KYBERNETIKA ČÍSLO 3, ROČNÍK 4/1968

Об алгорифмах с элементами памяти

Зденек Заставка

В статье определяется алгорифм с элементами памяти, заданный граф-схемой (это некоторое соответствие Л. А. Калужнина [1]). Доказывается нормализуемость такого алгорифма (в смысле определения нормального алгорифма А. А. Маркова [2]).

ВВЕДЕНИЕ

Непосредственным импульсом для написания настоящей статьи является работа Л. А. Калужнина [1]. После обоснования непригодности существующих до сих пор „классических" теорией алгорифмов, а именно теории нормальных алгорифмов А. А. Маркова [2], для задач программирования для современных вычислительных машин,* Калужнин предлагает новый вариант алгорифма — графические алгорифмы, граф-эхемы. Потом, кроме этого, автор обращает внимание на следующий факт: „Как самым рациональным образом ввести элементы памяти в граф-схему, причем так, чтобы сохранилась возможность и смысл операций подстановок, является важным, но пока не решенным вопросом" ([2]; стр. 64).

В статье предлагается ввести памяти в такие граф-схемы, которые после интерпретации узлов можно определить как алгорифм композиции данных алгорифмов. Граф-схема названа *системиой граф-схемой*, алгорифм — *композиционным алгорифмом*. Таким образом, принятая граф-схема будет отличаться от граф-схемы Калужнина; соответствующие узлы будут интерпретироваться (целыми) алгорифмами, Калужнин их интерпретирует (отдельными) операциями алгорифма (напр. у нормальных алгорифмов левыми частями формул подстановок и целыми этими формулами схемы, определяющей данный нормальный алгорифм).

Основные понятия определены в части I (глава 1 и 2). В теоремах 1—4 показано, что к каждой композиции алгорифмов существует композиционный алгорифм, который возникает путем такой интерпретацуией соответствующих узлов системной граф-схемы, что достаточно воспользоваться только компонированными алгорифмами (не надо нам прибавлять дальнейшие алгорифмы, как это наблюдается например при нормальных композициях), а также, что нам не надо расширять алфавит, в котором даны эти алгорифмы. Мы получим таким образом элементарные виды композиционных алгорифмов.

* Калужнин главным образом интересуется стандартными языками для алгорифмизации задач (именно математических и логических задач), которые пригодны к переводу на собственный язык конкретно данной вычислительной машины.

В части II проведено доказательство эквивалентности композиционных алгорифмов, определяемых только нормальными алгорифмами (т. н. квазинормальных композиционных алгорифмов) и нормальных алгорифмов.

Общее понятие алгорифма не более подробно определено; я ссылаюсь на цитированную работу Калужнина [1] и монографию А. А. Маркова [2], которая была исходным пунктом для моей работы. (Теория нормальных алгорифмов А. А. Маркова была началом и для Калужнина.) Поэтому все употребленные понятия теории алгорифмов, которые не будут подробнее опрецелены, взяты из А. А. Маркова [2].

Я хотел бы выразить свою благодарность всем членам семинара на философском факультете Карлова университета во главе с д-ром философских наук проф. Отакаром Зихом за то, что я мог с ними консультировать вопросы касающиеся значения и проведения в жизнь этой работы. За очень конкретные замечания я благодарю профессора д-ра Иосифа Метелку и кандидата философских наук д-ра Павла Тихого, которые подробно подчитали первоначальный вариант этой работы. За обсуждение работы и важные замечания я признателен тоже Ивану Гавелу, пром. мат.

I. КОМПОЗИЦИОННЫЕ АЛГОРИФМЫ

1. Системные граф-схемы

Пусть дана некоторая плоскость и в ней

а) множество *окружностей* обозначенных через $O_1, \ldots, O_m$ ($m > 0$), лежащих вне себя и каждая вне другой; это множество обозначим через $\mathbf{O}$ и назовем его множеством *операторов* $O_1, \ldots, O_m$ (следователно $\mathbf{O} = \{O_1, \ldots, O_m\}$);

б) множество *овалов* обозначенных через $I_1, \ldots, I_n$ ($n \geqq 0$), лежащих вне себя и вне элементов множества $\mathbf{O}$ и каждый из них лежит вне другого и вне каждого элемента множества $\mathbf{O}$; это множество обозначим через $\mathbf{I}$ и назовем его множеством *распознавателей* $I_1, \ldots, I_n$ (следовательно $\mathbf{I} = \{I_1, \ldots, I_n\}$);

в) множество *прямоугелников* обозначенных через $P_1, \ldots, P_p$ ($p \geqq 0$), лежащих вне себя и вне элементов множеств $\mathbf{O}$ и $\mathbf{I}$ и каждый из них лежит вне другого и вне элементов множеств $\mathbf{O}$ и $\mathbf{I}$; это множество обозначим через $\mathbf{P}$ и назовем его множеством *памятей* $P_1, \ldots, P_p$ (следовательно $\mathbf{P} = \{P_1, \ldots, P_p\}$);*

г) *точка* обозначена через V, которая не является элементом ни одного элемента множеств $\mathbf{O}$, $\mathbf{I}$ и $\mathbf{P}$ и находится вне этих элементов; эта точка — или множество состоящее из одного элемента, которое обозначим через $\mathbf{V}$ (следовательно $\mathbf{V} = \{V\}$) — названа *входом*;

д) *точка* обозначена через Vy, различна от входа и не является элементом ни одного элемента множеств $\mathbf{O}$, $\mathbf{I}$, и $\mathbf{P}$ и находится вне этих элементов; эта точка — или множество состоящее из одного элемента, которое обозначим через $\mathbf{Vy}$ (следовательно $\mathbf{Vy} = \{Vy\}$) — названа *выходом*.

* Для $m = 1$ или $n = 1$ или $p = 1$ будем пользоваться только обозначениями O, I, P; следовательно $\mathbf{O} = \{O\}$, $\mathbf{I} = \{I\}$, $\mathbf{P} = \{P\}$.

Совокупность (систему) конечных не пересекающихся множеств **V**, **O**, **I**, **P**, **Vy** обозначим через **M**:

$$\mathbf{M} = \{\mathbf{V}, \mathbf{O}, \mathbf{I}, \mathbf{P}, \mathbf{Vy}\}\ ;$$

объединение системы множеств **M** обозначим через **sM**:

$$\mathbf{sM} = \mathbf{V} \cup \mathbf{O} \cup \mathbf{I} \cup \mathbf{P} \cup \mathbf{Vy}\,.$$

Пусть m_1 и m_2 – два (равных или не равных) элемента объединения **sM**; тогда скажем, что m_1 *соединен направленным отрезком* (*стрелкой*) с m_2 (или только: m_1 соединен с m_2), когда стрелка исходит из m_1 и кончается в m_2 или наоборот. (Графически это можно записать следующим образом: $\overrightarrow{m_1 m_2}$ или $\overleftarrow{m_1 m_2}$.)

У каждой стрелки *начало* (точка элемента объединения **sM**, из которой стрелка исходит) и *конец* стрелки (точка элемента объединения **sM**, в котором стрелка кончается).

Соединение которого-нибуть элемента объединения **sM** с которой-нибудь памятью, будет соединением этого элемента с какой-нибудь стороной данного прямоугельника – скажем, что это соединение является соединением этого элемента с какой-нибудь *стороной* (*данной*) *памяти*. (Стрелка из какой-нибудь памяти исходит или в ней кончается.) Если соединены все элементы объединения **sM** так, что исходя из какого-нибудь элемента, можно достигнуть любой другой, следуя по отрезкам (не обязательно всегда в направлении стрелки), то эту плоскую фигуру будем называть *системной граф-схемой* (с. г. с.), если соединение элементов соответствует следующим правилам:*

1. Во *входе* не кончается ни одна стрелка и исходит из него одна и только одна стрелка.

2. Из *выхода* не исходит ни одна стрелка.

3. В каждом *операторе* кончается хотя бы одна стрелка и исходит из него одна и только одна стрелка.

4. В каждом *распознавателе* кончается одна и только одна стрелка и исходят из него две и только две стрелки, отмеченные соответственно знаками + и – (*плюс-стрелка* и *минус-стрелка*), которые кончаются в какой-нибудь памяти.**

5.1. Каждую сторону памяти в которой кончается хотя бы одна стрелка, будем называть *входом памяти*. Концы всех кончающихся стрелок в каждом входе памяти *сливаются* (это значит, что все стрелки кончаются во входе памяти в одной и только в одной точке; см. рис. 1).

* Ср. [1], § 3, 1.

** Плюс- и минус-стрелка кончается каждая в одинаковой или различной стороне одной и той же памяти; из характера распознавателей следует, что это большой частью различные стороны.

В каждой памяти всегда хотя бы один вход памяти.

5.2. Каждую сторону памяти из которой исходит хотя бы одна стрелка, будем называть *выходом памяти*; выходом памяти может быть и вход памяти и наоборот. Начала всех исходящих стрелок в выходе памяти *не сливаются* (это значит, что стрелки исходят из выхода памяти из различных точек). Если исходит из выхода памяти n (>1) стрелок, то стрелки нумерованы номерами $1, 2, \ldots, n$ (*нумерованные стрелки**; см. рис. 2).

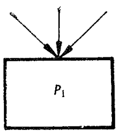

Рис. 1.

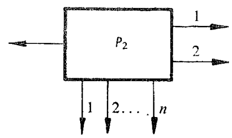

Рис. 2.

5.3. *Противоположная сторона входа памяти есть всегда выходом памяти и наоборот.* (Это значит, что сколько в памяти входов памяти, столько в ней и выходов памяти. В каждой памяти по крайней мере один вход памяти и один выход памяти и не больше четырех входов памяти и четырех выходов памяти.) В никакой стороне памяти не должно сливаться начало исходящей стрелки с концом никакой окончательной стрелки.

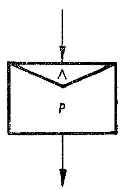

Рис. 3.

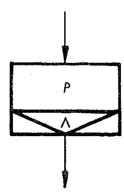

Рис. 4.

Вход памяти со специальным обозначением (рис. 3) мы будем называть *нон-входом памяти*; выход памяти со специальны обозначением (рис. 4) мы будем называть *нон-выходом памяти.*

В правилах $1-5.3$ мы можем повсюду вместо входа памяти обдумывать нон-вход памяти и вместо выхода памяти нон-выход памяти.**

* Множество ($n > 1$) исходящих стрелок из каждого выхода памяти взаимно однозначно отображено на конечное множество $\{1, 2, \ldots, n\}$ (единственную исходящую стрелку номером 1 не нумеруем!); это множество таким образом всегда упорядоченное множество.

** Мы *не говорим*, что если заменим в 5.1—5.3 „вход памяти" „нон-входом памяти", то мы должны заменить и „выход памяти" „нон-выходом памяти". Противоположной стороной входа памяти *может быть* нон-выход памяти, нон-входа памяти выход памяти или нон-выход памяти.

Таким образом соединенные элементы объединения $\mathbf{sM}$ мы будем называть *узлами* данной с. г. с.

2. Композиционный алгорифм

а) Дана с. г. с. с системой $\mathbf{M}$, где $\mathbf{O} = \{O_1, \ldots, O_m\}$ $(m > 0)$ и $\mathbf{I} = \{I_1, \ldots, I_n\}$ $(n \geqq 0)$.

б) Дан некоторый алфавит А.

в) Каждому узлу $O_i \in \mathbf{O}$, где $1 \leqq i \leqq m$, (каждому оператору) соответствует алгорифм $\mathfrak{O}_i$ в алфавите А (соответствие не нужно быть взаимно однозначное).

г) Каждому узлу $I_j \in \mathbf{I}$, где $0 \leqq j \leqq n$, (каждому распознавателю) соответствует алгорифм $\mathfrak{I}_j$ в алфавите А (соответствие не нужно быть взаимно однозначное).

Условия а)—г) определяют *алгорифмическую интерпретацию* данной с. г. с., которую будем обозначать

$$\langle \mathrm{A}; \mathbf{O} \to \mathfrak{O}; \mathbf{I} \to \mathfrak{I} \rangle\,,$$

где $\mathfrak{O} = \{\mathfrak{O}_1, \ldots, \mathfrak{O}_m\}$ и $\mathfrak{I} = \{\mathfrak{I}_1, \ldots, \mathfrak{I}_n\}$.*

Пусть дана некоторая алгорифмическая интерпретация

$$\langle \mathrm{A}; \mathbf{O} \to \mathfrak{O}; \mathbf{I} \to \mathfrak{I} \rangle\,.$$

Тогда всякую с. г. с. можно рассматривать как однозначное предписание для выполнения некоторого действия, в результате которого некоторые слова перейдут в новые слова в А. Как это действие мы осуществляем дано в следующем:

Пусть дана алгорифмическая интерпретация с. г. с. с системой $\mathbf{M}$, где $\mathbf{P} = \{P_1, \ldots, P_p\}$ для $p > 0$. Всегда, когда слово S в алфавите А поступает в узел $P_k \in \mathbf{P}$, где $1 \leqq k \leqq p$, этому узлу соответствует слово O, которое называется *содержанием памяти* P_k. Будем такое соответствие обозначать $P_k : O :$.

Сформулируем теперь *правила образования содержания памятей:*

аа) Если в память P_k поступает слово S в алфавите А каким-нибудь из ее входов, то

(1) если это слово является *первым словом* поступающим в данную память, то ее содержанием будет слово S:

$$P_k : S : .$$

* Для $m = 1$ или $n = 1$ будем пользоваться только обозначениями $\mathfrak{O}, \mathfrak{I}$; следовательно $\mathfrak{O} = \{\mathfrak{O}\}, \mathfrak{I} = \{\mathfrak{I}\}$.

(2) если перед поступлением этого слова было содержанием данной памяти слово O, то теперь содержанием памяти (после этого поступления слова S) является соединение слов O и S:

$$P_k : OS : .$$

(Если $S \neq \Lambda$, то можно сказать, что *содержание памяти расширилось*; если $S = \Lambda$, то *содержание памяти не тасширилось*.)

бб) Если в память P_k поступает слово S в алфавите А каким-нибудь из ее нон-входов, то

(3) если это слово является *первым словом* поступающим в данную память, то ее содержанием будет *пустое слово*:

$$P_k : \Lambda : .$$

(4) если это слово не является первым словом поступающим в данную память, то (существующее до сих пор) *содержание памяти не изменяется*. (Если было перед этим поступлением содержанием памяти слово O, то соответствие $P_k : O :$ не изменяется.)

вв) Если из памяти P_k выходит слово (и пустое слово) по *последней стрелке* (по стрелке с наибольшим номером) какого-нибудь выхода памяти или нон-выхода памяти, содержанием памяти станет *путсое слово* (содержание памяти аннулировано):

$$P_k : \Lambda : .$$

После определения алгорифмической интерпретации $\langle \mathrm{A};\ \mathbf{O} \to \mathfrak{O};\ \mathbf{I} \to \mathfrak{I} \rangle$ данной с. г. с., использованием правил образования содержания памятей нам можно сформулировать правила преобразования поступающих слов в отдельные узлы с определением направления выхода (из этих узлов) преобразованных слов — *правила ориентированного преобразования слов*:

ааа) Если слово S в альфавите А поступает в узел V (на вход) — условимся это обозначать символом $V : S :$ — то из этого узла продвигается по направлению выходящей стрелки в другой узел опять слово S.*

ббб) Если слово S в алфавите А поступает в узел $O_i \in \mathbf{O}$ (в оператор) и $\mathfrak{O}_i(S) = T$, где $\mathfrak{O}_i \in \mathfrak{O}$, то из этого узла продвигается по направлению выходящей стрелки к какой-нибудь (другой или тот же) узел слово T; если не существует слово T, то из узла O_i не выходит ни одно слово.

ввв) Если слово S в алфавите А поступает в узел $I_j \in \mathbf{I}$ (в распознаватель) и $\mathfrak{I}_j(S) = U$, где $\mathfrak{I}_j \in \mathfrak{I}$, то из этого узла продвигается в какой-нибудь (другой

* В каждой с. г. с. всегда хотя бы один оператор (см. а)), так что стрелка из входа никогда не кончается в выходе.

или тот же) узел слово U

– по направлению выходящей плюс-стрелки, когда $U = \Lambda$,

– по направлению выходящей минус-стрелки, когда $U \neq \Lambda$;

если не существует слово U, то из узла I_j не выходит ни одно слово.

ггг) Пусть после поступления слова в алфавите А в узел $P_k \in \mathbf{P}$ какой-нибудь его стороной

$$P_k : O : .$$

Тогда из этого узла продвигается по направлению выходящей стрелки из *противоположной стороны к данной стороне* с входом памяти или нон-входом памяти в какой-нибудь (другой или тот же) узел слово S, где

(1) $S = O$, если эта противоположная сторона *выход памяти*,

(2) $S = \Lambda$, если эта противоположная сторона *нон-выход памяти.*

Если в этом выходе памяти или нон-выходе памяти нумерованные стрелки (стрелки с номерами $1, \ldots, n$; $n > 1$), то слово S продвигается по направлению выходящей стрелки с номером $i + 1$, где

(3) $i = 0$, если слово S является *первым словом* выходящим *этой стороной* или если *последним выходом* (перед выходом слова S) из этой стороны был выход какого-нибудь слова по *последней стрелке* (по стрелке с номером n);

(4) i – номер стрелки, по которой выходило *последнее слово* (перед выходом слова S) из этой стороны, о котором знаем, что*

$$1 \leqq i < n .$$

ддд) Если слово S в алфавите А поступило в узел Vy (на выход) – условимся это обозначать символом $Vy : S :$ – то из этого узла не выходит ни одно слово.

Алгорифмическая интерпретация

$$\langle \mathrm{A};\ \mathbf{O} \to \mathfrak{O};\ \mathbf{I} \to \mathfrak{I} \rangle$$

и правила образования содержания памятей с правилами ориентированного преобразования слов (в данной с. г. с. с определенной системой множеств $\mathbf{M}$) определяют какой-нибудь алгорифм в алфавите А, который мы будем называть *композиционным алгорифмом* (к. а.) $\mathfrak{K}$ в алфавите А. Слово S в алфавите А поступившее в узел V – *исходное слово* данного применения этого к. а.; если существует слово T в алфавите А поступившее в узел Vy, то мы будем его называть *результатом* данного применения этого к. а. к слову S:

$$\mathfrak{K}(S) = T .$$

* Выход слов выходом памяти или нон-выходом памяти с больше чем одной стрелкой (с нумерованными стрелками) осуществляется *постепенно* начиная первой стрелкой; после выхода последней стрелкой выходы слов повторяются начиная опять первой стрелкой.

Композиционные алгорифмы выгодны для композиции алгорифмов.* Покажем это на наиболее распространенных видах композиции алгорифмов: суперпозиции, объединении, повторении и разветвлении алгорифмов:

Теорема 1. *Каковы бы ни были алгорифмы* $\mathfrak{D}_1, \ldots, \mathfrak{D}_n$ $(n \geqq 2)$ *в алфавите* A, *может быть построен такой к. а.* $\mathfrak{K}$ *в алфавите* A *с алгорифмической интерпретацией* $\langle \text{A}; \mathbf{O} \to \mathfrak{D} \rangle$, *что*

$$\text{(1-1)} \qquad \mathfrak{K}(S) \simeq \mathfrak{D}_n(\mathfrak{D}_{n-1}(\ldots(\mathfrak{D}_1(S))) \quad (S - \textit{слово в } \text{A}),$$

$$\text{(1-2)} \qquad \mathbf{O} = \{O_1, \ldots, O_n\} \quad \text{и} \quad \mathfrak{D} = \{\mathfrak{D}_1, \ldots, \mathfrak{D}_n\}.$$

Каждый такой к. а. $\mathfrak{K}$ *мы будем называть к. а. суперпозиции алгорифмов* $\mathfrak{D}_1, \ldots, \mathfrak{D}_n$.**

Доказательство теоремы 1 проведется построением такой с. г. с. с множеством операторов $\mathbf{O} = \{O_1, \ldots, O_n\}$ $(n \geqq 0)$, что после алгорифмической интерпретации $\langle \text{A}; \mathbf{O} \to \mathfrak{D} \rangle$ (согласно условию (1-2)) получим к. а. $\mathfrak{K}$ в алфавите A удовлетворяющий (1-1), как о том убедимся его применением к исходному слову S в алфавите A.***

С. г. с. имеет вид, изображенный на рис. 5.

Запись применения к. а. $\mathfrak{K}$ к слову S проведен так, что кроме первой и последней строки с записанным поступлением слова S в узел V и поступлением результата применения этого алгорифма в узел Vy, на каждой строке записаны слова выходящие постепенно из отдельных узлов с. г. с.:

$V : S :$

$\mathfrak{D}_1(S)$

$\ldots\ldots$

$\mathfrak{D}_{n-1}(\ldots(\mathfrak{D}_1(S)))$

$\mathfrak{D}_n(\mathfrak{D}_{n-1}(\ldots(\mathfrak{D}_1(S)))$

$Vy : \mathfrak{D}_n(\mathfrak{D}_{n-1}(\ldots(\mathfrak{D}_1(S)))) : .$

* Наш термин „композиция“ не соответствует этому же термину А. А. Маркова в [2]; „композиция алгорифмов“ в нашем значении для Маркова „сочетание алгорифмов“; „композиция алгорифмов“ Маркова — „суперпозиция алгорифмов“ для нас.

** Эта и следующие теоремы формулированы согласно нормальным композоцоям А. А. Маркова в [2], IV.

*** Приведенная с. г. с. не имеет ни одной памяти; потому здесь не приходят во внимание правила образования содержания памятей.

Слово поступающее в узел V_Y, т. е. результат применения к. а. $\mathfrak{K}$ в алфавите А к исходному слову S, есть слово $\mathfrak{O}_n(\mathfrak{O}_{n-1}(\ldots(\mathfrak{O}_1(\in))))$. Этот результат хватит к подтверждению действительности условия (1-1):

$$\mathfrak{K}(S) \simeq \mathfrak{O}_n(\mathfrak{O}_{n-1}(\ldots(\mathfrak{O}_1(S)))) .$$

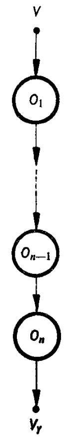

Рис. 5.

Замечание. Доказательство теоремы 1 приведено значительно сокращенным образом. Существенную часть совершенного доказательства помимо с. г. с. здесь замещает запись применения к. а. $\mathfrak{K}$ к исходному слову. Совершение доказательства зависело бы от приведения доказательства следующих лемм:

Лемма 1.1. Если имеет смысл выражение $\mathfrak{O}_n(\mathfrak{O}_{n-1}(\ldots(\mathfrak{O}_1(S))))$, то к. а. $\mathfrak{K}$ применим к слову S и

$$\mathfrak{K}(S) = \mathfrak{O}_n(\mathfrak{O}_{n-1}(\ldots(\mathfrak{O}_1(S)))) .$$

Лемма 1.2 Если к. а. $\mathfrak{K}$ применим к слову S в алфавите А, то имеет смысл выражение $\mathfrak{O}_n(\mathfrak{O}_{n-1}(\ldots(\mathfrak{O}_1(S)))$.

Так как речь идет о аналогии доказательств теорем о нормальных алгорифмах А. А. Маркова (см. [2], III) и так как наши теоремы и доказательства являются значительно более простые, мы не считаем необходимым и целесообразным (именно ввиду обширности этого доказательства) здесь целое доказательство выполнить.

Это аналогично действительно и для в дальшем приведенные теоремы и их доказательства.

Теорема 2. *Каковы бы ни были алгорифмы* $\mathfrak{O}_1, \ldots, \mathfrak{O}_n$ $(n \geqq 2)$ *в алфавите* А, *может быть построен такой к. а.* $\mathfrak{K}$ *в алфавите* А *с алгорифмической интер-*

 претацией $\langle \mathrm{A};\ \mathbf{O} \to \mathfrak{O} \rangle$, *что*

(2-1) $$\mathfrak{K}(S) \simeq \mathfrak{O}_1(S) \ldots \mathfrak{O}_n(S) \quad (S - \text{слово в } \mathrm{A}),$$

(2-2) $$\mathbf{O} = \{O_1, \ldots, O_n\} \quad и \quad \mathfrak{O} = \{\mathfrak{O}_1, \ldots, \mathfrak{O}_n\}.$$

Каждый такой к. а. $\mathfrak{K}$ *мы вудем называть к. а. объединения алгорифмов* $\mathfrak{O}_1, \ldots, \mathfrak{O}_n$.

С. г. с. для этого доказательства имеет кроме множества $\mathbf{O}$ и множество $\mathbf{P} = \{P_1, P_2\}$ (рис. 6).

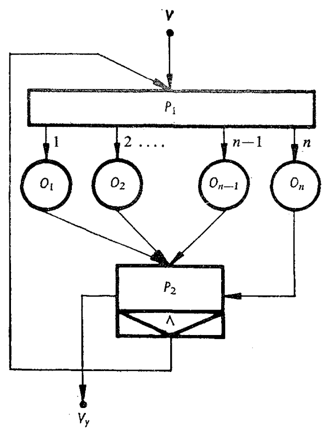

Рис. 6.

Запись применения этого к. а. $\mathfrak{K}$ к слову S имеет кроме строк с исходным словом, выходящими словами из операторов (узлов $O_1, \ldots, O_n$) и словом поступающим на выход, как этому было в доказательстве теоремы 1, также строки с содержанием отдельных памятей (узлов P_1 и P_2) и у которых обозначено стрелком (или нумерованной стрелкой) направление дальнейшего продвижения слов выходящих из этих узлов (согласно правилам образования содержания памятей и соответствующим правилам ориентированного преобразования слов); направление данное стрелкой исходящей из нон-выхода памяти имеет символ Λ (это предупреждение о выходящем из этого узла слове Λ); после выхода слова последней нумерованной стрелкой обозначено содержание памяти (это пустое слово – анулирование содержания памяти) в квадратных скобках:

$V : S :$

$P_1 : S : \downarrow 1$

$\mathfrak{D}_1(S)$
$P_2 : \mathfrak{D}_1(S) : \downarrow \Lambda$
$P_1 : S : \downarrow 2$
$\mathfrak{D}_2(S)$
$P_2 : \mathfrak{D}_1(S)\,\mathfrak{D}_2(S) : \downarrow \Lambda$
....
$P_2 : \mathfrak{D}_1(S)\,\mathfrak{D}_2(S) \ldots \mathfrak{D}_{n-1}(S) : \downarrow 1$
$P_1 : S : \downarrow n$
$[P_1 : \Lambda :]$
$\mathfrak{D}_n(S)$
$P_2 : \mathfrak{D}_1(S)\,\mathfrak{D}_2(S) \ldots \mathfrak{D}_{n-1}(S)\,\mathfrak{D}_n(S) : \leftarrow$
$Vy : \mathfrak{D}_1(S)\,\mathfrak{D}_2(S) \ldots \mathfrak{D}_{n-1}(S)\,\mathfrak{D}_n(S) : .$

Теорема 3. *Каковы бы ни были алгорифмы* $\mathfrak{D}$ *и* $\mathfrak{I}$ *в алфавите* A, *может выть построен такой к. а.* $\mathfrak{K}$ *в алфавите* A *с алгорифмической интерпретацией* $\langle$A; $\mathbf{O} \to \mathfrak{D}$; $\mathbf{I} \to \mathfrak{I}\rangle$, *что*

(3-1) $$\mathfrak{K}(S) = T \quad (S - \text{слово в } \mathrm{A}),$$

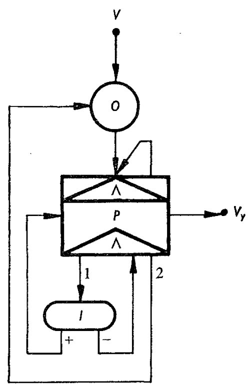

Рис. 7.

тогда и только тогда, когда существует ряд слов $S_0, S_1, \ldots, S_n$ $(n > 0)$ *удовлетборяющий условиям*

(i) $$S_0 = S\,,$$

(ii) $$S_i = \mathfrak{D}(S_{i-1}) \quad (0 < i \leqq n)\,,$$

(iii) $$S_n = T,$$

(iv) $$\mathfrak{J}(S_i) \neq \Lambda \quad (0 < i < n),$$

(v) $$\mathfrak{J}(S_n) = \Lambda,$$

(3-2) $$\mathbf{O} = \{O\}, \quad \mathbf{I} = \{I\} \quad \text{и} \quad \boldsymbol{\mathfrak{D}} = \{\mathfrak{D}\}, \quad \boldsymbol{\mathfrak{J}} = \{\mathfrak{J}\}.$$

Каждый такой к. а. $\mathfrak{K}$ *мы будем называть к. а. повторения алгорифма* $\mathfrak{D}$, *управляемое алгорифмом* $\mathfrak{J}$.

С. г. с. имеет вид, изображенный на рис. 7 ($\boldsymbol{\mathfrak{D}} = \{O\}$, $\mathbf{I} = \{I\}$, $\mathbf{P} = \{P\}$).

Запись применения к. а. $\mathfrak{K}$ к исходному слову S в алфавите А:

$$
\begin{array}{lll}
V : S : & & \\
\mathfrak{D}(S) & & \\
P : \mathfrak{D}(S) : \downarrow 1 & & \\
\mathfrak{J}(\mathfrak{D}(\in) \downarrow + \, * \quad \mathfrak{J}(\mathfrak{D}(S)) \downarrow - \, * & & \\
P : \mathfrak{D}(S) : \rightarrow ** \quad P : \mathfrak{D}(S) : \uparrow \Lambda \; ** & & \\
Vy : \mathfrak{D}(S) : & P : \mathfrak{D}(S) : \downarrow 2 & \\
 & [P : \Lambda :] & \\
 & \mathfrak{D}(\mathfrak{D}(S)) & \\
 & P : \mathfrak{D}(\mathfrak{D}(S)) : \downarrow 1 & \\
 & \mathfrak{J}(\mathfrak{D}(\mathfrak{D}(S))) \downarrow + & \mathfrak{J}(\mathfrak{D}(\mathfrak{D}(S))) \downarrow - \\
 & P : \mathfrak{D}(\mathfrak{D}(S)) : \rightarrow & P : \mathfrak{D}(\mathfrak{D}(S)) : \uparrow \Lambda \\
 & Vy : \mathfrak{D}(\mathfrak{D}(S)) : & P : \mathfrak{D}(\mathfrak{D}(S)) : \downarrow 2 \\
 & & [P : \Lambda :] \\
 & & \ldots\ldots\ldots\ldots\ldots \\
 & & Vy : \mathfrak{D}(\mathfrak{D}(\ldots(\mathfrak{D}(S)))) : .
\end{array}
$$

Теорема 4. *Каковы бы ни были алгорифмы* $\mathfrak{D}_1, \mathfrak{D}_2, \mathfrak{J}$ *в алфавите* А, *может быть построен такой к. а.* $\mathfrak{K}$ *в алфавите* А *с алгорифмической интерпретацией* $\langle$A; $\mathfrak{D} \to \boldsymbol{\mathfrak{D}}$; $I \to \boldsymbol{\mathfrak{J}}\rangle$, *что*

(4-1.1) $$\mathfrak{K}(S) \simeq \mathfrak{D}_1(S) \quad (S - \text{слово в А}, \ \mathfrak{J}(S) = \Lambda),$$

(4-1.2) $$\mathfrak{K}(S) \simeq \mathfrak{D}_2(S) \quad (S - \text{слово в А}, \ \mathfrak{J}(S) \neq \Lambda),$$

(4-2) $$\mathbf{O} = \{O_1, O_2\}, \quad \mathbf{I} = \{I\} \quad \text{и} \quad \boldsymbol{\mathfrak{D}} = \{\mathfrak{D}_1, \mathfrak{D}_2\}, \quad \boldsymbol{\mathfrak{J}} = \{\mathfrak{J}\}$$

* Таким образом обозначен выход слова $\mathfrak{J}_j(S')$ (S' — слово поступающее в узел I_j; здесь $I_j = I$, $S' = \mathfrak{D}(S)$) плюс-стрелкой и минус-стрелкой.

** Согласно правилам ориентированного преобразования слов известно, что из плюс-стрелки выходит всегда слово Λ.

и что $\mathfrak{K}$ применим к тем и только к тем словам в алфавите А, *к которым применим $\mathfrak{I}$. Каждый такой к. а. $\mathfrak{K}$ мы вудем называть к. а. разветвления алгорифмов $\mathfrak{O}_1$, и $\mathfrak{O}_2$, иуправляемое алгорифмом $\mathfrak{I}$.*

С. г. с. имеет вид, изображенный на рис. 8. ($\boldsymbol{I} = \{I\}$, $\boldsymbol{P} = \{P\}$).

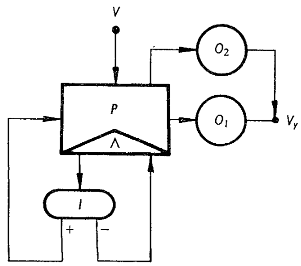

Рис. 8.

Запись применения к. а. $\mathfrak{K}$ к исходному слову S в алфавите А:

$V : S :$	
$P : S : \downarrow$	
$\mathfrak{I}(S) \downarrow +$	$\mathfrak{I}(S) \downarrow -$
$P : S : \rightarrow$	$P : S : \uparrow$
$\mathfrak{O}_1(S)$	$\mathfrak{O}_2(S)$
$Vy : \mathfrak{O}_1(S) :$	$Vy : \mathfrak{O}_2(S) : .$

II. КВАЗИНОРМАЛЬНЫЕ КОМПОЗИЦИОННЫЕ АЛГОРИФМЫ И НОРМАЛЬНЫЕ АЛГОРИФМЫ

1. Квазинормальные композиционные алгорифмы

Алгорифмическую интерпретацию

$$\langle \text{А};\ \boldsymbol{O} \rightarrow \mathfrak{O};\ \boldsymbol{I} \rightarrow \mathfrak{I} \rangle$$

некоторой с. г. с. мы будем называть *нормальной алгорифмической интерпретацией*, если алгорифмы $\mathfrak{O}_i$ ($\in \mathfrak{O}$) и $\mathfrak{I}_j$ ($\in \mathfrak{I}$) нормальные алгорифмы в алфавите А.

Каждый к. а. заданный нормальной алгорифмической интерпретацией мы будем называть *квазинормальным копмозиционным алгорифмом* (к. к. а.).

Ясно, что имеет место

Теорема 5. *Для любого нормального алгорифма* $\mathfrak{K}$ *в алфавите* А *существует такой к. к. а.* $\mathfrak{K}$ *в альфавите* А, *что*

$$\mathfrak{A}(S) \simeq \mathfrak{K}(S) \quad (S - \textit{слово в } \mathrm{A}) .$$

Докажем, что имеет место

Теорема 6. *Для любого к. к. а.* $\mathfrak{K}$ *в алфавите* А *существует такой нормальный алгдрифм* $\mathfrak{A}$ *над алфавитом* А, *что*

$$\mathfrak{K}(S) \simeq \mathfrak{A}(S) \quad (S - \textit{слово в } \mathrm{A}) .$$

Доразательство этой теоремы будем вести подобным способом, как в работе Н. П, Тер-Захаряна [4] доказывается эквивалентность квазинормальных и нормальных алгорифмов (где квазинормальными алгорифмами названы алгорифмы получаемые нормальной интерпретацией граф-схем в смысле Л. А. Калужнина). Пусть

$$\langle \mathrm{A};\ \mathbf{O} \to \mathfrak{O};\ \mathbf{I} \to \mathfrak{I} \rangle$$

нормальна алгорифмическая интерпретация данной с. г. с. на которой был определен к. к. а. $\mathfrak{K}$ в алфавите А.

Концы окончательных стрелок в входах памяти или в нон-входах всех памятей (узлов $P_1, \ldots, P_p$) мы будем называть квази-узлами (*к-узлами*) данной с. г. с. Каждая память (узел $P_k \in \mathbf{P}$, для $1 \leqq k \leqq p$) так с одной до четырех к-узлов.

Противоположностью входа или оператора (узла V или узла $O_i \in \mathbf{O}$, где $1 \leqq i \leqq m$) мы будем называть узел (если речь идет о операторе, распознавателе или выходе) или к-узел (если речь идет о к-узле какой-то памяти), к которому подходит стрелка из входа или данного оператора.

Плюс и *минус-противоположностью* распознавателя мы будем называть узел (речь идет о операторе, другом распознавателе или выходе) к которому подходит плюс и минус-стрелка из данного распознавателя.

Противоположностью или *нон-противоположностью* к-узла данного входа памяти или нон-входа памяти мы будем называть узел (если речь идет о операторе, распознавателе или выходе) или к-узел (если речь идет о к-узле некоторой памяти), к которому подходит стрелка (не нумерованная стрелка) из противоположного выхода памяти или нон-выхода памяти.

Упорядоченными противоположностями или *упорядоченными нон-противоположностями* к-узла данного входа памяти или нон-входа памяти мы будем называть все узлы (если речь идет о операторах, распознавателях или выходе) или к-узлы (если речь идет о к-узлах некоторых памятей), к которым подходят нумерованные стрелки из *противоположного выхода памяти или нон-выхода памяти*; если стрелки имеют номера $1, \ldots, n$, потом мы скажем, что к-узел имеет *первую по n-той противоположности или нон-противоположности.*

Знаем, что m и n числа операторов и распознавателей; пусть r — число к-узлов и $s = r + m + n$. Если выходу поставим в соответствие номер $s + 1$ и каждому оператору, распознавателю и к-узлу всегда только один из номеров $1, \ldots, s$, то

— множество номеров поставленных в соответствие операторам (элементам множества **O**) мы будем обозначать через **On**;

— множество номеров поставленных в соответствие распознавателям (элементам множества ***I***) мы будем обозначать через **In**;

— множество номеров поставленных в соответствие к-узлам в входах памятей с противоположностью мы будем обозначать через Pn;

— множество номеров поставленных в соответствие к-узлам в нон-входах памятей с противоположностью мы будем обозначать через $\mathsf{Pn}^{\wedge}$;

— множество номеров поставленных в соответствие к-узлам в входах памятей с нон-противоположностью мы будем обозначать через $\mathsf{Pn}_{\wedge}$;

— множество номеров поставленных в соответствие к-узлам в нон-входах памятей с нон-противоположностью мы будем обозначать через $\mathsf{Pn}^{\wedge}_{\wedge}$;

— множество номеров поставленных в соответствие к-узлам в входах памятей с упорядоченными противоположностями мы будем обозначать через Pno;

— множество номеров поставленных в соответствие к-узлам в нон-входах памятей с упорядоченными противоположностями мы будем обозначать через $\mathsf{Pno}^{\wedge}$;

— множество номеров поставленных в соответствие к-узлам в входах памятей с упорядоченными нон-противоположностями мы будем обозначать через $\mathsf{Pno}_{\wedge}$;

— множество номеров поставленных в соответствие к-узлам в нон-входах памятей с упорядоченными нон-противоположностями мы будем обозначать через $\mathsf{Pno}^{\wedge}_{\wedge}$.

Соединение множеств Pn, $\mathsf{Pn}^{\wedge}$, $\mathsf{Pn}_{\wedge}$ и $\mathsf{Pn}^{\wedge}_{\wedge}$ мы будем обозначать через $\mathbf{Pn}$:

$$\mathbf{Pn} = \mathsf{Pn} \cup \mathsf{Pn}^{\wedge} \cup \mathsf{Pn}_{\wedge} \cup \mathsf{Pn}^{\wedge}_{\wedge}\ ;$$

аналогично

$$\mathbf{Pno} = \mathsf{Pno} \cup \mathsf{Pno}^{\wedge} \cup \mathsf{Pno}_{\wedge} \cup \mathsf{Pno}^{\wedge}_{\wedge}\ .$$

Для $1 \leqq i \leqq s$ пусть $\Theta(i)$ — число противоположностей или нон-противоположностей или упорядоченных противоположностей или упорядоченных нон-противоположностей или плюс или минус-противоположностей узла или к-узла с поставленным в соответствие номером i; ясно, что $\Theta(i) = 1$, если $i \in \mathbf{On} \cup \mathbf{Pn}$; $\Theta(i) = 2$, если $i \in \mathbf{In}$; $\Theta(i) > 1$, если $i \in \mathbf{Pno}$.

 Определим функцию $\Psi(i, j)$, для $1 \leqq i \leqq s$, $1 \leqq j \leqq \Theta(i)$:

$$\Psi(i,j) = \begin{cases} \text{номер противоположности или нон-противоположности узла} \\ \quad \text{или к-узла с номером } i, \text{ если } i \in \mathbf{On} \cup \mathbf{Pn} \text{ (всегда } j = 1); \\ \text{номер плюс -противоположности узла с номером } i, \text{ если} \\ \quad i \in \mathbf{In} \text{ и } j = 1; \\ \text{номер минус-противоположности узла с номером } i, \text{ если} \\ \quad i \in \mathbf{In} \text{ и } j = 2; \\ \text{номер } j\text{-ой противоположности или нон-противоположности} \\ \quad \text{к-узла с номером } i, \text{ если } i \in \mathbf{Pno}. \end{cases}$$

Искомый нормальный алгорифм $\mathfrak{A}$ над алфавитом А построим как суперпозицию трех алгорифмов:

$$\mathfrak{A} = \mathfrak{D} \circ \mathfrak{C} \circ \mathfrak{B}\,,$$

где $\mathfrak{B}$ будет такой алгорифм, который к входному слову S „присоединит" со специальным обозначением эти основные информации о с. г. с. (о нумерации узлов и их соедении):

– во первых здесь будут введены (в специальном устроении) элементы множества **Pno**;

– далее потом алгорифмы, которые будут „выражать" способ трансформации слов в отдельных узлах и к-узлах (это будет по существу устроенная алгорифмическая переписка правил 1 – 5.3);

– наконец здесь будет отметкой „$\triangle$" выражено до которого узла поступающее слово поступает.

Алгорифм $\mathfrak{C}$ будет „моделировать" настоящее применение к. к. а. $\mathfrak{K}$ к исходному слову; он будет построен как повторение суперпозиции нормальных алгорифмов $\mathfrak{E}$ и $\mathfrak{F}$, управляемое нормальным алгорифмом $\mathfrak{G}$, который будет „устанавливать", если полученное слово („междурезультат") уже результатом применения алгорифма $\mathfrak{K}$. Направление поступления „междурезультата" на следующий узел в с. г. с. будет назначено при помощи введенной отметки „$\triangle$" (отметка будет всегда перед „моделирующим" алгорифмом данного узла или к-узла с поставленным в соответствие номером). Результат трансформации поступленного слова на соответствующий узел будет „обеспечивать" алгорифм $\mathfrak{F}$ и алгорифм $\mathfrak{E}$ потом „проведет" передвижение отметки „$\triangle$" согласно соединению этого узла со следующим узлом, куда трансформированное слово будет поступать.

Алгорифм $\mathfrak{D}$ будет принят к слову, которому надо поступить на выход – он нам обеспечит, что слово, которое поступит на выход, будет в самом деле словом $\mathfrak{K}(S)$.

Замечание. К характеристике основной идеи доказательства, когда „междурезультатом" всегда какое-то слово, к которому присоединена „легенда" с записью откуда слово „приходило“ и куда ему поступить, позволяю себе привести это сравнение, которое формулировал профессор д-р Иосиф Метелка, когда прочитал первую версию этой статьи: „Может быть, что отцом этой идеи была снова практика, именно практика применяемая обычно во время производства частей, которые проходят многими производственными операциями. Эти части с собой в течении всего произведственного процесса несут сопроводительную карточку, обозначающую порядок работ. При каждом процессе обработается не только продукт, но и сопроводительная карточка, именно иногда в зависимости от самого продукта, если например только что исполнимой операцией был контроль, который установил какой-то недостаток.“

2. 1. *Алгорифм* $\mathfrak{B}$

Операторам, распознавателям и к-узлам в данной с. г. с. мы поставим в соответствие номера $1, \ldots, s$; к каждому номеру $1 \leqq i \leqq s$ определим алгорифм $\mathfrak{A}_i$, который будет „алгорифмическим выражением" функции данного узла или к-узла с. г. с.

Каждому номеру $i \in \mathbf{Pno}$ поставим в соответствие знак (букву) $[i]_0$ и из всех этих букв составим алфавит $[.]_0$ Каждому номеру $i \in \mathbf{On} \cup \mathbf{In} \cup \mathbf{Pn} \cup \mathbf{Pno}$ поставим в соответствие знак (букву) $[i]_1$ и из всех этих букв составим алфавит $[.]_1$.

Символом $U^{[\cdot]}$ обозначим любое слово удовлетворяющее следующим условиям:

1. в слово $U^{[\cdot]}$ входят все буквы алфавита $[.]_0$;
2. пусть $i \in \mathbf{Pno}$, и слово

$$\underbrace{[i]_0 \ldots [i]_0}_{a \text{ раз}}$$

обозначено символом $[i]_0^a$;* тогда для каждое i в слово $U^{[\cdot]}$ входит слово $[i]_0^a$ всегда только один раз и всегда $1 \leqq a \leqq \Theta(i) + 1$.

Когда нас будет интересовать в слове $U^{[\cdot]}$ (первое и единственное) вхождение слова $[i]_0^a$ с данным i, потом слово

$$U_1[i]_0^a U_2 ,$$

где U_1 есть начало слова $U^{[\cdot]}$ и U_2 есть конец слова $U^{[\cdot]}$, мы будем обозначать символом $U^{[i]}$; так что

$$U^{[i]} = U_1[i]_0^a U_2 .$$

Замечание. Как увидим далее в тексте, вхождение слова $[i]_0^a$ в слово $U^{[\cdot]}$ для фиксированного номера $i \in \mathbf{Pno}$, нас будет интересовать много раз; это вхождение точно записано имеет этой вид (см.: А. А. Марков [2], I, § 4):

$$U_1 * [i]_0^a * U_2 ,$$

* Подобным образом (в следующим будем употреблять)

$$[i]_1^a = \underbrace{[i]_1 \ldots [i]_1}_{a \text{ раз}} .$$

где U_1 левым крылом и U_2 правым крылом этого вхождения. Таким образом данное вхождение нам дает возможность каждое слово $U^{[\cdot]}$ написать в виде

$$U_1[i]_0^a U_2$$

а именно для так написанного слова $U^{[\cdot]}$ мы выбрали обозначение $U^{[i]}$.

Пусть все нужные информации о с. г. с. выражены в слове S^B:

$$S^B = U^{[\cdot]}\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [v-1]_1\mathfrak{A}_{v-1}^{\text{и}} \triangle [v]_1\mathfrak{A}_v^{\text{и}}[v+1]_1\mathfrak{A}_{v+1}^{\text{и}} \dots \mathfrak{A}_s^{\text{и}}[s+1]_1\,,$$

где v — номер противоположности входа и для каждого слова $[i]_0^a$ входящего в слово $U^{[\cdot]}$ есть $a = 1$; δ — „отделяющий" знак.

Объединением алфавитов, в которых запишем изображение алгорифмов $\mathfrak{A}_1, \dots, \mathfrak{A}_s$ (т. е. алгорифмов $\mathfrak{A}_1^{\text{и}}, \dots, \mathfrak{A}_s^{\text{и}}$), будет алфавит C.*

Алгорифм $\mathfrak{B}$ построим теперь в алфавите $\mathbf{C} \cup [.]_1 \cup \{\delta, \omega, \triangle\}$ со схемой

$$\begin{cases} \omega\xi \to \xi\omega \ (\xi \in \mathrm{A}) \\ \omega \ \to \cdot\, S^B \\ \quad \to \omega \end{cases}.$$

Пусть S — исходное слово в алфавите A, тогда

$$\mathfrak{B}(S) = SS^B\,.$$

2.2. Алгорифм $\mathfrak{C}$

Первым исходным словом алфорифма $\mathfrak{C}$ является таким образом слово SS^B, которое особым случаем слова

$$PU^{[\cdot]}\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \dots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1\,,$$

где P — слово в A.

Дальнейшие исходные слова этого алгорифма будут содержать слова, которые будут содержаниями отдельных памятей. К каждому узлу $P_k \in \boldsymbol{P}$ (в каждой памяти) построим поэтому алфавит двойников букв алфавита A, который обозначим через $\overline{\mathrm{A}}^k$; объединение всех этих алфавитов будет алфавит $\overline{\mathrm{A}}$:

$$\overline{\mathrm{A}} = \overline{\mathrm{A}}^1 \cup \dots \cup \overline{\mathrm{A}}^p\,.$$

Пусть таким образом является исходным словом алгорифма $\mathfrak{C}$ слово

$$PU^{[\cdot]}\overline{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \dots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1\,,$$

где $\overline{Q}$ — слово в $\overline{\mathrm{A}}$.

* Надо нам осознать, что $\mathrm{A} \cup \overline{\mathrm{A}} \cup [.]_0 \cup \{\blacktriangle\} \subset \mathrm{C}$. Принимая во внимание определения дальнейших алгорифмов, не должен алфавит C строиться как расширение объединения

$$[.]_1 \cup [.]_2 \cup \{[1]_3, \dots, [s+1]_3\} \cup \{[1]_4, \dots, [s+1]_4\} \cup \{\delta, \omega, \triangle\}\,.$$

Для

$$\mathfrak{C} = (\mathfrak{E} \circ \mathfrak{F}) \,|\, \mathfrak{G}$$

алгорифм $\mathfrak{F}$ построим так, чтобы

$$\mathfrak{F}(PU^{[\cdot]}\bar{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \ldots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \ldots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1) =$$
$$= \mathfrak{A}_i(PU^{[\cdot]}\bar{Q})\,\delta[1]_1\mathfrak{A}_1^{\text{и}} \ldots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle \mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \ldots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1 .$$

Результат действия этого алгорифма будет зависеть от узла или к-узла с поставимым в соответствие номером i. Построим теперь алгорифмы $\mathfrak{A}_i, \ldots, \mathfrak{A}_s$ такие, что

— если $i \in$ **On**, то

$$\mathfrak{A}_i(PU^{[\cdot]}\bar{Q}) = \mathfrak{O}_i(P)\,U^{[\cdot]}\bar{Q} ,$$

где $\boldsymbol{i}$ — индекс обозначения оператора, которому мы поставили в соответствие номер i (это номер узла $O_i \in \mathbf{O}$ интерпретированного алгорифмом $\mathfrak{O}_i \in \mathfrak{O}$).

Пусть B — расширение алфавита A:

$$\mathrm{B} = \mathrm{A} \cup \bar{\mathrm{A}} \cup [.]_0$$

и $\mathfrak{S}_i$ — распространение алгорифма $\mathfrak{O}_i$ (в алфавите A для $i = 1, \ldots, m$) на алфавит B. Тогда искомый алгорифм $\mathfrak{A}_i$ представляется в виде

$$\mathfrak{A}_i = \mathfrak{S}_i .$$

— если $i \in$ **In**, то

$$\mathfrak{A}_i(PU^{[\cdot]}\bar{Q}) = \mathfrak{I}_i(P)\,U^{[\cdot]}\bar{Q} ,$$

где $\boldsymbol{i}$ — индекс обозначения распознавателя, которому мы поставили в соответствие номер i (это номер узла $I_i \in \boldsymbol{I}$ интерпретированного алгорифмом $\mathfrak{I}_i \in \mathfrak{I}$).

Пусть $\mathfrak{T}_i$ — распространение алгорифма $\mathfrak{I}_i$ (в алфавите A для $i = 1, \ldots, m$) на алфавит B. Тогда искомый алгорифм $\mathfrak{A}_i$ представляется в виде

$$\mathfrak{A}_i = \mathfrak{T}_i .$$

— если $i \in$ Pn, то для

$$\bar{Q} = \bar{Q}_1\bar{Q}_2^i\bar{Q}_3 ,$$

где $\bar{Q}_1$ — слово в алфавите $\bar{\mathrm{A}}^1 \cup \ldots \cup \bar{\mathrm{A}}^{i-1}$, $\bar{Q}_2^i$ — слово в алфавите $\bar{\mathrm{A}}^i$ и $\bar{Q}_3$ — слово в алфавите $\bar{\mathrm{A}}^{i+1} \cup \ldots \cup \bar{\mathrm{A}}^p$ (вхождение слова $\bar{Q}_2^i$ в слово $\bar{Q}$ будем обсуждать и далее),

$$\mathfrak{A}_i(PU^{[\cdot]}\bar{Q}) = Q_2PU^{[\cdot]}\bar{Q}_1\bar{Q}_2^i\bar{P}^i\bar{Q}_3 ,$$

где $\boldsymbol{i}$ — индекс обозначения памяти, в которой находится к-узел с номером i (i есть номер к-узла в узле $P_i \in \mathbf{P}$); $\boldsymbol{i}$ есть определено тем самым образом и в следующих построениях.

Пусть $\mathfrak{H}_i$ — алгорифм в алфавите $\mathrm{A} \cup \overline{\mathrm{A}}^1 \cup \overline{\mathrm{A}}^2 \cup \ldots \cup \overline{\mathrm{A}}^i \cup [.]_0$, для $i = 1, \ldots, p$, со схемой

$$\begin{cases} \xi\eta \to \eta\zeta & (\xi \in \mathrm{A}, \eta \in [.]_0 \cup \overline{\mathrm{A}}^1 \cup \ldots \cup \overline{\mathrm{A}}^i) \\ \xi \to \bar{\zeta}^i & (\xi \in \mathrm{A}) \end{cases}$$

и $\mathfrak{L}_i$ — алгорифм в алфавите $\mathrm{A} \cup \overline{\mathrm{A}}^1 \cup \overline{\mathrm{A}}^2 \cup \ldots \cup \overline{\mathrm{A}}^i \cup [.]_0 \cup \{\blacktriangle\}$ для $i = 1, \ldots, p$, со схемой

$$\begin{cases} \xi\eta \to \eta\xi & (\xi \in \overline{\mathrm{A}}^1 \cup \ldots \cup \overline{\mathrm{A}}^i \cup [.]_0, \quad \eta \in \mathrm{A}) \\ \blacktriangle\xi \to \xi\bar{\xi}^i\blacktriangle & (\xi \in \mathrm{A}) \\ \blacktriangle \to \cdot & \\ \bar{\xi}^i \to \blacktriangle\bar{\xi}^i & (\xi \in \mathrm{A})\,. \end{cases}$$

Тогда искомый алкорифм $\mathfrak{A}_i$ представляется в виде

$$\mathfrak{A}_i = \mathfrak{L}_i \circ \mathfrak{H}_i\,;$$

очевидно, что

$$\mathfrak{L}_i(\mathfrak{H}_i(PU^{[\cdot]}\bar{Q})) = \mathfrak{L}_i(U^{[\cdot]}\bar{Q}_1\bar{Q}_2^i\bar{P}^i\bar{Q}_3) = Q_2PU^{[\cdot]}\bar{Q}_1\bar{Q}_2^i\bar{P}^iQ_3\,.$$

— если $i \in \mathsf{Pn}^{\wedge}$, то

$$\mathfrak{A}_i(PU^{[\cdot]}\bar{Q}) = Q_2U^{[\cdot]}\bar{Q}\,.$$

Пусть $\mathfrak{L}_i$ — алгорифм употребляемый в предыдушем и $\mathfrak{M}$ — алгорифм в алфавите А со схемой

$$\{\xi \to \quad (\xi \in \mathrm{A})\,.$$

Тогда искомый алгорифм $\mathfrak{A}_i$ представляется в виде

$$\mathfrak{A}_i = \mathfrak{L}_i \circ \mathfrak{M}\,;$$

очевидно, что

$$\mathfrak{L}_i(\mathfrak{M}(PU^{[\cdot]}\bar{Q})) = \mathfrak{L}_i(U^{[\cdot]}\bar{Q}) = Q_2U^{[\cdot]}\bar{Q}\,.$$

— если $i \in \mathsf{Pn}_{\wedge}$, то

$$\mathfrak{A}_i(PU^{[\cdot]}\bar{Q}) = U^{[\cdot]}\bar{Q}_1\bar{Q}_2^i\bar{Q}_3\,.$$

Пусть $\mathfrak{H}_i$ — алгорифм употребляемый в предыдущем. Тогда искомый алгорифм $\mathfrak{A}_i$ представляется в виде

$$\mathfrak{A}_i = \mathfrak{H}_i\,.$$

— если $i \in \mathsf{Pn}_{\wedge}^{\wedge}$, то

$$\mathfrak{A}_i(PU^{[\cdot]}\bar{Q}) = U^{[\cdot]}\bar{Q}\,.$$

Пусть $\mathfrak{M}$ — алгорифм употребляемый в предыдущем. Тогда искомый алгорифм $\mathfrak{A}_i$ представляется в виде

$$\mathfrak{A}_i = \mathfrak{M}\,.$$

— если $i \in \mathsf{Pno}$, то

$$\mathfrak{A}_i(PU^{[i]}\overline{Q}) = \begin{cases} Q_2PU_1[i]_0U_2\overline{Q}_1\overline{Q}_3, \text{ если для } U^{[i]} = U_1[i]_0^jU_2 \\ \qquad \text{есть } j = \Theta(i) + 1\,, \\ Q_2PU_1[i]_0^{j+1}U_2\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3, \text{ если для } U^{[i]} = U_1[i]_0^jU_2 \\ \qquad \text{есть } j < \Theta(i) + 1\,. \end{cases}$$

Пусть $\mathfrak{N}_i$ — алгорифм в алфавите $\mathrm{A} \cup \overline{\mathrm{A}} \cup [.]_0 \cup \{\blacktriangle\}$, для каждого $i \in \mathsf{Pno}$, со схемой

$$\begin{cases} [i]_0^{\Theta(i)} \to \blacktriangle & \\ \xi\blacktriangle \to \blacktriangle & (\xi \in \mathrm{A} \cup [.]_0) \\ \blacktriangle\xi \to \blacktriangle & (\xi \in \mathrm{A} \cup [.]_0) \\ \blacktriangle \to & ; \end{cases}$$

$\mathfrak{P}_i$ — алгорифм в алфавите A, для $i = 1, \ldots, p$, со схемой

$$\{\,\overline{\xi}^i \to \quad (\xi \in \mathrm{A})\,;$$

$\mathfrak{Q}_i$ — алгорифм в алфавите $\{[i]_0\}$, для каждого $i \in \mathsf{Pno}$, со схемой

$$\{[i]_0^2 \to [i]_0\,;$$

$\mathfrak{R}_i$ — алгорифм в алфавите $\{[i]_0\}$, для каждого $i \in \mathsf{Pno}$, со схемой

$$\{[i]_0 \to \cdot[i]_0^2\,;$$

$\mathfrak{H}_i$ и $\mathfrak{L}_i$ — алгорифмы употребляемые в предыдущем.

Тогда искомый алгорифм $\mathfrak{A}_i$ представляется в виде*

$$\mathfrak{A}_i = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i) \circ \mathfrak{L} \circ_i \mathfrak{H}_i\,;$$

очевидно, что

$$((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(\mathfrak{L}_i(\mathfrak{H}_i(PU^{[i]}\overline{Q}))) = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(\mathfrak{L}_i(U^{[i]}\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3)) =$$

$$= ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(Q_2PU^{[i]}\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3) =$$

$$= \begin{cases} \mathfrak{Q}_i(\mathfrak{P}_i(Q_2PU^{[i]}\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3)) = \mathfrak{Q}_i(Q_2PU^{[i]}\overline{Q}_1\overline{Q}_3) = Q_2PU_1[i]_0U_2\overline{Q}_1\overline{Q}_3\,, \\ \qquad \text{если } \mathfrak{N}_i(Q_2PU^{[i]}\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3) = \Lambda, \text{ т. е. если для} \\ \qquad U^{[i]} = U_1[i]_0^jU_2 \quad \text{есть} \quad j = \Theta(i) + 1\,, \\ \mathfrak{R}_i(Q_2PU^{[i]}\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3) = Q_2PU_1[i]_0^{j+1}U_2\overline{Q}_1\overline{Q}_2^i\overline{P}^i\overline{Q}_3\,, \\ \qquad \text{если } \mathfrak{N}_i(Q_2PU^{[i]}\overline{Q}_1\overline{Q}_2^i\overline{P}^iQ_3) \neq \Lambda, \text{ т. е. если для} \\ \qquad U^{[i]} = U_1[i]_0^jU_2 \quad \text{есть} \quad j < \Theta(i) + 1\,. \end{cases}$$

* $\mathfrak{A} \curlyvee \mathfrak{B} \mid \mathfrak{C}$ есть обозначение для разветвления (алгорифмов) $\mathfrak{A}$ и $\mathfrak{B}$, упревляемого (алгорифмом) $\mathfrak{C}$; подобно будет употреблаться обозначение $\mathfrak{A} \mid \mathfrak{B}$ для повторения алгорифма $\mathfrak{A}$, управляемого алгорифмом $\mathfrak{B}$ и обозначение $\mathfrak{A}\mathfrak{B}$ для объединения алгорифмов $\mathfrak{A}$ и $\mathfrak{B}$. Эти обозначения приведены в последней работе А. А. Маркова [3].

— если $i \in \mathsf{Pno}^\wedge \cup \mathsf{Pno}_\wedge \cup \mathsf{Pno}^\wedge_\wedge$, то алгорифм $\mathfrak{A}_i$ перерабатывает слово $PU^{[i]}\bar{Q}$ во слово подобное слову $\mathfrak{A}_i(PU^{[i]}\bar{Q})$ для $i \in \mathsf{Pno}$.

Если $i \in \mathsf{Pno}^\wedge$, алгорифм $\mathfrak{A}_i$ построим в виде

$$\mathfrak{A}_i = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i) \circ \mathfrak{L}_i \circ \mathfrak{M} ;$$

очевидно, что

$$((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(\mathfrak{L}_i(\mathfrak{M}(PU^{[i]}Q))) = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(\mathfrak{L}_i(U^{[i]}\bar{Q}))$$

и далее подобно как для $i \in \mathsf{Pno}$.

Если $i \in \mathsf{Pno}$, алгорифм $\mathfrak{A}_i$ построим в виде

$$\mathfrak{A}_i = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i) \circ \mathfrak{H}_i ;$$

очевидно, что

$$((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(\mathfrak{H}_i(PU^{[i]}\bar{Q})) = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(U^{[i]}\bar{Q}_1\bar{Q}_2^i\bar{P}^i\bar{Q}_3)$$

и далее подобно как для $i \in \mathsf{Pno}$.

Если $i \in \mathsf{Pno}^\wedge_\wedge$, алгорифм $\mathfrak{A}_i$ построим в виде

$$\mathfrak{A}_i = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i) \circ \mathfrak{M} ;$$

очевидно, что

$$((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(\mathfrak{M}(PU^{[i]}\bar{Q})) = ((\mathfrak{Q}_i \circ \mathfrak{P}_i) \curlyvee \mathfrak{R}_i \mid \mathfrak{N}_i)(U^{[\cdot]}_i\bar{Q})$$

и далее подобно как для $i \in \mathsf{Pno}$.

Построим искомый алгорифм $\mathfrak{F}$ как объединение суперпозиции $\mathfrak{U} \circ \mathfrak{F}_1$ и алгорифма $\mathfrak{F}_2$:

$$\mathfrak{F} = (\mathfrak{U} \circ \mathfrak{F}_1)\,\mathfrak{F}_2 ,$$

где $\mathfrak{U}$ — универсальный алгорифм над алфавитом $\mathrm{C} \cup \{\delta\}$.

Построим алгорифм $\mathfrak{F}_1$ в $\mathrm{C} \cup [.]_1 \cup \{\triangle, \delta\}$ такой, что

$$\mathfrak{F}_1(PU^{[\cdot]}\bar{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_{i+1}^{\text{и}} \dots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1) = PU^{[\cdot]}\bar{Q}\delta\mathfrak{A}_i^{\text{и}},$$

где $1 \leqq i \leqq s$; это значит, что потом

$$\mathfrak{U}(PU^{[\cdot]}Q\delta\mathfrak{A}_i^{\text{и}}) = \mathfrak{A}_i(PU^{[\cdot]}Q) .$$

Схема этого алгорифма будет иметь вид

$$\left\{\begin{array}{ll} \xi\triangle \to \triangle & (\xi \in \mathbf{C} \cup [.]_1) \\ \triangle[i]_1 \to & (1 \leqq i \leqq s) \\ [i]_1\xi \to [i]_1 & (\xi \in \mathbf{C} \cup [.]_1;\ 1 \leqq i \leqq s) \\ [i]_1 \to & (1 \leqq i \leqq s) \end{array}\right. .$$

Построим алгорифм $\mathfrak{F}_2$ в $\mathbf{C} \cup [.]_1 \cup \{\delta, \triangle\}$ такой, что

$$\mathfrak{F}_2(PU^{[\cdot]}Q\delta[1]_1\mathfrak{A}_1^{\text{и}} \ldots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \ldots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1) =$$
$$= \delta[1]_1\mathfrak{A}_1^{\text{и}} \ldots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \ldots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1 .$$

Схема этого алгорифма имеет вид

$$\begin{cases} \xi\delta \to \delta(\xi \in \mathrm{A} \cup \overline{\mathrm{A}} \cup [.]_0) \\ \delta \to \cdot\, \delta \end{cases} .$$

2.2.2. *Алгорифм* $\mathfrak{G}$

Сопоставим каждой букве (знаку) $[i]_0$ алфавита $[.]_0$ двойника обозначенного через $[i]_2$; алфавит этих двойников (т. е. двойников всех букв алфавита $[.]_0$) обозначим через $[.]_2$; алгорифм $\mathfrak{E}$ построим в алфавите $\mathbf{C} \cup [.]_1 \cup [.]_2 \cup \cup \{[1]_3, \ldots, [s+1]_3\} \cup \{[1]_4, \ldots, [s+1]_4\} \cup \{\delta, \triangle, \nabla\}$ со схемой

$$\begin{cases}
\triangle[i]_1 \to [i]_2[i]_1 & (i \in \mathbf{Pno}) \\
\xi[i]_2 \to [i]_2\xi & (\xi \in \mathrm{C} \cup \{\delta\} \cup [.]_1 \cup [.]_0 \setminus \{[i]_0\};\ i \in \mathbf{Pno}) \\
[i]_0^a[i]_2 \to [i]_0^a[\Psi(i, a-1)]_3 & (2 \leqq a \leqq \Theta(i) + 1;\ i \in \mathbf{Pno}) \\
[i]_3\xi \to \xi\,[i]_3 & (\xi \in \mathrm{C} \cup \{\delta\} \cup \{[1]_1, \ldots, [i-1]_1\}; \\
 & \quad 1 \leqq i \leqq s+1) \\
[i]_3[i]_1 \to \cdot \triangle[i]_1 & (1 \leqq i \leqq s+1) \\
\nabla\xi \to \xi\nabla & (\xi \in \mathrm{C} \cup \{\xi\} \cup [.]_1) \\
\nabla\triangle[i]_1 \to [i]_1 \triangle[\Psi(i, 2)]_4 & (i \in \mathbf{In}) \\
\xi\triangle[i]_4 \to \triangle[i]_4\xi & (\xi \in \mathrm{C};\ 1 \leqq i \leqq s+1) \\
[i]_1 \triangle [i]_4 \to \cdot \triangle[i]_1 & (1 \leqq i \leqq s+1) \\
[i]_1 \triangle [j]_4 \to \triangle [j]_4[i]_1 & (1 \leqq i \leqq s+1;\ 1 \leqq j \leqq s+1;\ i > j) \\
\triangle [i]_4[j]_1 \to [j]_1 \triangle [i]_4 & (1 \leqq i \leqq s+1;\ 1 \leqq j \leqq s+1;\ i > j) \\
\triangle [i]_1 \to [i]_1 \triangle [\Psi(i, 1)]_4 & (i \in \mathbf{On} \cup \mathbf{Pn}) \\
\xi \to \xi\nabla & (\xi \in \mathrm{A}) \\
\triangle [i]_1 \to [i]_1 \triangle [\Psi(i, 1)]_4 & (i \in \mathbf{In})
\end{cases} .$$

Можем убедиться, что если P и Q — пустые слова или слова в алфавите А; то

$$\mathfrak{E}(PU^{[i]}\overline{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \ldots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}} \triangle [i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \ldots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1) =$$
$$= PU^{[i]}\overline{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \ldots [\Psi(i, x) - 1]_1\mathfrak{A}_{\Psi(i,x)-1}^{\text{и}} \triangle$$
$$[\Psi(i, x)]_1\mathfrak{A}_{\Psi(i,x)}^{\text{и}}[\Psi(i, x) + 1]_1\mathfrak{A}_{\Psi(i,x)}^{\text{и}} \ldots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1$$

(если $i \in \mathbf{In}$ и $P = \Lambda$, то $x = 1$,
если $i \in \mathbf{In}$ и $P \neq \Lambda$, то $x = 2$,
если $i \in \mathbf{On} \cup \mathbf{Pn}$, то $x = 1$,
если $i \in \mathbf{Pno}$, то для $U^{[i]} = U_1[i]_0^aU_2$ есть $x = a - 1$).

2.2.3. *Алгорифм* $\mathfrak{G}$

Алгорифм $\mathfrak{G}$ построим в алфавите $\mathrm{C} \cup [.]_1 \cup \{\delta, \triangle\}$ со схемой

$$\begin{cases} \xi\triangle[s+1]_1 \to \triangle[s+1]_1 & (\xi \in \mathrm{C} \cup [.]_1 \cup \{\xi\}) \\ \triangle[s+1]_1 \to & ; \end{cases}$$

если $T = PU^{[\cdot]}\bar{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [i-1]_1\mathfrak{A}_{i-1}^{\text{и}}\triangle[i]_1\mathfrak{A}_i^{\text{и}}[i+1]_1\mathfrak{A}_{i+1}^{\text{и}} \dots [s]_1\mathfrak{A}_s^{\text{и}}[s+1]_1$,
то

$$\mathfrak{G}(T) = T, \text{ для } 1 \leqq i \leqq s\,;$$

$$\mathfrak{G}(PU^{[\cdot]}\bar{Q}\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [s]_1\mathfrak{A}_s^{\text{и}}\triangle[s+1]_1) = \Lambda\,.$$

2.2.4. *Алгорифм* $\mathfrak{D}$

Алгорифм $\mathfrak{D}$ в суперпозиции $\mathfrak{A}$ построим в алфавите $\mathrm{D} = \mathrm{C} \cup [.]_1 \cup \{\delta, \triangle\}$ со схемой

$$\{\xi \to \quad (\xi \in \mathrm{D} \setminus \mathrm{A})\,;$$

$$\mathfrak{D}(PU^{[\cdot]}Q\delta[1]_1\mathfrak{A}_1^{\text{и}} \dots [s]_1\mathfrak{A}_s^{\text{и}}\triangle[s+1]_1) = P \quad (P - \text{слово в } \mathrm{A})\,.$$

Замечание. Считаю излишным доказательство того, что

— если имеет смысл выражение $\mathfrak{K}(S)$, то вышеприведенные алгорифмы нам гарантируют, что имеет смысл и выражение $\mathfrak{D}(\mathfrak{C}(\mathfrak{B}(S)))$, что для равенства

$$\mathfrak{K}(S) = T$$

имеет место равенство

$$\mathfrak{A}(S) = T\,,$$

где

$$\mathfrak{A} = \mathfrak{D} \circ \mathfrak{C} \circ \mathfrak{B}\,;$$

— и наоборот, если имеет смысл выражение $\mathfrak{D}(\mathfrak{C}(\mathfrak{B}(S)))$, то способ определения алгорифмов $\mathfrak{B}, \mathfrak{C}, \mathfrak{D}$ нам гарантирует, что имеет смысл и вытажение $\mathfrak{K}(S)$, что для равенства

$$\mathfrak{A}(S) = T\,,$$

где

$$\mathfrak{A} = \mathfrak{D} \circ \mathfrak{C} \circ \mathfrak{B}\,,$$

имеет место равенство

$$\mathfrak{A}(S) = T\,.$$

(Поступило 7-го апреля 1967 г.)

[1] Л. А. Калужнин: Об алгоритмизации математических задач. Сб. „Проблемы кибернетики", вып. 2; Физматгиз, Москва 1959, 51—67.
[2] А. А. Марков: Теория алгорифмов. Труды Матем. ин-та АН СССР, *XLII*. Москва—Ленинград 1954.
[3] А. А. Марков: О некоторых алгорифмах, связанных с системами слов. Известия АН СССР, серия математическая *27* (1963), 101—160.
[4] Н. П. Тер-Захарян: О нормализуемости граф-схемных алгорифмов. Сб. „Математические вопросы кибернетики и вычислительной техники", Ереван 1965, 30—39.

VÝTAH

O algoritmech s paměťovými prvky

ZDENĚK ZASTÁVKA

V článku je definován algoritmus s paměťovými prvky pomocí grafického schématu (pojem grafického schématu je obdobou Kalužninovy definice v jeho práci [1]). Tento algoritmus je nazván kompozičním algoritmem, protože je zejména vhodný pro sestavování algoritmů kompozic daných algoritmů (kompozičních algoritmů), jejichž nejpoužívanější druhy jsou definovány ve větách 1 až 4.

Interpretujeme-li jednotlivé uzly grafického schématu definovaného algoritmu normálními algoritmy (ve smyslu práce A. A. Markova [2]), dostaneme tzv. kvazinormální kompoziční algoritmy (k. k. a.). Ukáže se zřejmým, že ke každému normálnímu algoritmu existuje ekvivalentní k. k. a. (věta 5); je dokázáno, že ke každému k. k. a. existuje ekvivalentní normální algoritmus (věta 6.).

Dr. Zdeněk Zastávka, CSc., oddělení logiky katedry filosofie, filosofická fakulta Karlovy university, náměstí Krasnoarmějců 2, Praha 1.