**Инженерный вестник Дона**, №2 (2020)
ivdon.ru/ru/magazine/archive/n2y2020/5323

# Методика доопределения последовательного алгоритма

*В.С. Поляков, О.А. Авдеюк, Е.С. Павлова, И.Г. Лемешкина,*
*И.В. Приходькова, Р.Н. Никулин*

*Волгоградскийй государственный технический университет*

**Аннотация:** В статье описан вариант задания последовательных алгоритмов в виде двудольных графов путём их доопределения, что дает возможность в дальнейшем работать с алгоритмами методами теории графов. Рассмотрены две формы задания: модульная и функционально-предикативная. Показана возможность задания алгоритма в таблично-предикатном виде. Сделан вывод о том, что помимо стандартных способов описания алгоритма, он может быть представлен в матрично-предикатном виде, что в дальнейшем делает возможным при работе с алгоритмами использовать также методы теории матриц и методы теории предикатов; задание алгоритма в матрично-предикатном виде позволяет избежать изоморфизма при проведении алгебраических и теоретико-множественных операций над ним; представление алгоритмов в матрично-предикатном виде делает возможным осуществлять с ними различные операции.
**Ключевые слова:** граф-схема алгоритма, последовательный алгоритм, предикативный блок, функциональный блок, доооопределение, двудольный граф, таблично-предикативная форма, теория графов, изоморфизм.

Среди распространённых вариантов описания алгоритмов [1,2] наибольшее распространение получили графические способы [3,4]. Поскольку граф - схемы алгоритмов представляют собой графы, то их в дальнейшем можно анализировать и преобразовывать, привлекая методы теории графов. Также в результате анализа общепринятых подходов к представлению алгоритмов [5,6] выявлено, что они, как правило, позволяют описывать процессы без учета разделения на параллельные потоки данных, что часто требуется на практике при управлении сложными техническими объектами. Поэтому предлагается методика, которая позволяет использовать возможности интеллектуальных систем для создания алгоритмов. В данной работе приводится лишь один аспект этого подхода к описаниям алгоритма в матрично-предикатном виде – в виде двудольного графа.

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020

Рассмотрим произвольную граф-схему последовательного алгоритма (ГСА) (рис. 1).

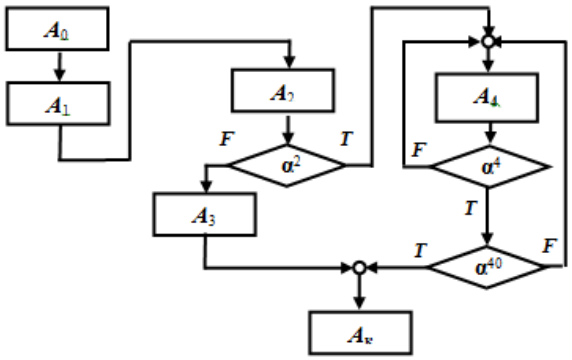
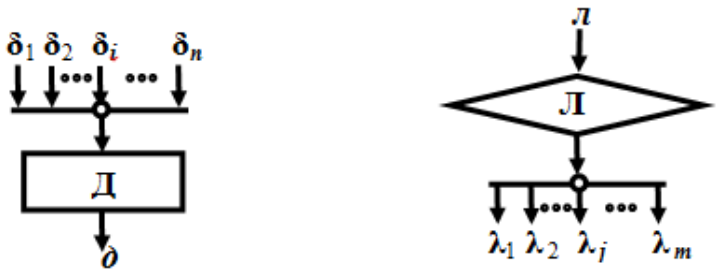



Рис. 1. – Произвольная граф - схема алгоритма

На рис .1 приняты следующие обозначения: $A=\{A_0, A_1, A_2, A_3, A_4, A_к\}$ – функциональные блоки; $\alpha=\{\alpha^2, \alpha^4, \alpha^{40}\}$ – предикативные (логические) блоки.

Недостатками работы с алгоритмами, описанными на рисунке 1, являются зачастую как отсутствие четкого перехода при смене операций, так и сложность логических функций в условных операторах. Первый из этих недостатков можно исправить, заменяя последовательно соединённые операторы одним, при фиксации окончания его работы, т.е. получить модульные блоки алгоритма. Назовем такую операцию операций доопределения. Рассмотрим более подробно процесс получения модульных блоков алгоритма, состоящих из связанных между собой функционального – **Д** (рис. 2, а) и предикативного – **Л** (рис. 2, б) блоков:




а) б)

Рис. 2. – Функциональный Д (а) и предикативный **Л** (б) блоки

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020

Здесь:

$\lambda_1\ \lambda_2\ \ldots\lambda_j\ \ldots\ \lambda_m$ – истинные значения предиката, вырабатываемые предикативным блоком – **Л**,

***л*** – входной сигнал, поступающий на вход предикативного блока (ПБ) – **Л**,

$\delta_1\ \delta_2\ \delta_i\ \ \delta_n$ – сигналы, поступающий на вход функционального блока (ФБ) – **Д**, являющиеся истинными значениями предиката, вырабатываемыми соседними предикативными блоками;

***д*** – выходной сигнал, вырабатываемый функциональным блоком – **Д**.

Для получения модульного вида алгоритма необходимо зафиксировать окончание работы каждого ФБ определенным ПБ. Если между ФБ отсутствует ПБ, то определим предикативную вершину $\alpha^i_0$ (рис. 3):

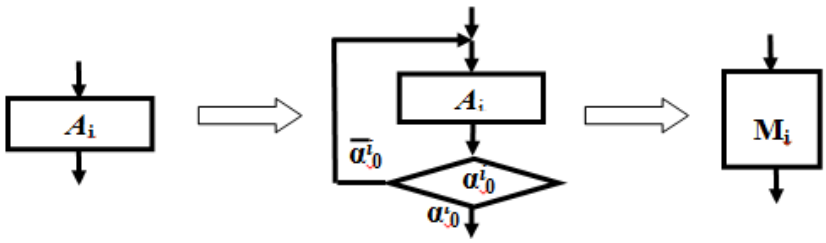



Рис.3. – Алгоритм получения модуля функциональной вершины

В случае существования после ФБ некоторого количества ПБ, без фиксации окончания работы ФБ, необходимо определить предикативную вершину (ПВ) как $\alpha^i_0$, и объединить все ПВ в единый многозначный ПБ (рис. 4):

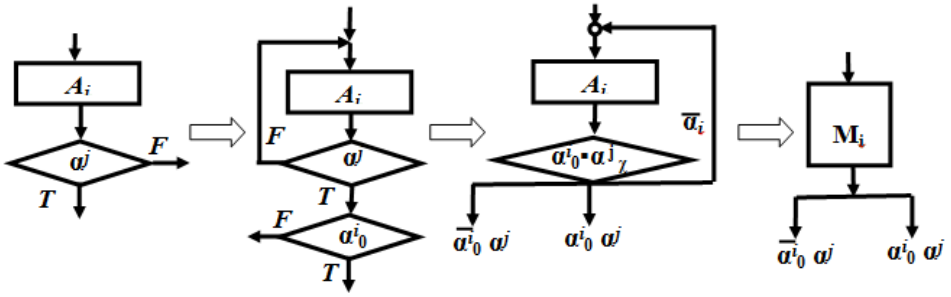
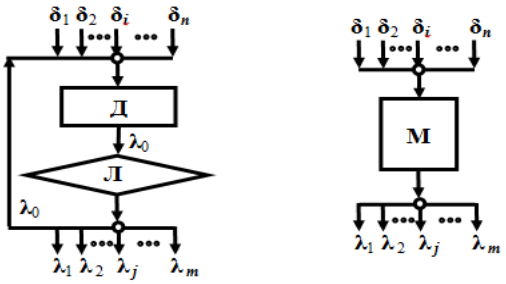



Рис.4. – Алгоритм получения модуля предикативной вершины

Связанная пара, состоящая из ФБ – **Д** и ПБ – **Л** (рис. 5,а), будет называться функционально-предикативным модулем (ФПМ) – **М** (рис. 5,б):

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020




а) б)

Рис.5. – Связь ФБ и ПБ (а) и ФПБ (б)

По аналогии, используя вышеописанные принципы получения модулей, доопределим ФБ рассматриваемого алгоритма (рис. 6). После завершения последовательности операций доопределения, получаем следующую структуру алгоритма в виде набора модульных блоков (рис.7):

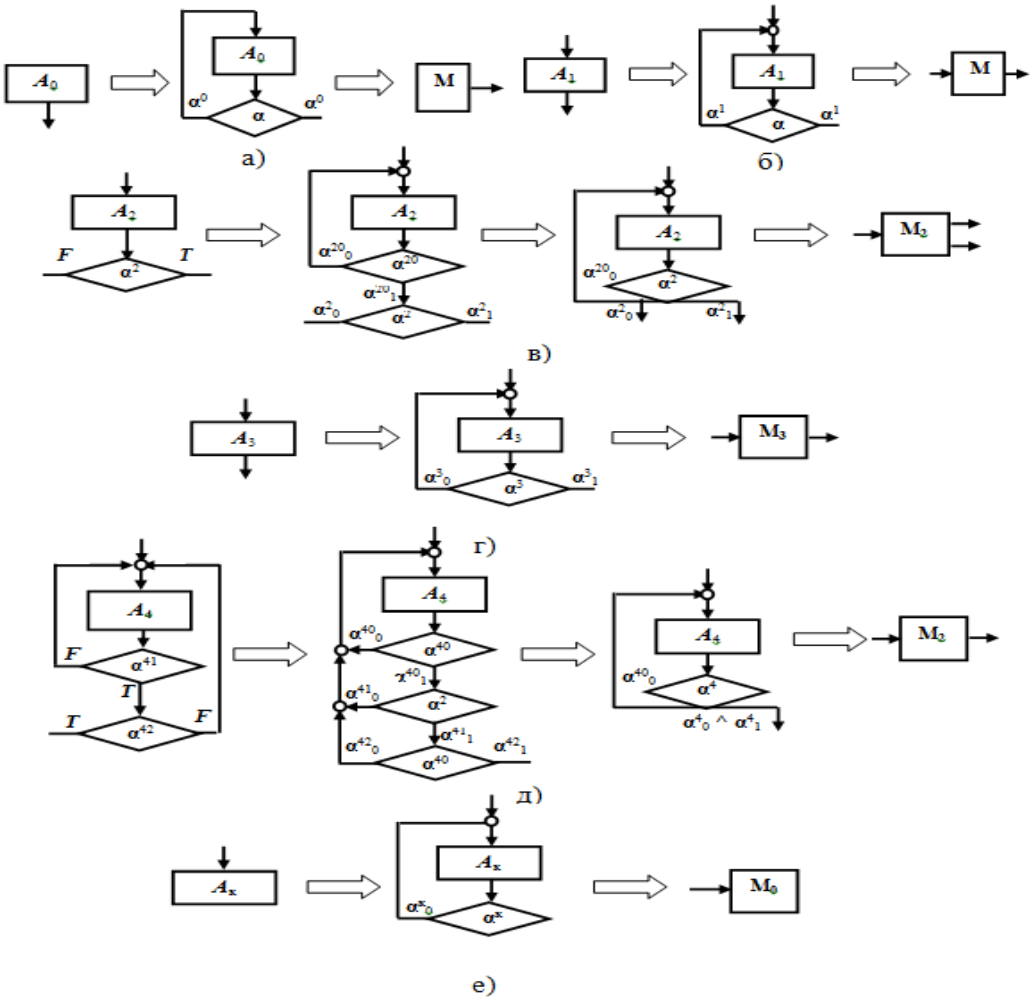



Рис.6. – Последовательность доопределения ФБ $A_i$ (j)

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020

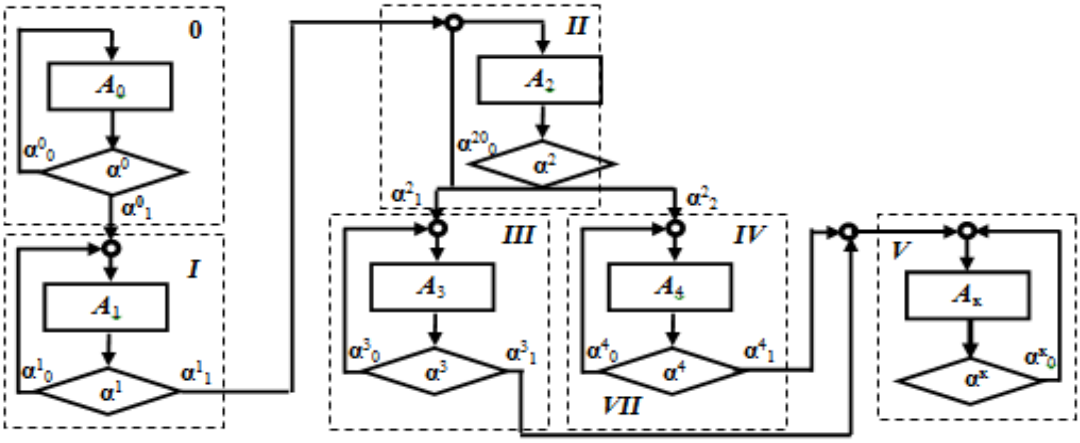



Рис.7. – Алгоритм после операции доопределения

На рис.8 представлена более компактная форма представления алгоритма в виде набора модулей (рис. 8,а), а также его графовая форма – двудольный граф Бержа (рис. 8,б).

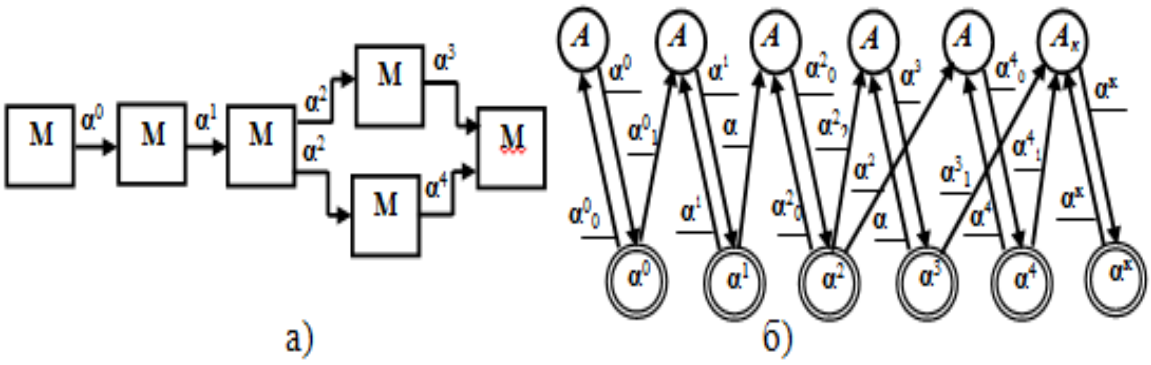



Рис.8. – Модульная запись алгоритма (а) и в виде двудольного графа (б)

Как указано в [7,8], любой алгоритм, представленный с помощью графа Бержа [9,10], можно описать в виде матрицы $\mathbf{M}^A$ (рис. 9). Подобное задание алгоритма в матрично-предикатном виде (МПВ) также является модульным**.** Причем, если в матрице $\mathbf{M}^A$ по разным частям распределить функциональные и предикативные вершины, описание алгоритма в виде $\mathbf{M}^{A*}$ будет выглядеть, как на рисунке 10.

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020

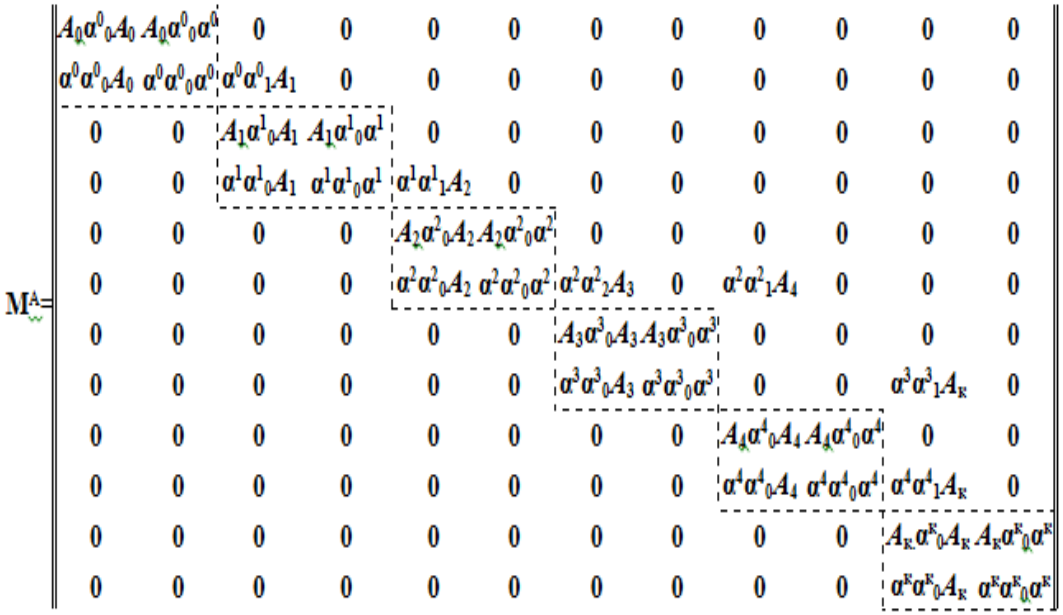
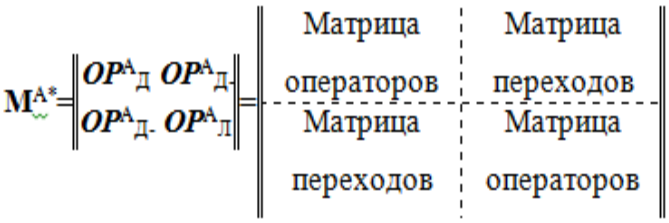


Рис.9. – Модульное представление алгоритма в МПВ



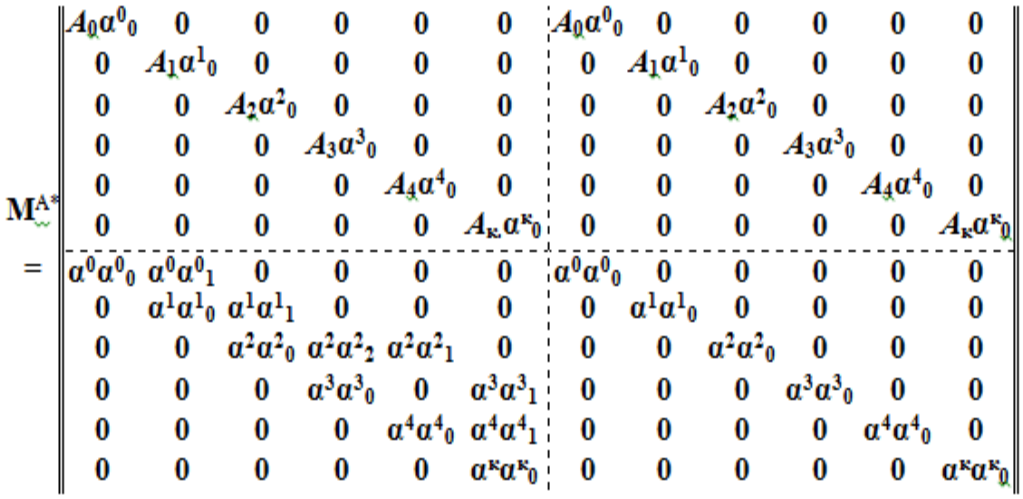



Рис.10. – Функционально-предикативное описание алгоритма в МПВ

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020

Описание алгоритмов в матрично-предикатном виде дает возможность осуществлять множество различных операций над ними. Но при значительном количестве компонентов вывод результатов на носитель информации получается очень громоздким из-за большой разреженности исходных матриц, что, безусловно, является недостатком. Поэтому, при таком соотношении нулевых и значащих элементов матрицы лучше использовать для представления алгоритма таблицу, первая и последняя строки в которой определяют связи истинных значений предиката между собой (таблично-предикатная форма).

Таким образом, помимо стандартных способов описания алгоритма, он может быть задан в матрично-предикатном виде, что позволяет при работе с алгоритмами использовать методы теории матриц и методы теории предикатов. Кроме того, задание алгоритма в матрично-предикатном виде позволяет избежать изоморфизма при проведении алгебраических и теоретико-множественных операций над ними.

## Литература

1. Колмогоров А.Н. Теория информации и теория алгоритмов. Москва: Наука, 1987. 304 с.

2. Альфред В. Ахо, Джон Е. Хопкрофт, Джеффри Д. Ульман. Структуры данных и алгоритмы. Москва: Издательский дом "Вильяме", 2000. 384 с.

3. Янов Ю.И. О логических схемах алгоритмов. Москва: Проблемы кибернетики,1958. 256 с.

4. Макконелл Дж. Основы современных алгоритмов. Москва: Техносфера, 2004. 368 с.

5. Гудман С., Хидетниеми С. Введение в разработку и анализ алгоритмов. Москва: Мир, 1981. 368 с.

6. Поляков В.С., Поляков С.В. Представление алгоритма в матрично-предикатном виде // European research. 2016.№ 2. С. 29-35.

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020

7. Поляков В.С., Поляков С.В., Авдеюк О.А., Наумов В.Ю., Павлова Е.С., Скворцов М.Г. Развитие графовых и матричных способов представления алгоритмов // Инженерный вестник Дона, 2017, №2 URL: ivdon.ru/ru/magazine/archive/N2y2017/4145.

8. Поляков В.С., Авдеюк О.А., Никулин Р.Н., Авдеюк Д.Н. Представление конечного автомата в матрично-предикатной форме // Инженерный вестник Дона, 2019, №4 URL: ivdon.ru/ru/magazine/archive /n4y2019/5828.

9. Diestel R., Graph Theory. Springer-Verlag Berlin Heidelberg, 2010. 410 p.

10. Bapat R.B. Graphs and Matrices. Hindustan Book Agency (India), 2010. 171 p.

## References

1. Kolmogorov A.N. Teoriya informacii i teoriya algoritmov [Information theory and algorithm theory]. Moskva: Nauka, 1987. 304 p.

2. Al'fred V. Aho, Dzhon E. Hopkroft, Dzheffri D. Ul'man. Struktury dannyh i algoritmy [Data structures and algorithms]. Moskva: Izdatel'skij dom "Vil'yame", 2000. 384 p.

3. Yanov Yu.I. O logicheskih skhemah algoritmov [About logical schemes of algorithms]. Moskva: Problemy kibernetiki, 1958. 256 p.

4. Makkonell Dzh. Osnovy sovremennyh algoritmov [The foundations of modern algorithms]. Moskva: Tekhnosfera, 2004. 368 p.

5. Gudman S., Hidetniemi S. Vvedenie v razrabotku i analiz algoritmov [Introduction to algorithm development and analysis]. Moskva: Mir, 1981. 368 p.

6. Polyakov V.S., Polyakov S.V. European research. 2016. № 2. pp. 29-35.

7. Polyakov V.S., Polyakov S.V., Avdeyuk O.A., Naumov V.YU., Pavlova E.S., Skvorcov M.G. Inzenernyj vestnik Dona, 2017, №2. URL: ivdon.ru/ru/magazine/archive/N2y2017/4145.

8. Polyakov V.S., Avdeyuk O.A., Nikulin R.N., Avdeyuk D.N. Inzenernyj vestnik Dona, 2019, №4 URL: ivdon.ru/ru/magazine/archive /n4y2019/5828.

9. Diestel R., Graph Theory. Springer-Verlag Berlin Heidelberg, 2010. 410 p.

10. Bapat R.B. Graphs and Matrices. Hindustan Book Agency (India), 2010. 171 p.

© Электронный научный журнал «Инженерный вестник Дона», 2007–2020