

Кукса П.П. Шмаков Е.В. Руденко Ю.М.

Московский Государственный Технический Университет  
им. Н.Э. Баумана

E-mail: kouxa@online.ru  
WWW: <http://www.geocities.com/pkouxa>

## АЛГОРИТМ НАХОЖДЕНИЯ МАТРИЦЫ ЛОГИЧЕСКОЙ НЕСОВМЕСТИМОСТИ $L_t$

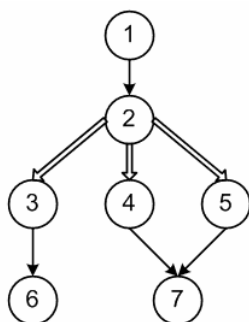
*Аннотация.* В статье рассматривается алгоритм нахождения матриц логической несовместимости операторов с транзитивными связями.

Параллельные архитектуры являются наиболее перспективным и динамичным направлением увеличения скорости решения прикладных задач. Методы достижения высокой производительности охватывают аппаратуру, логическую организацию и архитектуру системы. Построение эффективных систем требует комплексного применения новых достижений и передовых технологий на всех уровнях, начиная от быстродействующей элементной базы и заканчивая специальными алгоритмами и языками программирования. Структура алгоритма, как правило, должна отображаться в архитектуре ВС, что может в конечном счёте, существенно улучшить её эксплуатационные характеристики.

Использование ВС и МВК в системах управления в строгом временном режиме обработки информации выдвинуло ряд задач оптимального планирования совместного выполнения множества информационно и логически взаимосвязанных комплексов алгоритмов. При этом одной из главных задач является нахождение плана выполнения данного набора алгоритмов за минимальное время на данной ВС. А это требует нахождения множеств операторов, которые могут выполняться одновременно. При формировании плана параллельного выполнения необходимо выполнить анализ задающих и транзитивных связей логической несовместимости операторов алгоритма.

Построение граф-схемы алгоритма как его абстрактной модели позволяет ясно представить и проанализировать как общую структуру алгоритма, так и связи между отдельными операторами, а также выявить формальные преобразования, приводящие к решению задач оптимального планирования его выполнения [1]. Однако следует отметить, что предлагаемый в [1] алгоритм нахождения матрицы логической несовместимости с транзитивными связями (АНМЛН) не учитывает все случаи образования транзитивных связей логической несовместимости. Что приводит в конечном счёте к неполному учёту всех возможностей распараллеливания алгоритма.

При представлении алгоритма информационно-логической граф-схемой множество вершин графа соответствует множеству операторов алгоритма, а множество дуг - множеству информационных ( $\rightarrow$ ) и управляющих связей ( $\Rightarrow$ ) между операторами.



Для удобства вычислений, представим исходный граф (граф G) в виде матрицы следования S [1], отражающей ограничения на порядок выполнения операторов.

Рассмотрим пример (рис. 1) графа G. Из примера видно, что если оператор 3 логически несовместим с оператором 4 и 5, то операторы 6 и 7 также логически несовместимы, так как существуют связи 3- $\rightarrow$ 5 и 4- $\rightarrow$ 6. Эти операторы не могут выполняться при одной реализации алгоритма, то есть при формировании плана параллельного выполнения алгоритма они не должны рассматриваться совместно.

Рис. 1. Граф G

Таким образом, появляется необходимость на основе заданных связей логической несовместимости операторов определение такой несовместимости для всех операторов схемы, то есть необходимость введения транзитивных связей логической несовместимости операторов.

Возможна следующая ситуация, представленная в примере: операторы 4 и 5 несовместимы, но существуют связи  $4 \rightarrow 7$  и  $5 \rightarrow 7$ . Таким образом, оператор 7 нельзя считать логически несовместимым ни с одним из операторов 4 и 5. В то же время операторы 4 и 5 несовместимы с оператором 3, а так как существуют связи  $5 \rightarrow 7$  и  $4 \rightarrow 7$ , но отсутствует связь  $3 \rightarrow 7$ , то оператор 7 несовместим с оператором 3.

В общем случае при решении вопроса о наличии связей логической несовместимости данного оператора  $i$  необходимо найти множество  $\{\alpha_1, \dots, \alpha_r\}$  всех операторов, которые образуют связи вида  $\alpha_m \rightarrow i, m=1, \dots, r$

Для каждой вершины  $i$  находим связанные с ней вершины  $j_1 \dots j_m$  такие, что существуют связи  $j_m \rightarrow i$ . Для каждой вершины  $j_m$  находим множество  $IC_m = B_{j_m}$ , где  $B_{j_m}$  - множество операторов, логически несовместимых с  $j$  (с учётом транзитивных связей). Определяем пересечение этих множеств  $\bigcap_m IC_m$ . Если помимо связей по информации с вершиной  $i$  существуют связи по управлению вида  $k_L \Rightarrow i, L=1..t$ , то находим множество  $LC_L = (C_{k_L} \setminus (C_i \cup i)) \cup B_{k_L}$  где  $C_{k_L}$  - множество всех операторов, которые образуют связи вида  $k_L \rightarrow \alpha$ . Определяем пересечение этих множеств  $\bigcap_L LC_L$ .

Множество  $A_i$  операторов, логически несовместимых с  $i$ , находим как пересечение  $\left( \bigcap_m IC_m \right) \cap \left( \bigcap_L LC_L \right)$  (если одно из множеств пусто, то  $A_i$  принимается равным непустому множеству).

Использование приведенных правил позволяет построить следующий алгоритм нахождения связей логической несовместимости:

1. Организуем просмотр строк треугольной матрицы следования  $S$ , отбрасывая операторы-входы как не имеющие логически несовместимых операторов.
2. В очередной  $i$ -той строке матрицы  $S$  находим множество ненулевых элементов, т.е. множества номеров операторов, образующих информационные связи  $\{j_m\}, j_m \rightarrow i$  и управляющие связи  $\{k_L\}, k_L \Rightarrow i$ .
3. Для каждого оператора из  $\{j_m\}$  находим множество  $B_{j_m}$  и определяем их пересечение.
4. Для каждого оператора из  $\{k_L\}$  находим множества  $(C_{k_L} \setminus (C_i \cup i)) \cup B_{k_L}$  и их пересечение, где  $\setminus$  - разность множеств.
5. Формируем множество  $A_i$  операторов как пересечение найденных множеств.
6. В  $i$ -тую строку матрицы  $Lt$  заносим единицы в те позиции, номера которых содержатся в  $A_i$ .
7. Если просмотр не окончен, то переходим к пункту 2.
8. Транспонируем по столбцам полученную матрицу  $Lt$ .

Пример расчета графа G с помощью приведенного алгоритма представлен на рис. 2.

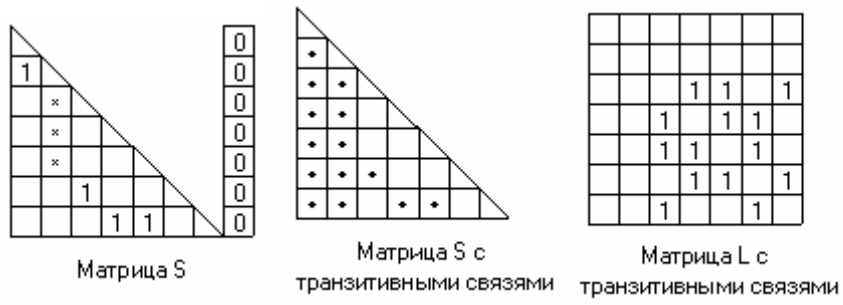


Рис. 2.

## Литература

1. А.Б. Барский Параллельные процессы в вычислительных системах. М.: «Радио и связь», 1990.