

## ПРАВИЛЬНАЯ НУМЕРАЦИЯ ДВУХПОЛЮСНОГО ОРИЕНТИРОВАННОГО ГРАФА

© 2012 Д. А. Попова-Коварцева

Самарский государственный аэрокосмический университет имени академика С. П. Королёва  
(национальный исследовательский университет)

Во многих задачах системного анализа активно используется теория графов, позволяющая с помощью математического аппарата анализа бинарных матриц получать важную информацию об внутреннем устройстве и свойствах моделируемых объектов. В частности алгоритмы правильной нумерации орграфов находят применение в задачах топологической сортировки бесконтурных графов. В статье рассматривается универсальный алгоритм топологической сортировки, позволяющий производить правильную нумерацию, включая контурные орграфы.

*Параллельная программа, ориентированный граф, правильная нумерация, вершина ветвления, вершина поглощения, ранг вершины.*

### Введение

Разработка параллельных программ является более сложным процессом по сравнению с созданием аналогичных последовательных кодов. Отчасти это связано с архитектурными особенностями современных многопроцессорных вычислительных систем параллельной обработки. Учитывая широкую распространенность вычислительных систем с распределенной памятью, для организации информационного взаимодействия между процессорами часто используется интерфейс передачи данных MPI. Несмотря на широкие возможности, которые предоставляет для разработчиков параллельных программ библиотека MPI, её применение сопровождается рядом трудностей связанных с необходимостью рациональной организации управления потоками данных между процессорами и синхронизацией вычислительных процессов. Неслучайно, что в последнее время разрабатываются высокоуровневые средства параллельного программирования, базирующиеся на MPI и облегчающие создание параллельных программ. К таким средствам относятся и система PGRAPH [1, 2].

В системе PGRAPH модель параллельного алгоритма представляется в графическом виде, близком к формализму представ-

ления блок-схем. Описание параллельных программ в виде граф-моделей позволяет реализовать образный, визуальный стиль представления алгоритмов программ и открывает широкие возможности их анализа с помощью методов теории графов.

### Основные положения

Система PGRAPH предоставляет пользователю визуальные средства формирования графических образов разрабатываемых алгоритмов, представляемых в виде графа управления, в котором в вершинах граф-модели реализуются функциональные преобразования данных, а на дугах проверяются условия возможности перехода к следующей вершине графа. Различаются три типа дуг графа: *последовательные*, управляемые логическими условиями (предикатами); *параллельные* (безусловные) дуги, «открывающие» параллельную ветку программы и *терминирующие* дуги, завершающие выполнение параллельной ветки программы. Несколько упрощая ситуацию, условимся обозначать последовательные вершины графа квадратами, начало параллельной ветки программы – пустым кружочком, терминирующие вершины – «залитым» кружочком. Примеры граф-моделей программ приведены на рис. 1.

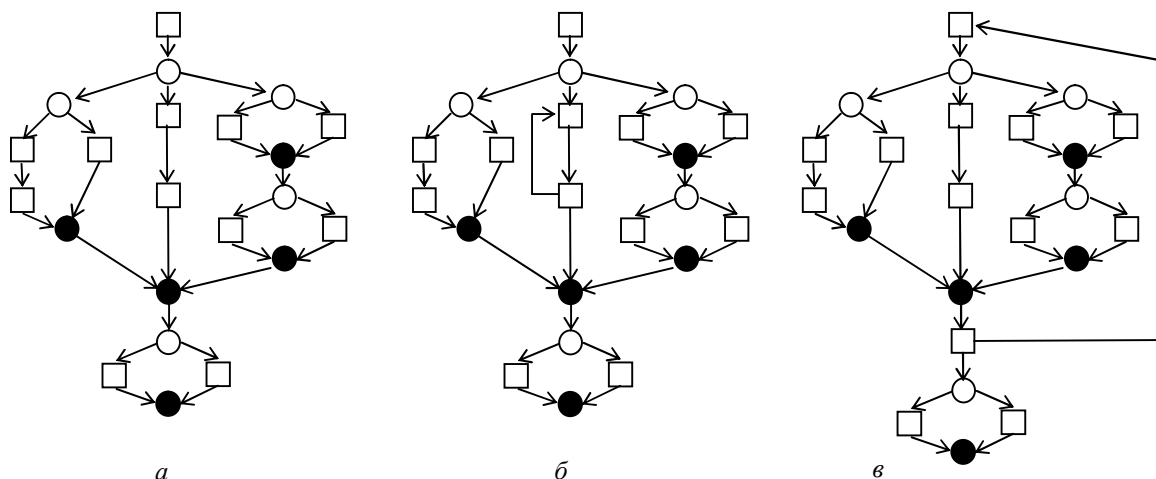


Рис. 1. Примеры граф-программ

По аналогии с последовательно-параллельными схемами соединения ненадежных элементов, используемыми в теории надежности, можно ввести в рассмотрение «классические» последовательно-параллельные модели вычислительных алгоритмов (см. рис. 1, а).

Понятие *классического Р-графа* определим следующим образом: граф, состоящий из одной вершины, является Р-графом; последовательное или параллельное соединение Р-графов (построенное с использованием вершин связок: параллельных и терминирующих вершин) является Р-графом; других Р-графов нет.

Из определения следует, что по построению классический Р-граф является двухполюсным направленным бесконтурным графом. Далее будем использовать алгебраическое определение ориентированного графа, заимствованное из работы [3], где под орграфом понимается пара  $\dot{G} = (V, a)$ . Здесь  $V$  – множество вершин орграфа,  $a \subseteq V \times V$  – отношение на множестве  $V$  (пара  $(u, v) \in a$  называется дугой графа).

В качестве полюсов модели параллельных вычислений, представленной классическим Р-графом, рассматриваются один источник (корень графа) и один сток (концевая вершина графа). Для определения факта принадлежности графа к классу направленных бесконтурных графов удобно использовать

теорему 3.33 работы [3]. Теорема утверждает, что в орграфе  $\dot{G}$  существует правильная нумерация вершин тогда и только тогда, когда  $\dot{G}$  – бесконтурный граф.

В данном случае под *правильной нумерацией* будем понимать такую нумерацию вершин графа  $v_1, v_2, \dots, v_n$ , когда из  $(v_i, v_j) \in a$  следует неравенство  $i \leq j$ . Таким образом, проверка свойств орграфа на принадлежность классу классических Р-графов сводится к построению алгоритма правильной нумерации графа.

Однако в практических приложениях при построении моделей параллельных алгоритмов в системе PGRAPH используются не только классические схемы Р-графа. В принципе орграф может содержать циклы (см. рис. 1, случаи б и в). В этом случае необходимо произвести предварительное расконтуривание графа, когда из всех контуров графа удаляют по одной дуге. Теорема 3.36 [3] позволяет производить подобные операции без нарушения связности графа. Теорема утверждает, что минимальное расконтуривание связного графа не нарушает его связности. Фактически расконтуривание превращает двухполюсный орграф в классический Р-граф.

Введем понятие *ранга* вершины двухполюсного графа. Под *рангом*  $r(v)$  вершины  $v$  графа  $\dot{G} = (V, a)$  будем понимать наибольшую из длин простых путей, начи-

нающихся из корневой вершины и заканчивающихся в вершине  $v$ .

*Вершиной поглощения* назовем вершину, в которую входят одна или несколько непомеченных дуг. *Вершиной ветвления* назовем вершину, в которую входят только помеченные дуги и которая имеет, по крайней мере, две исходящие дуги. Статус «поглощения» будем считать более приоритетным по сравнению со статусом «ветвления».

### Алгоритм правильной нумерации

Предлагается алгоритм нумерации вершин графа, состоящий из двух этапов. На первом этапе для всех вершин графа вычисляются их ранги. Для контурных графов производится его расконтуривание. На втором - реализуется правильная нумерация его вершин.

Введем четыре списка:  $S_{out}$  – вершин ветвления,  $S_{in}$  – вершин поглощения,  $S_r$  – список вершин, которым присвоен ранг,  $S_L$  – список неиспользуемых (свободных) вершин  $S_L = V / (S_{in} \cup S_{out} \cup S_r)$ . Введем операции добавления и удаления из списков его элементов:  $add(\cdot, \cdot)$  и  $del(\cdot, \cdot)$ .

#### Алгоритм P-нумерации.

Шаг 0. Фиктивной дуге  $(\infty, v_0)$ , входящей в корневую вершину, присваивается ранг равный  $r(\infty, v_0) = -1$ . Если корневая вершина имеет статус «поглощающей», то все входящие в корневую вершину дуги удаляются, а в качестве текущей вершины рассматривается корневая вершина  $v_0$ . Переходим к шагу 1.

Шаг 1. Текущей вершине  $v_{i_k}$ , рассматриваемой на  $k$ -й итерации алгоритма, присваивается ранг:

$$r(v_{i_k}) = \begin{cases} r((v_{i_{k-1}}, v_{i_k})) + 1, & \text{если } v_{i_k} \text{ – вершина ветвления,} \\ \max_b r((v_b, v_{i_k})) + 1, & \text{иначе } ((v_b, v_{i_k}) \in a). \end{cases}$$

Если вершина  $v_{i_k}$  является вершиной ветвления, то выполняется операция пополнения списка  $S_{out} : add(S_{out}, v_{i_k})$ .

Шаг 2. Для текущей вершины  $v_{i_k}$  выбирается непомеченная дуга перехода в следующую вершину. Выбранная дуга помечается рангом вершины  $v_{i_k}$ :

$$r((v_{i_k}, v_{i_{k+1}})) = r(v_{i_k}).$$

Если все дуги, исходящие из этой вершины помечены, то автоматически выполняется операция  $del(S_{out}, v_{i_k})$ .

Шаг 3. Анализируется статус вершины  $v_{i_{k+1}}$ , исходящей из  $v_{i_k}$ , в направлении помеченной дуги:  $(v_{i_k}, v_{i_{k+1}}) \in a$ .

Шаг 4. Если вершина  $v_{i_{k+1}}$  не является вершиной поглощения, то в качестве текущей вершины рассматривается вершина  $v_{i_{k+1}}$ . Переходим к шагу 1.

Шаг 5. Иначе, при условии  $v_{i_{k+1}} \notin S_{in}$ , выполняется операция пополнения списка  $S_{in} : add(S_{in}, v_{i_{k+1}})$ . Если не все дуги, входящие в вершину  $v_{i_{k+1}}$  помечены, то из списка  $S_{out}$  выбирается последний элемент  $v_j \in S_{out}$ , который рассматривается как текущий. Переходим к шагу 2.

Шаг 6. Если все дуги, входящие в вершину  $v_{i_{k+1}}$  помечены, то выполняется операция  $del(S_{in}, v_{i_{k+1}})$  и в качестве текущей вершины выбирается  $v_{i_{k+1}}$ . Переходим к шагу 1.;

Шаг 7. Если список  $S_{out}$  – пуст, а в списке  $S_{in}$  имеются элементы  $S_{in} = [v_{i_1}, v_{i_2}, \dots, v_{i_{k+1}}]$ , то это означает, что в графе  $\hat{G}$  имеются контуры. Причем в циклических маршрутах участвует одна или несколько вершин списка  $S_{in}$ . Для идентификации вершин, принадлежащих контурам, для подграфа, составленного из вершин  $V / S_r$ , строится двоичная булева матрица

смежности  $M$  и ее степени  $M^2, M^3, \dots, M^k, k \leq n$  до тех пор, пока не возникнет ситуация  $\exists j (m_{jj}^{(k)} = 1)$ . Последнее означает, что в вершину  $v_j$  существует циклический маршрут, состоящий из  $k$  шагов. Составляется список

$$\tilde{S}_{in} = [v_{j_1}, v_{j_2}, \dots, v_{j_m}] \subseteq S_{in}, m_{j_i j_i}^{(k)} = 1, i = 1, \dots, m.$$

Из списка  $\tilde{S}_{in}$  выбирается вершина  $v_b$ , имеющая наименьшее количество предшественников, не принадлежащих спискам  $S_{in}, S_r$ . В графе удаляются все непомянутые дуги, входящие в вершину  $v_b$  и выполняется операция  $del(S_{in}, v_b)$ , т.е. производится расконтуривание графа. В качестве текущей вершины выбирается вершина  $v_b$ . Переходим к шагу 1.

Шаг 8. Если списки  $S_{out}$  и  $S_{in}$  пусты и достигнута конечная вершина, то переходим к шагу 9.

Шаг 9. Вершины орграфа нумеруются послойно от корневой вершины до конечной. Под слоем графа будем понимать вершины равного ранга.

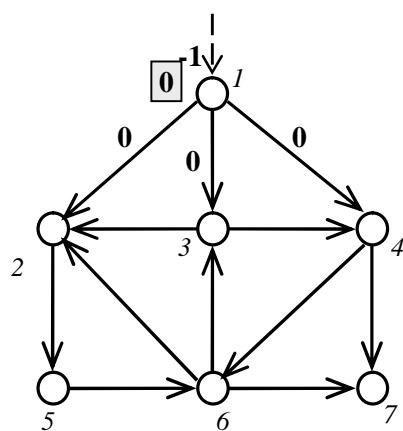
**Утверждение**

Для двухполюсного орграфа алгоритм  $R$ -нумерации реализует правильную нумерацию его вершин.

**Пример**

Рассмотрим работу алгоритма на примере контурного графа, представленного на рисунке 3. Разметка, приведенная на рис. 3, соответствует случаю, когда алгоритм обнаружил первый контур графа. Дуги размечены рангом вершины, из которых они исходят. Из рисунка видно, что ранг равный нулю приобрела вершина 1, размечены все исходящие дуги 1-2, 1-3, 1-4.

Для определения контурных вершин на множестве  $V/S_r$  построим матрицу смежности и матрицы её степеней.



$$S_{in} = [2, 3, 4],$$

$$S_{out} = \emptyset,$$

$$S_r = [1].$$

Рис. 3

$$M = \begin{matrix} & \begin{matrix} 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix};$$

$$M^2 = \begin{matrix} & \begin{matrix} 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{vmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix};$$

$$M^3 = \begin{matrix} & \begin{matrix} 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{vmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix}.$$

Из матрицы  $M^3$  видно, что в циклических маршрутах участвуют вершины 2, 3, 4, 5, 6. Причем вершины 2, 3, 4 входят в список  $S_{in}$  и каждая из них связана с циклами. Например, вершина 2 входит в цикл 2-5-6-2; вершина 3 – 3-4-6-3 и 3-2-5-6-3; вершина 4 – 4-6-3-4. В качестве разрешающего элемента выбирается вершина 3, поскольку в неё входит только одна дуга, а в цикле участвуют две вершины (3 и 4). Из графа удалим дугу 6-3. В результате получим разметку, представленную на рис. 4, а).

Алгоритм вновь «наталкивается» на новый контур 2-5-6-2, который определяется из матрицы:

$$M^3 = \begin{array}{c|ccccc} & 2 & 5 & 6 & 7 \\ \hline 2 & 1 & 0 & 0 & 1 \\ 5 & 0 & 1 & 0 & 0 \\ 6 & 0 & 0 & 1 & 0 \\ 7 & 0 & 0 & 0 & 0 \end{array}.$$

Удалив дугу 6-2, получим бесконтурный граф, представленный на рис. 4, б) с окончательной разметкой вершин графа. Правильная нумерация графа представлена на рис. 5.

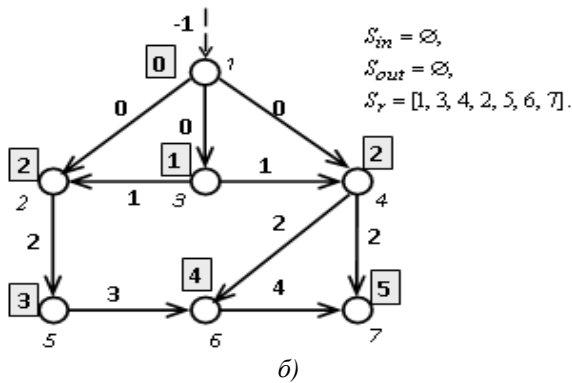
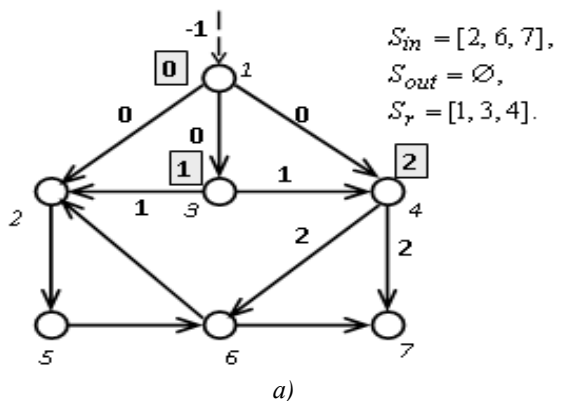


Рис. 4

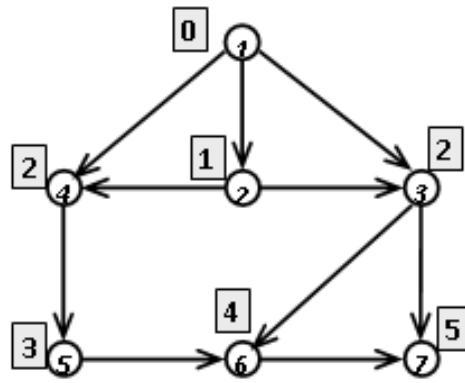


Рис. 5

### Заключение

В работе предложен эффективный универсальный алгоритм правильной нумерации двухполосного направленного графа. Его универсальность заключается в том, что он одновременно позволяет выявлять контуры в графе, реализовывать их расконтуривание и производить правильную нумерацию вершин графа. Предложенный алгоритм полезен на этапе анализа параллельного алгоритма, созданного разработчиком с помощью средства PGRAPH, поскольку реальная модель в отличие от классической последовательно-параллельной бесконтурной схемы, может содержать контуры в последовательных ветках алгоритма (см. рис. 1, б) и в циклах исполнения параллельных «группировок» (см. рис. 1, в). В то время как в области параллельного программирования, корректность разрабатываемой модели алгоритма имеет решающее значение для обеспечения качества программного продукта.

### Библиографический список

1. Коварцев, А.Н. Автоматизация разработки и тестирования программных средств [Текст] / А.Н. Коварцев. – Самара: Самарский государственный аэрокосмический университет, 1999. – 150 с.
2. Жидченко, В. В. Моделирование синхронных параллельных вычислений при построении математических моделей сложных систем [Текст] / В. В. Жидченко, А. Н. Коварцев // Первая международная конференция «Системный анализ и информационные технологии» САИТ-2005: Труды конференции. В 2 т. – Т.2. – М.: КомКнига, 2005. – С. 154-160.

3. Богомолов А.М., Салий В.Н. Алгебраические основы теории дискретных систем. – М.: Наука, 1997. - 368 с.

## **CORRECT NUMERING OF TWO-POLE DIRECTED GRAPHS**

© 2012 D. A. Popova-Kovartseva

Samara State Aerospace University named after academician S. P. Korolyov  
(National Research University)

In many problems of the system analysis a graph theory is widely used, which allows using the mathematical apparatus of analysis of binary matrices to obtain important information about the internal structure and properties of the simulated objects. In particular, algorithms for proper numbering directed graphs are used in problems of topological sorting acyclic graphs. The article deals with the universal topological sorting algorithm that allows for the correct numbering, including contour digraphs.

*Parallel program, directed graph, correct numering, top of branching, top of absorption, rank of top.*

### **Информация об авторе**

**Попова-Коварцева Дарья Александровна**, аспирант кафедры программных систем, Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет). E-mail: [DakKovr@mail.ru](mailto:DakKovr@mail.ru). Область научных интересов: автоматизации программирования, системный анализ.

**Popova-Kovartseva Daria Alexandrovna**, post-graduate student of the sub-department software systems, Samara State Aerospace University named after academician S. P. Korolyov (National Research University). E-mail: [DakKovr@mail.ru](mailto:DakKovr@mail.ru). Area of scientific: automated programming, systems analysis.