Presentation of the algorithm in the form of a matrix-predicate
Poliakov V.[1], Poliakov S.[2] (Russian Federation)
Представление алгоритма в матрично-предикатном виде
Поляков В. С.[1], Поляков С. В.[2] (Российская Федерация)

[1]Поляков Владимир Сергеевич / Poliakov Vladimir – кандидат технических наук, доцент;
[2]Поляков Сергей Владимирович / Poliakov Sergei – кандидат технических наук, доцент,
кафедра вычислительной техники,
Волгоградский государственный технический университет, г. Волгоград

Аннотация: *в работе рассматривается представление алгоритмов в матрично-предикатном виде. Рассмотрены два варианта такого представления алгоритма: модульное и функционально-предикативное. Показана возможность разбиения последнего на четыре составляющие, что облегчает проведение операций над ними.*
Abstract: *the paper considers the idea of algorithms in the form of a matrix-predicate. Two variants of the presentation of the algorithm: a modular and functional and predicative. The possibility of splitting the last four components that facilitates operations on them.*

Ключевые слова: *алгоритм, параллелизм, граф, предикат.*
Keywords: *algorithm, parallelism, graph, predicate.*

Существует несколько способов представления алгоритмов [1 - 3]. Наибольшее распространение получили графические способы, как наиболее наглядные и компактные. На практике чаще всего встречаются граф-схемы алгоритма (ГСА), диаграммы Насси — Шнейдермана, ДРАКОН и др. [3–5].

Современные методики построения алгоритмов хороши и достаточны лишь в тех случаях, когда требуется описывать достаточно простые процессы. В противном случае существует возможность разделения основной ветви алгоритма на несколько. Среди них может быть как одна главная и несколько побочных, так и их равноценность с точки зрения важности. В данной статье рассматривается задание алгоритма в матрично-предикатном виде посредством использования в качестве базы двудольного графа.

Количество блочных символов при различных подходах к заданию алгоритма в графическом виде различно, зависит от поставленных задач и методов их решения. Например, в [6] показано, что любая блок-схема – это ориентированная сеть с вершинами трёх типов: функциональные, предикативные и объединяющие.

Рассмотрим произвольную ГСА (рис. 1).

Здесь:

$\boldsymbol{A=\left\{\ A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\right\}}$ – вершины, определяющие выполнение отдельных операций, будем называть функциональными блоками;

$\boldsymbol{\alpha=\left\{\alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^{40}\right\}}$– вершины, определяющие логику (порядок) выполнения алгоритма, будем называть функциональными или логическими блоками.

Работа с алгоритмами, заданными в таком виде (рис. 1), осложняется следующими недостатками:

- переход от выполнения одной операции к выполнению другой в некоторых случаях ничем не обозначен, например, вершины A_0 – A_1 – A_2, здесь подразумевается, что после выполнения операции A_1 следует переход к выполнению операции A_2, хотя такой переход ничем не фиксируется;
- условия, определяющие порядок выполнения алгоритма, часто задаются несколькими логическими функциями, что усложняет рассмотрение алгоритма.

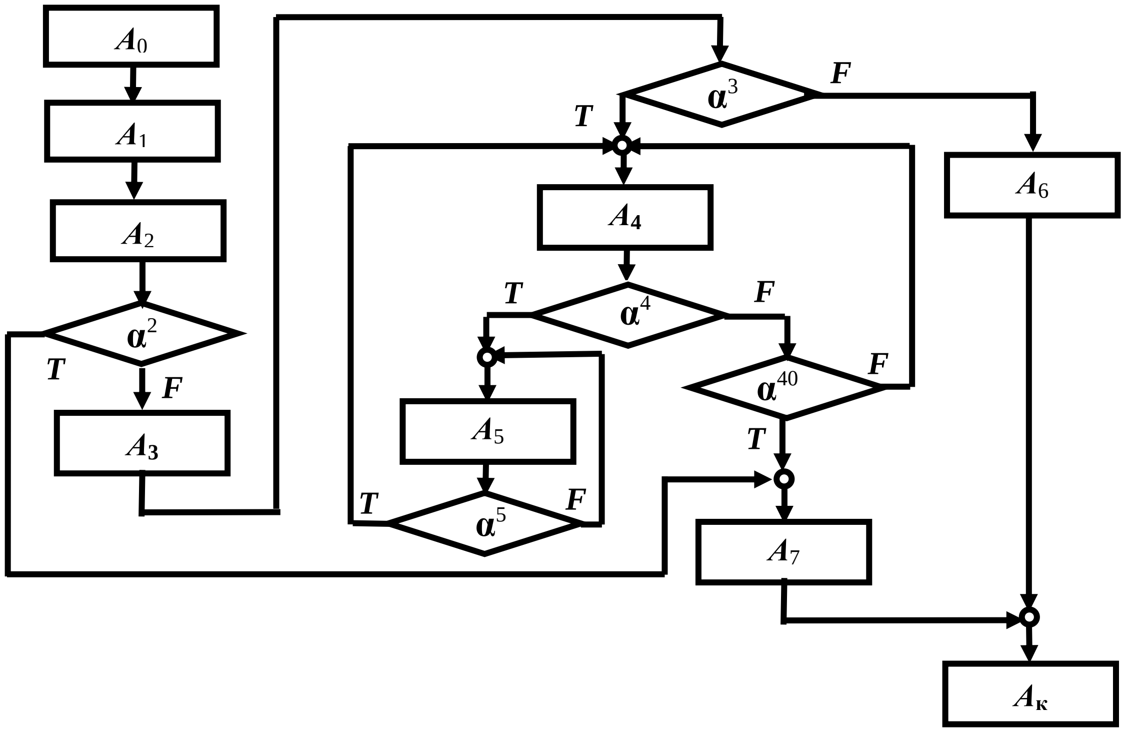


Рис. 1. Граф – схема алгоритма

Возможны два варианта устранения первого из этих недостатков:

1. При наличии последовательно соединённых операторов действия их заменяют на один, например, в рассматриваемом примере операторы. $\boldsymbol{A}_1$ и $\boldsymbol{A}_2$ заменяют на оператор $\boldsymbol{A}_{12}$. В этом случае окончание выполнения объединенного оператора $\boldsymbol{A}_{12}$ будет фиксироваться оператором логики $\boldsymbol{\alpha}^2$.
2. Необходимо определить окончание каждого из функциональных операторов и, тем самым, определить зоны их действия. В результате проведения такой операции приходим к получению модульных блоков алгоритма. Такую операцию будем называть в дальнейшем операций доопределения.

Рассмотрим связанную пару из функционального – **Д** (рис. 2) и предикативного – **Л** (рис. 3) блоков:

Д – блок имеет один выход и может иметь несколько входов;

Л – блок имеет один вход и может иметь несколько выходов.

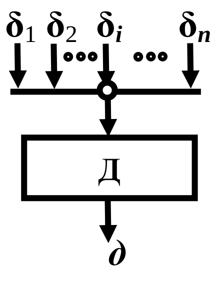


Рис. 2. Функциональный блок

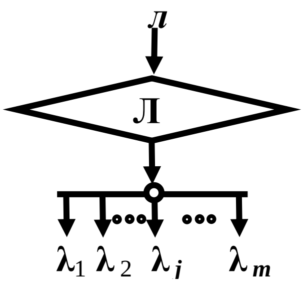
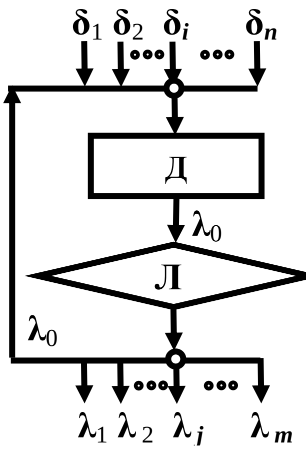


Рис. 3. Предикативный блок

Связанную пару (рис. 4) будем называть функционально-предикатным модулем – **М**.

Внутренняя связь $\boldsymbol{\lambda}_0$ показывает, что работа функционального узла – **Д** проводится до момента исчезновения сигнала. В общем виде модуль – **М** будет выглядеть (рис. 5).



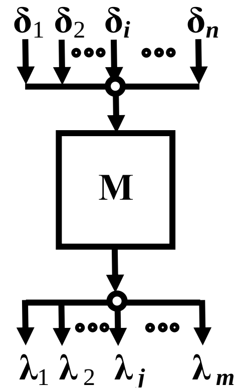


Рис. 4. Функционально-предикатный *Рис. 5. Модульный блок*

Для построения алгоритма в модульном виде проведём ряд предварительных операций. Зафиксируем окончание выполнения каждого функционального блока соответствующим предикативным блоком. Если переход от выполнения одной операции к выполнению другой ведётся напрямую, иначе говоря, если между функциональными блоками отсутствует предикативный, то, введя предикативную вершину $\boldsymbol{\alpha^{i}_{0}}$, получим модуль (рис 6).

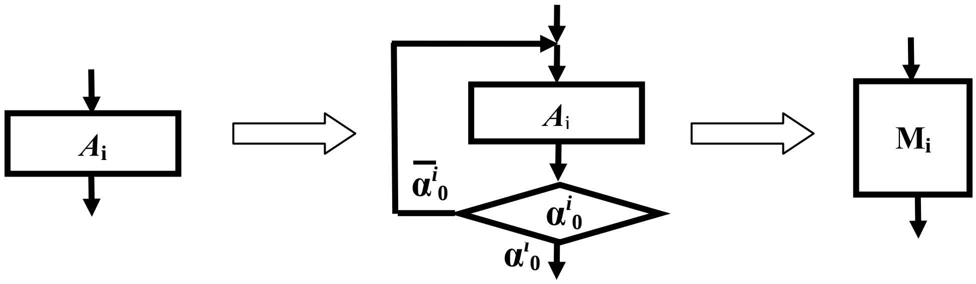


Рис.6. Введение предикативной вершины

Если после функционального блока существует один или несколько предикативных блока, но отсутствует фиксация окончания работы функционального блока, вводим дополнительно предикативную вершину $\boldsymbol{\alpha^{i}_{0}}$. Объединим все предикативные вершины в один многозначный предикативный блок и получим модуль (рис. 7).

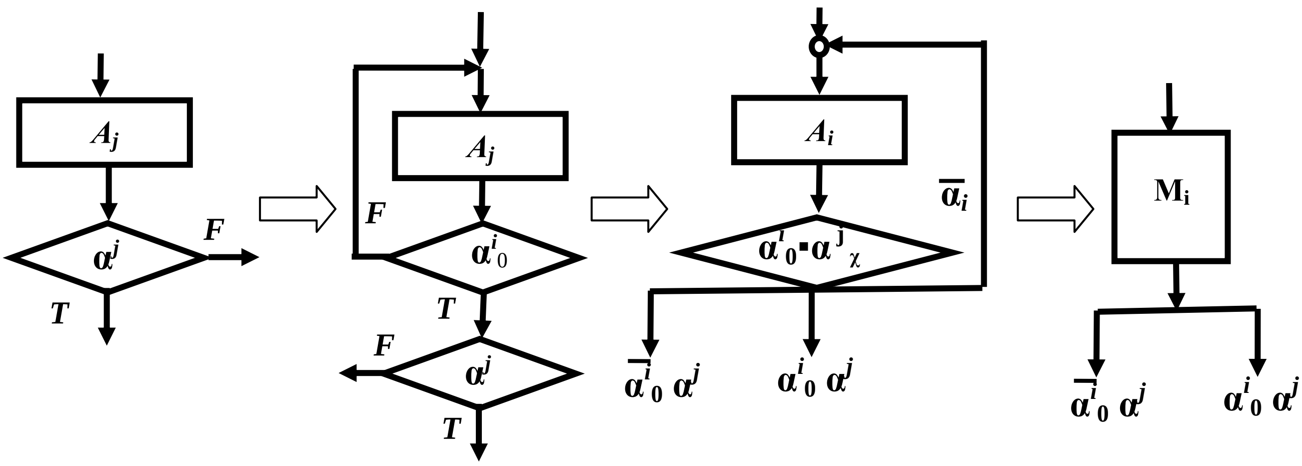


Рис. 7. Переход к модульному блоку

Используя вышеприведённые принципы получения модулей «доопределим» функциональные блоки алгоритма (рис. 1) в итоге исходная ГСА примет вид (рис. 8).

Пунктиром выделены модули алгоритма, а на рисунке (рис. 9) приведено изображение алгоритма в модульном исполнении, которое представляет собой граф Бержа.

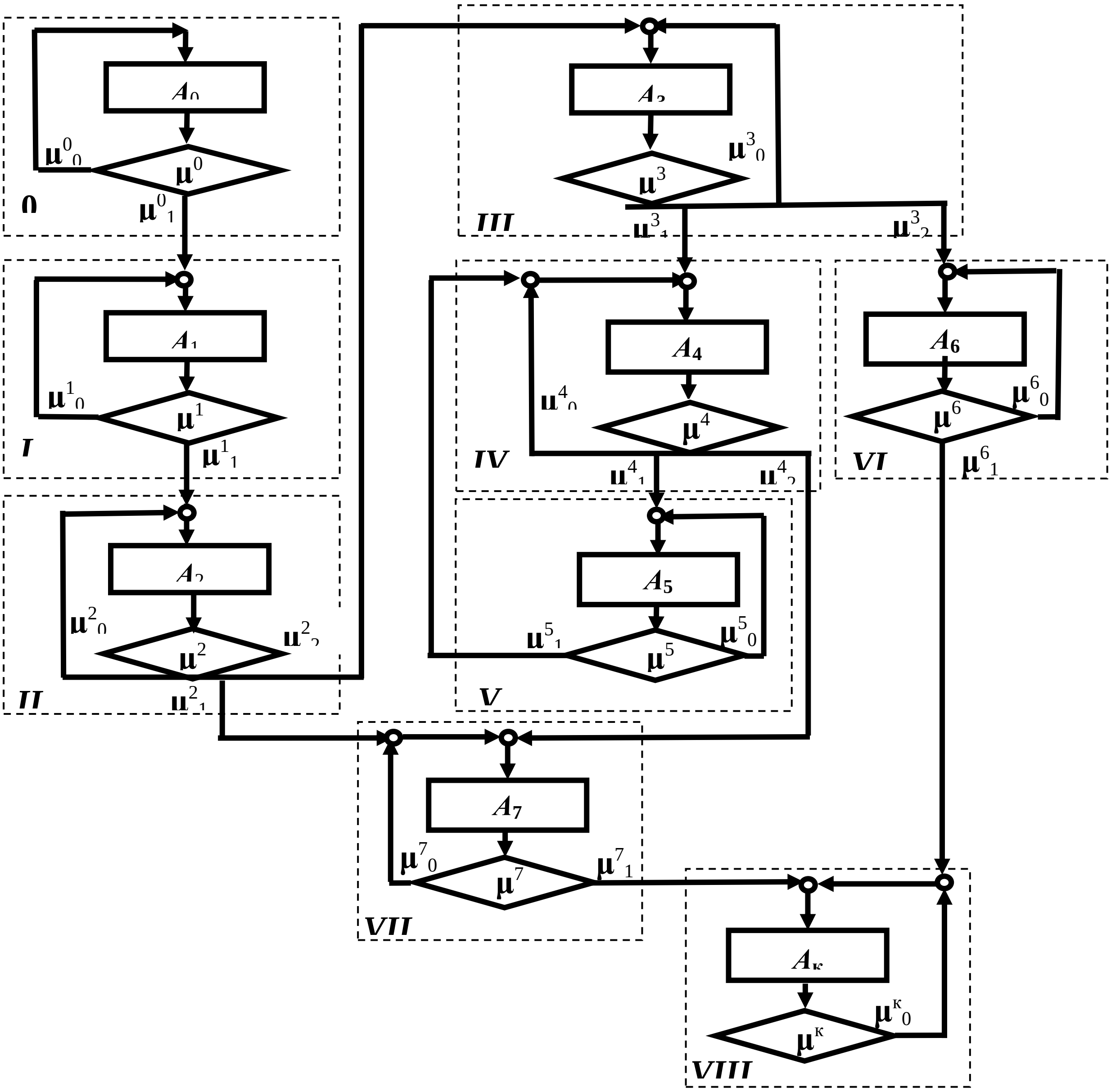


Рис. 8. «Доопределённая» ГСА

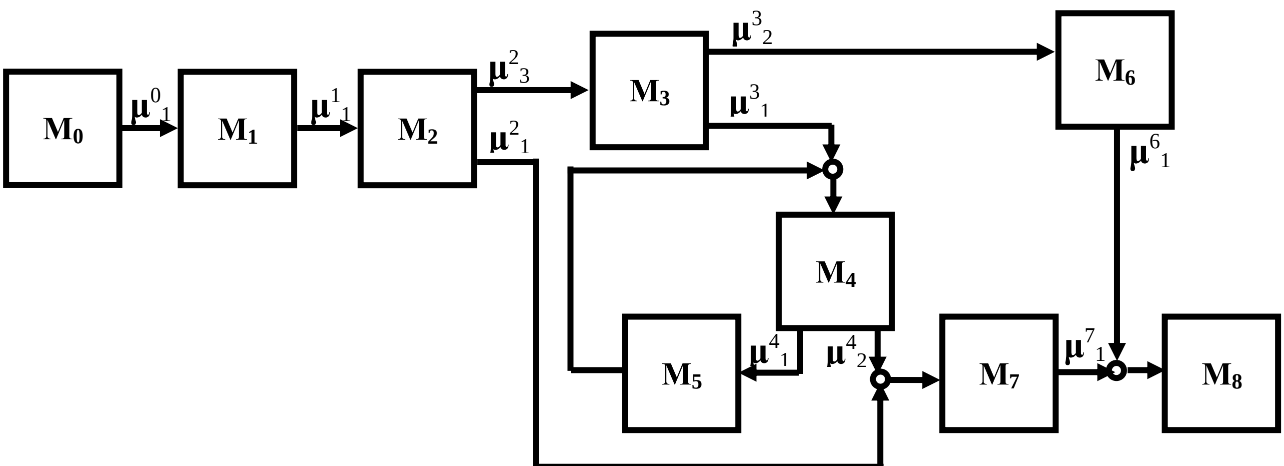


Рис. 9. Модульное представление ГСА

Алгоритм можно задать в виде графа (рис. 10), вершины которого делятся на два непересекающихся множества, то есть граф будет дуальным. Использование матрично-предикатного способа [7 – 9] представления модулей алгоритма позволяет задать алгоритм в виде матрицы (рис. 11). Такое представление алгоритма будем называть модульным в матрично-предикатном виде.

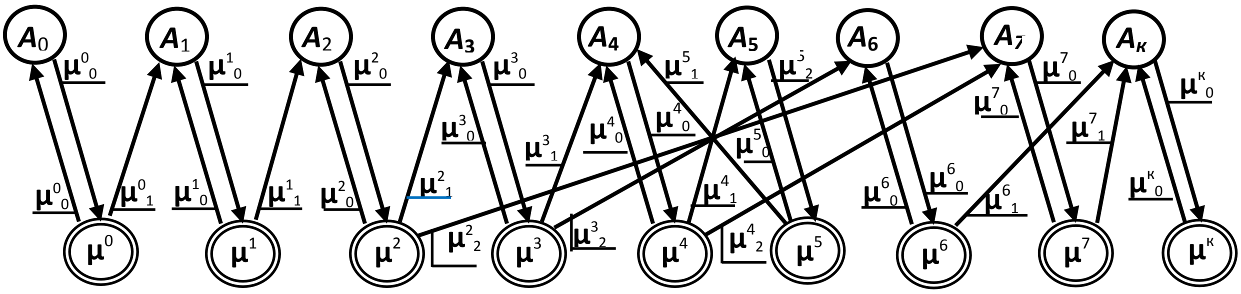


Рис. 10. Представление ГСА в виде двудольного графа

Квадратная матрица алгоритм $\mathbf{M}^{\mathrm{A}}$, заданная в матрично-предикатном виде, обладает свойством неизменности свойств описываемого объекта при одновременной замене строки и столбца с одинаковыми номерами на соответствующую пару с другим номером.

Проведём такую операцию с матрицей $\mathbf{M}^{\mathrm{A}}$, разнеся функциональные и предикативные вершины алгоритма в противоположные стороны. В итоге получим описание того же алгоритма в виде в $\mathbf{M}^{\mathrm{A*}}$ в несколько иной форме (рис. 12). Такое представление алгоритма будем называть **функционально-предикативным** в матрично-предикатной форме.

Матрицу, заданную таким способом, легко представить в виде четырех частей.

$$\mathbf{M}^{\mathrm{A*}} = \left\| \begin{array}{cc} \boldsymbol{OP}^{\mathrm{A}}_{\text{Д}} & \boldsymbol{OP}^{\mathrm{A}}_{\text{Д-Л}} \\ \boldsymbol{OP}^{\mathrm{A}}_{\text{Д-Л}} & \boldsymbol{OP}^{\mathrm{A}}_{\text{Л}} \end{array} \right\| = \left\| \begin{array}{c|c} \text{Матрица операторов действия} & \text{Матрица переходов операторы действия– –операторы логики} \\ \hline \text{Матрица переходов операторы логики– –операторы действия} & \text{Матрица операторов логики} \end{array} \right\|$$

$$\mathbf{M}^{\mathrm{A}} = \left\| \begin{array}{cccccccccccccccccc}
A_0\mu^0{}_0A_0 & A_0\mu^0{}_0\mu^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mu^0\mu^0{}_0A_0 & \mu^0\mu^0{}_0\mu^0 & \mu^0\mu^0{}_1A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A_1\mu^1{}_0A_1 & A_1\mu^1{}_0\mu^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu^1\mu^1{}_0A_1 & \mu^1\mu^1{}_0\mu^1 & \mu^1\mu^1{}_1A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_2\mu^2{}_0A_2 & A_2\mu^2{}_0\mu^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu^2\mu^2{}_0A_2 & \mu^2\mu^2{}_0\mu^2 & \mu^2\mu^2{}_2A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^2\mu^2{}_1A_7 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu^2\mu^2{}_0A_2 & 0 & A_3\mu^3{}_0A_3 & A_3\mu^3{}_0\mu^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu^3\mu^3{}_0A_3 & \mu^3\mu^3{}_0\mu^3 & \mu^3\mu^3{}_1A_4 & 0 & 0 & 0 & \mu^3\mu^3{}_2A_6 & 0 & 0 & 0 & 0 & 0
\end{array} \right\|$$

$$
\left\|\begin{array}{cccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4\mu^4{}_0A_4 & A_4\mu^4{}_0\mu^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^4\mu^4{}_0A_4 & \mu^4\mu^4{}_0\mu^4 & \mu^4\mu^4{}_1A_5 & 0 & 0 & 0 & \mu^4\mu^4{}_2A_7 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_5\mu^5{}_0A_5 & A_5\mu^5{}_0\mu^5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^5\mu^5{}_0A_4 & 0 & \mu^5\mu^5{}_0A_5 & \mu^5\mu^5{}_0\mu^5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_6\mu^6{}_0A_6 & A_{6.}\mu^6{}_0\mu^6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^6\mu^6{}_0A_6 & \mu^6\mu^6{}_0\mu^6 & 0 & 0 & \mu^6\mu^6{}_1A_{\text{к}} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_7\mu^7{}_0A_6 & A_7\mu^7{}_0\mu^7 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^7\mu^7{}_0A_7 & \mu^7\mu^7{}_0\mu^7 & \mu^7\mu^7{}_0A_{\text{к.}} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & A_{\text{к.}}\mu^{\text{к}}{}_0A_{\text{к}} & A_{\text{к}}\mu^{\text{к}}{}_0\mu^{\text{к}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^{\text{к}}\mu^{\text{к}}{}_0A_{\text{к}} & \mu^{\text{к}}\mu^{\text{к}}{}_0\mu^{\text{к}}
\end{array}\right\|
$$

Рис. 11. Модульное представление алгоритма в матрично-предикатном виде

$$
\mathbf{M}^{\text{A}*}=\left\|\begin{array}{ccccccccc|ccccccccc}
A_0\mu^0{}_0A_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_0\mu^0{}_0\mu^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_1\mu^1{}_0A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_1\mu^1{}_0\mu^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A_2\mu^2{}_0A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2\mu^2{}_0\mu^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & A_3\mu^3{}_0A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_3\mu^3{}_0\mu^3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_4\mu^4{}_0A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4\mu^4{}_0\mu^4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_5\mu^5{}_0A_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_5\mu^5{}_0\mu^5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & A_6\mu^6{}_0A_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{6.}\mu^6{}_0\mu^6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & A_7\mu^7{}_0A_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{7.}\mu^7{}_0\mu^7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{\text{к.}}\mu^{\text{к}}{}_0A_{\text{к}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{\text{к.}}\mu^{\text{к}}{}_0\mu^{\text{к}} \\
\hline
\mu^0\mu^0{}_0A_0 & \mu^0\mu^0{}_1A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^0\mu^0{}_0\mu^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mu^1\mu^1{}_0A_1 & \mu^1\mu^1{}_1A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^1\mu^1{}_0\mu^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu^2\mu^2{}_0A_2 & \mu^2\mu^2{}_2A_3 & 0 & 0 & 0 & \mu^2\mu^2{}_1A_7 & 0 & 0 & 0 & \mu^2\mu^2{}_0\mu^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mu^3\mu^3{}_0A_3 & \mu^3\mu^3{}_1A_4 & & \mu^3\mu^3{}_2A_6 & 0 & 0 & 0 & 0 & 0 & \mu^3\mu^3{}_0\mu^3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu^4\mu^4{}_0A_4 & \mu^4\mu^4{}_1A_5 & 0 & \mu^4\mu^4{}_2A_7 & 0 & 0 & 0 & 0 & 0 & \mu^4\mu^4{}_0\mu^4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu^5\mu^5{}_0A_4 & \mu^5\mu^5{}_0A_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^5\mu^5{}_0\mu^5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu^6\mu^6{}_0A_6 & 0 & \mu^6\mu^6{}_1A_{\text{к}} & 0 & 0 & 0 & 0 & 0 & 0 & \mu^6\mu^6{}_0\mu^6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^7\mu^7{}_0A_7 & \mu^7\mu^7{}_0A_{\text{к}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^7\mu^7{}_0\mu^7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^{\text{к}}\mu^{\text{к}}{}_0A_{\text{к}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu^{\text{к}}\mu^{\text{к}}{}_0\mu^{\text{к}}
\end{array}\right\|
$$

Рис. 12. Функционально-предикативное представление алгоритма в матрично-предикатном виде

Показанные в данной работе два вида представления алгоритма, а именно модульный и функционально-предикативный, имеют одинаково важное значение. Использовать для серьезной многокомпонентной задачи или же для работы со сложными системами один из них, исключая другой, было бы нерационально. Лучшим вариантом является сочетание обоих методов по ходу работы, используя на каждом этапе преимущества того или иного метода.

Работать с алгоритмами, представленными в таком виде, значительно легче, так как появляется возможность частично автоматизировать эвристические методы их построения.

Литература

1. *Колмогоров. А.Н.* Теория информации и теория алгоритмов.— М.: Наука, 1987.-304 с.
2. *Альфред В. Ахо, Джон Е. Хопкрофт, Джеффри Д. Ульман,* Структуры данных и алгоритмы: М.: Издательский дом "Вильяме", 2000. — 384 с.
3. *Янов Ю.И.* О логических схемах алгоритмов.// Проблемы кибернетики. М.: 1958
4. *Паронджанов В. Д.* Учись писать, читать и понимать алгоритмы. Алгоритмы для правильного мышления. Основы алгоритмизации. — М.: ДМК Пресс, 2012. — 520 с.
5. *Дж. Макконелл,* Основы современных алгоритмов, М.: «Техносфера», 2004, С. 10-11
6. *Гудман С., Хидетниеми С.* Введение в разработку и анализ алгоритмов. – М.: Мир, 1981. 368 с.
7. *Поляков В.С., Поляков С.В., Федченков П.В.* Построение формального описания технологического процесса в матрично-предикатной форме // Известия ВолгГТУ. Серия «Прогрессивные технологии в

машинострое-нии». Вып. 9: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2013. - № 7 (110). - С. 105-108.

8. *Поляков В.С., Поляков С. В*. Запись алгоритма матрицей инцидентора // Инновации на основе информационных и коммуникационных технологий. Инфо 2014: матер. XI междунар.научн.-практ. Конф. (г. Сочи, 1–10 окт. 2014) /Национальный исследовательский ун-т «Высшая школа экономики» [и др.]. – М., 2014. – с. 149-152.
9. *Поляков В.С., Поляков С.В.* Использование нагруженных матриц инцидентора (операторов) для моделирования сложных систем / // Контроль. Диагностика. - 2013. - № 3. - С. 57-62.