

Развитие графовых и матричных способов представления алгоритмов

В.С. Поляков, С.В. Поляков, О.А. Авдеюк, В.Ю. Наумов,

Е.С. Павлова, М.Г. Скворцов

Волгоградский государственный технический университет

Аннотация: В статье рассмотрен вопрос последовательного доопределения граф-схемы абстрактного алгоритма и представления его двудольным графом. Приведен пример возможности задания алгоритмов в матрично-предикатном виде. В заключении указано, что полученный в результате матрично-предикатный вид полностью идентичен графической форме, но вместе с тем позволяет работать с алгоритмами значительно эффективнее, так как появляется возможность задавать их в матрично-предикатном виде и частично автоматизировать эвристические методы их построения.

Ключевые слова: алгоритм, граф-схема, двудольный граф, множество, матрица, предикат, логический оператор.

Известно, что алгоритм – это конечный набор правил, позволяющий чисто механически решить поставленную задачу. Большинство алгоритмов, применяемых в настоящее время, являются последовательными. Запись алгоритма осуществляется в следующем виде: словесная или вербальная (языковая, формульно-словесная и т.д.); графическая (например, блок-схемы, ДРАКОН-схемы, диаграммы Насси- Шнейдермана); в виде псевдокодов, представляющие собой частично формализованные описания алгоритмов, которое состоит из отдельных фраз естественного языка и некоторых элементов языка программирования; – программная – тексты алгоритма записываются на языке программирования (программа). Разработка алгоритмов осуществляется, как правило, человеком и носит эвристический характер. Процессы разработки алгоритмов и формализации взаимодействия их между собой сложны и трудоёмки, поэтому плохо поддаются автоматизации. Поэтому актуальной задачей является разработки такой формы представления алгоритма, которая позволит автоматизировать эвристические методы их построения.

Организацию взаимодействия алгоритмов между собой (например, параллельную их работу) можно разбить на два этапа:

- представление алгоритма в виде удобном для обработки в ЭВМ,
- разработка операций над алгоритмами.

Рассмотрим произвольный алгоритм, заданный в виде граф-схемы (рис. 1).

При таком задании выделяются два типа вершин:

- вершины, определяющие выполнение отдельных операций (множество $A = \{ A_0, A_1, A_2, A_3, A_i, A_{i+1}, A_k \}$, операторы действия);
- вершины, определяющие логику (порядок) перемещения от выполнения одной операции к выполнению следующей ($\alpha = \{ \alpha_{1-1}, \alpha_{1-2}, \alpha_{2-1}, \alpha_{2-2} \}$, операторы логики).

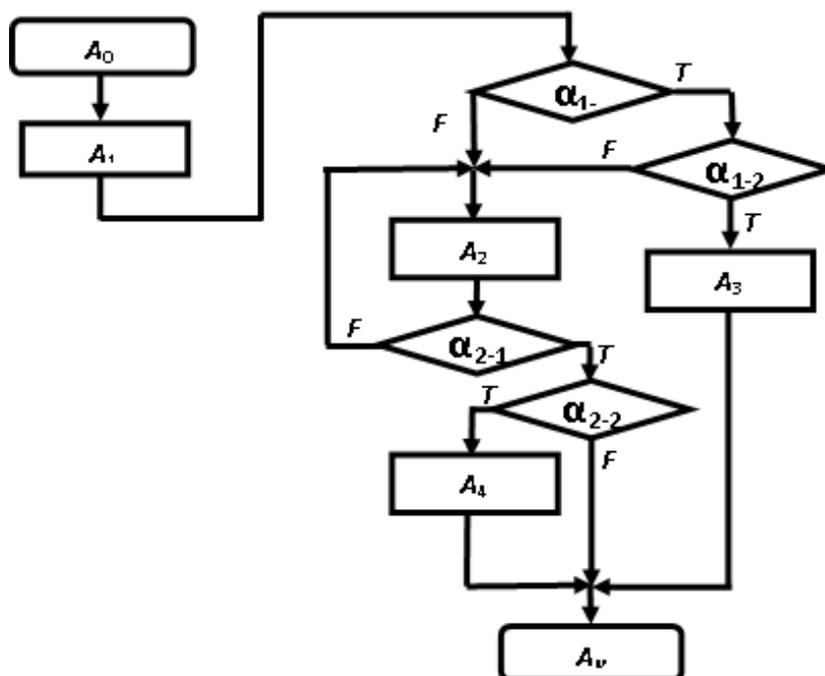


Рис. 1. – Граф-схема произвольного алгоритма

Такой способ работы с алгоритмами имеет следующие недостатки: переход от одного оператора действия к другому в некоторых случаях ничем не обозначен; определяющие логику выполнения алгоритма условия часто задаются несколькими логическими функциями. Для устранения этих

проблем предлагается доопределить соответствующие операторы следующим образом и использовать указанную ниже последовательность действий:

1. Будем задавать конец выполнения каждого оператора действия A_j с помощью соответствующего оператора логики α_j (рис. 2).

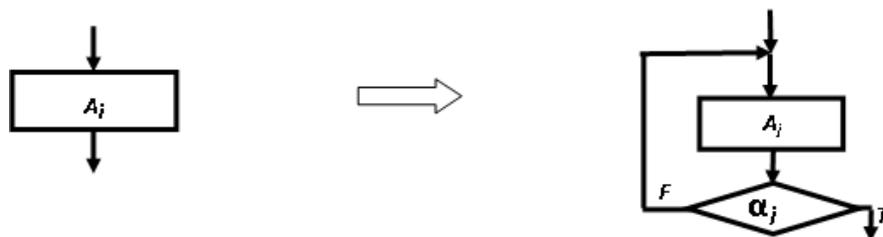


Рис. 2. – Доопределение оператора действия A_j оператором логики α_j

2. Различные композиции операторов логики заменим на один многозначный оператор. На рис. 3а приведены две логические функции α и β , а на рис. 3б реализующий аналогичную функцию многозначный оператор $\alpha \cdot \beta$.

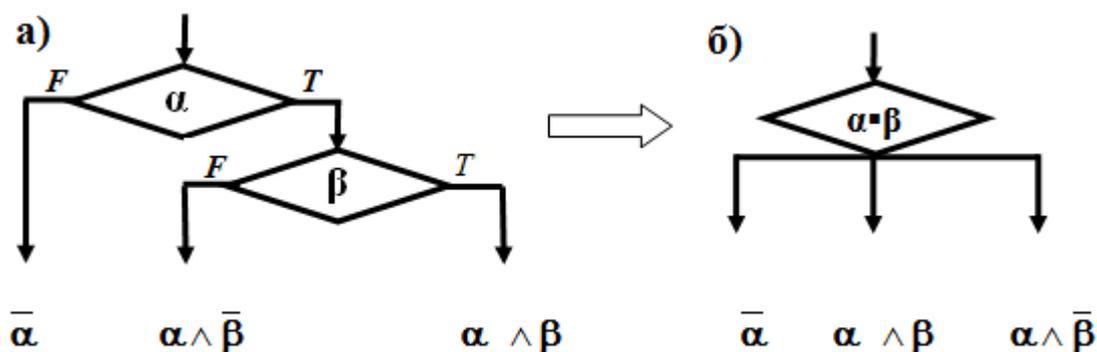


Рис. 3. – Замена двух операторов логики одним многозначным оператором

3. Доопределяем каждый оператор действия исходного алгоритма A_i алгоритма A как показано на рисунках 4а, б, в, г, д.
4. Введем следующие обозначения:

Таблица № 1

Композиция и логические значения функций

Композиция функций	Логические значения функций			
$\alpha_{00} = \mu^0$	$\bar{I} = \mu^0_0$	$I = \mu^0_1$		
$\alpha_{01} \alpha_{11} = \mu^1$	$\bar{\alpha}_{01} = \mu^1_0$	$\alpha_{01} \bar{\alpha}_{1-1} = \mu^1_1$	$\alpha_{01} \alpha_{1-1} \bar{\alpha}_{1-2} = \mu^1_2$	$\alpha_{01} \alpha_{1-1} \alpha_{1-2} = \mu^1_3$
$\alpha_{02} \alpha_{2-1} \alpha_{2-2} = \mu^2$	$\bar{\alpha}_{02} = \mu^2_0$	$\alpha_{02} \bar{\alpha}_{2-1} = \mu^2_1$	$\alpha_{02} \alpha_{2-1} \bar{\alpha}_{2-2} = \mu^2_2$	$\alpha_{02} \alpha_{2-1} \alpha_{2-2} = \mu^2_3$
$\alpha_3 = \mu^3$	$\bar{\alpha}_3 = \mu^3_0$	$\alpha_3 = \mu^3_1$		
$\alpha_4 = \mu^4$	$\bar{\alpha}_4 = \mu^4_0$	$\alpha_4 = \mu^4_1$		

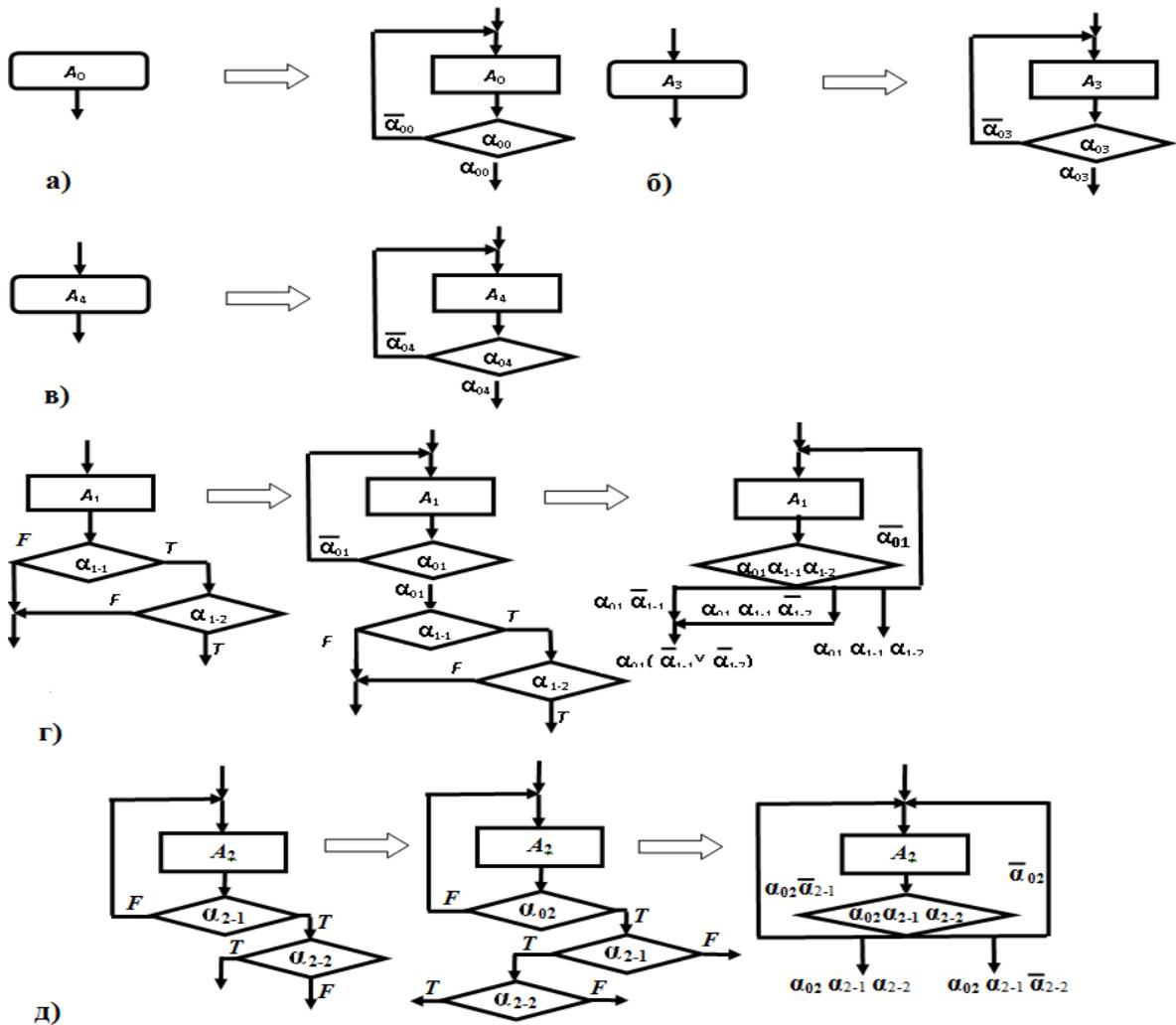


Рис. 4. – Доопределение оператора действия A_i алгоритма A

5. Применим доопределённые операторы действия (рис.4) и новые обозначения (таблица 1) для исходного алгоритма (рис. 1) и получим новый доопределённый алгоритм (рис. 5).

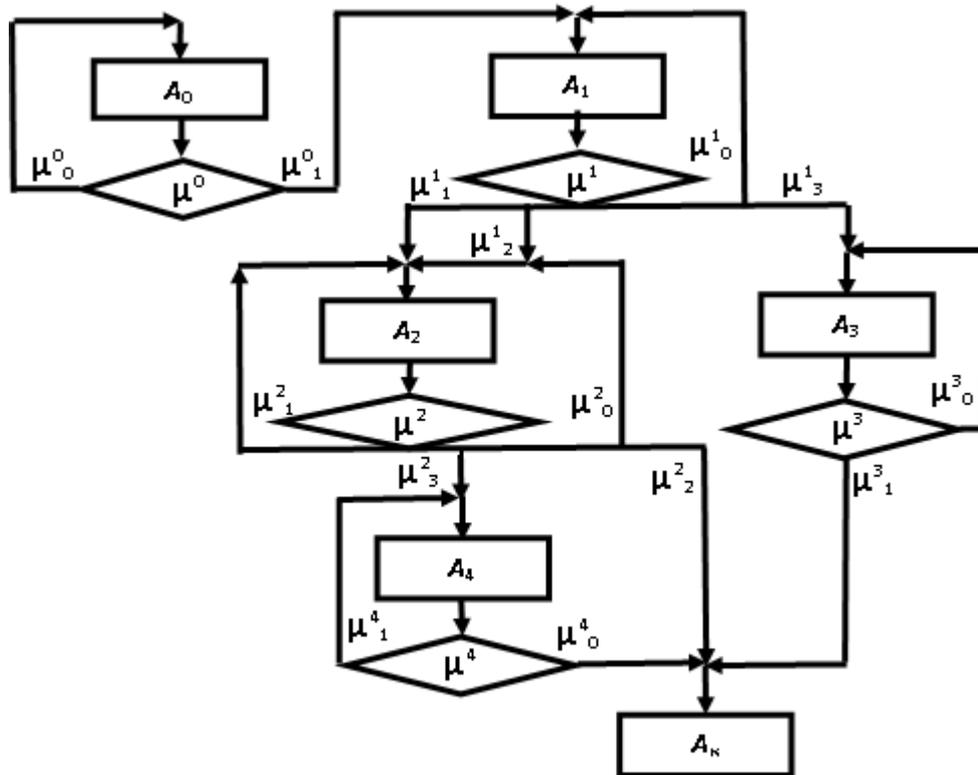


Рис. 5 – Доопределённый алгоритма A с новыми обозначениями

6. Доопределённый алгоритм (рис. 5), можно представить двудольным графом, то есть ориентированным графом с двумя типами непересекающихся вершин (рис. 6) [3]. При этом вводятся обозначения: $\mu_{1,2}^1 = \mu_{1,1}^1 \vee \mu_{1,2}^1$ и $\mu_{0,1}^2 = \mu_{0,0}^2 \vee \mu_{0,1}^2$

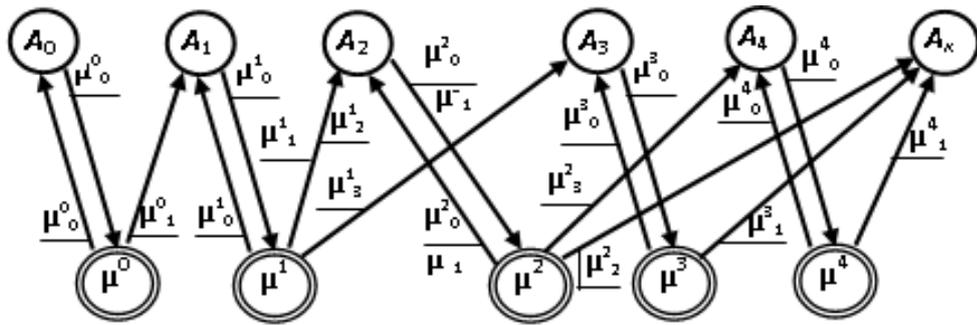


Рис. 6 – Двудольный граф алгоритма A

7. Двудольный граф (рис. 6) запишем оператором действия исходного алгоритма Op^A в матрично-предикатном виде :

$$Op^A = \begin{array}{c|cccc|cccc|c} \mu^0 & 0 & 0 & 0 & 0 & \mu^0 \mu_0^0 A_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu^1 & 0 & 0 & 0 & 0 & \mu^1 \mu_{0,1}^1 A_1 & \mu^1 \mu_{1,2}^1 A_2 & \mu^1 \mu_3^1 A_3 & 0 & 0 \\ 0 & 0 & \mu^2 & 0 & 0 & 0 & 0 & \mu^2 \mu_{0,1}^2 A_2 & 0 & \mu^2 \mu_3^2 A_4 & \mu^2 \mu_2^2 A_k \\ 0 & 0 & 0 & \mu^3 & 0 & 0 & 0 & 0 & \mu^3 \mu_0^3 A_3 & 0 & \mu^3 \mu_3^3 A_k \\ 0 & 0 & 0 & 0 & \mu^4 & 0 & 0 & 0 & 0 & \mu^4 \mu_0^4 A_4 & \mu^4 \mu_4^4 A_k \\ \hline A_0 \mu_0^0 \mu^0 & 0 & 0 & 0 & 0 & A_0 \mu_0^0 A_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 \mu_0^1 \mu^1 & 0 & 0 & 0 & 0 & A_1 \mu_0^1 A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_2 \mu_0^2 \mu^2 & 0 & 0 & 0 & 0 & A_2 \mu_0^2 A_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_3 \mu_0^3 \mu^3 & 0 & 0 & 0 & 0 & A_3 \mu_0^3 A_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_4 \mu_0^4 \mu^4 & 0 & 0 & 0 & 0 & A_4 \mu_0^4 A_4 & 0 \\ \hline \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_k \alpha A_k \end{array}$$

Полученный в результате матрично-предикатный вид полностью идентичен графической форме, но вместе с тем позволяет работать с алгоритмами значительно эффективнее, так как появляется возможность задавать их в матрично-предикатном виде и частично автоматизировать эвристические методы их построения.

Литература

1. Поляков В.С., Поляков С.В., Муха Ю.П. Представление транспортно-информационного потока взаимодействием операторов трассы и движущихся по ней объектов // Телекоммуникации. 2013. № 7. С. 3-7.

2. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая Линия – Телеком, 2007. 452 с.
3. Дворецкий С.И., Муромцев Ю.Л. и др. Моделирование систем. М.: Academia, 2009. 320 с.
4. Поляков С.В., Поляков В.С. Моделирование параллельно протекающих процессов блоками взаимодействующих компонентов. // Контроль. Диагностика. 2008. № 8. С. 70-72.
5. Астанин С.В., Драгныш Н.В., Жуковская Н.К. Вложенные метаграфы как модели сложных объектов // Инженерный вестник Дона, 2012, №4. URL: ivdon.ru/magazine/archive/n4p2y2012/1434/.
6. Поляков В.С., Поляков С.В., Муха Ю.П. Применение совокупности новых установок, представлений и терминов при моделировании информационных потоков в сложных системах // Телекоммуникации. 2013. № 1. С. 2-5.
7. Поляков В.С., Поляков С.В., Нефедьев А.И. Математическая модель подвески подвижной части электроизмерительного прибора.// Инженерный вестник Дона, 2013. №3. URL: ivdon.ru/magazine/archive/n3y2013/1788.
8. Лацис А.О. Параллельная обработка данных. М.: Academia, 2010. 336 с.
9. Voevodin V.V. Information structure of sequential programs // Russ. J. of Num. An. and Math. Modelling. 1995. V.10. №3. P. 279286.
10. Voevodin V.V. Mathematical foundations of parallel computing. World Scientific Publishing Co., Series in computer science. 1992. V.33. 343 p.

References

1. Poljakov V.S., Poljakov S.V., Muha Ju.P. Telekommunikacii. 2013. № 7. pp. 3-7.
-

2. Rutkovskaja D., Pilin'skij M., Rutkovskij L. Nejronnye seti, geneticheskie algoritmy i nechetkie sistemy [Neural networks, genetic algorithms and fuzzy systems]. M.: Gorjachaja Linija. Telekom, 2007. 452 p.
3. Dvoreckij S.I., Muromcev Ju.L. i dr. Modelirovanie system [Modeling systems]. M.: Academia, 2009. 320 p.
4. Poljakov S.V., Poljakov V.S. Kontrol'. Diagnostika. 2008. № 8. pp. 70-72.
5. Astanin S.V., Dragnysh N.V., Zhukovskaja N.K. Inženernyj vestnik Dona (Rus), 2012, №4. URL: ivdon.ru/magazine/archive/n4p2y2012/1434.
6. Poljakov V.S., Poljakov S.V., Muha Ju.P. Telekommunikacii. 2013. № 1. pp. 2-5.
7. Poljakov V.S., Poljakov S.V., Nefed'ev A.I. Inženernyj vestnik Dona (Rus), 2013. №3. URL: ivdon.ru/magazine/archive/n3y2013/1788.
8. Lacis A.O. Parallel'naja obrabotka dannyh [Parallel processing of data]. M.: Academia, 2010. 336 p.
9. Voevodin V.V. Information structure of sequential programs. Russ. J. of Num. An. and Math. Modelling. 1995. V.10. №3. pp. 279286.
10. Voevodin V.V. Mathematical foundations of parallel computing. World Scientific Publishing Co., Series in computer science. 1992. V.33. 343 p.