора условного перехода, с указанием номера оператора,к выполнению которого нужно перейти в случае невыполнения условия, а вместо правой полускобки использовать входящую стрелку с указанием номера оператора условного перехода, от которого осуществляется переход к данному.

Иногда для наглядности вместо полускобок или стрелок с адресами будем использовать для описания условных переходов в схемах ρ -алгоритмов линии, соединяющие оператор условного перехода с оператором , к выполнению которого следует перейти при невыполнении условия.

Для выяснения структуры связей между ветвями вычислений будем также применять запись логических схем ρ -алгоритмов в виде графов. Вершинами графов служат сами операторы, а также источники информации, дугами являются информационные и управляющие связи.

Приведем пример записи схемы ρ -алгоритма на ρ -языке, для чего рассмотрим задачу умножения двух квадратных матриц $\{a_{ij}\}$ и $\{b_{ij}\}$ n -го порядка. Элемент c_{ij} результирующей матрицы имеет вид:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} .$$

Пусть имеется вычислительная система из n^2 элементарных машин, тогда, вычисляя каждый элемент c_{ij} в своей машине, будем предполагать для простоты, что в ее памяти хранится строка a_i и столбец b_j .

Схема ρ -алгоритма на матричном ρ -языке запишется в виде:

$$
\begin{array}{c}
1 \\ 2 \\ \vdots \\ \\ n^2
\end{array}
\begin{bmatrix}
\dashv_1 & A_1 & A_2 & P_3 & \llcorner_1 & \text{Я}_4 \\
\dashv_1 & A_1 & A_2 & P_3 & \llcorner_1 & \text{Я}_4 \\
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & & & & & \\
\dashv_1 & A_1 & A_2 & P_3 & \llcorner_1 & \text{Я}_4
\end{bmatrix}
$$

где оператор A_1 вычисляет $d_{ijk}=a_{ik}\cdot b_{kj}$;

оператор A_2 вычисляет $d_{ijk-1}+d_{ijk}=l_{ijk}, d_{ij0}=0$;

P_3 - оператор условного перехода (по $k=n$);

Я_4 - оператор останова.

В векторной записи эта схема примет вид:

$$\dashv_1 \dot{A}_1 \dot{A}_2 \dot{P}_3 \llcorner_1 \dot{\text{Я}}_4$$

либо

$$\dot{A}_1 \dot{A}_2 \dot{P}_3 \dot{\text{Я}}_4 \ .$$

Пример граф-схемной записи для этой же задачи приведен в работе [2].

Как можно видеть из [9], P-язык в матричной и граф-схемной формах записи позволяет сравнительно просто описывать схемы параллельных алгоритмов.

Поступила в редакцию
I8.IX.1964 г.

ЛИТЕРАТУРА

I. Евреинов Э.В., Косарев Ю.Г. О вычислительных системах высокой производительности. Изв. АН СССР, сер. техническая кибернетика, I963, № 4, 3-25.

2. Косарев Ю.Г. О методике решения задач на универсальных вычислительных системах. Данный сборник, стр.6I-99.

3. Марков А.А. Теория алгоритмов. Труды Матем. ин-та АН СССР им. Стеклова, XLII, I954 г.

4. Ляпунов А.А. О логических схемах программ. Сб. "Проблемы кибернетики", Физматгиз, I958, вып. I, 46-74.

5. Янов Ю.И. О логических схемах алгоритмов. Там же, стр. 75-I27.

6. Калужнин Л.А. Об алгоритмизации задач. Там же, вып. 2, I959, 5I-67.

7. Базилевский Ю.Я., Шрейдер Ю.А. Методы оценки производительности универсальных цифровых машин с программным управлением. В сб.: "Вопросы теории математических машин". М., Физматгиз, 1958, Вып. I, 127-134.
8. Евреинов Э.В. Универсальные вычислительные системы с частично-переменной структурой. Данный сборник, стр.3-60.
9. Евреинов Э.В., Косарев Ю.Г. О решении задач на универсальных вычислительных системах. Данный сборник, стр.106-164.