А. Г. ПИЛИПОСЯН, С. М. ДАВТЯН

ОПИСАНИЕ УНИВЕРСАЛЬНОЙ ПРОГРАММИРУЮЩЕЙ ПРОГРАММЫ, ВЫПОЛНЯЮЩЕЙ НЕКОТОРЫЕ ПРЕОБРАЗОВАНИЯ ЛОГИЧЕСКИХ СХЕМ

Введение

Решение математических задач на современных быстродействующих электронных вычислительных машинах практически осуществляется следующим образом:

1) выбирается численный метод, обеспечивающий решение задачи с требуемой точностью;

2) строится алгоритм счета (схема счета), соответствующий выбранному численному методу;

3) составляется программа, которая реализует алгоритм решения данной задачи на конкретной вычислительной машине.

Составление программы по алгоритму решения данной задачи представляет собою трудоемкий процесс, который зачастую требует больше времени, чем само решение задачи на электронной вычислительной машине. Возникает вопрос об эффективном увеличении производительности труда при программировании решаемых задач. Этот вопрос решается путем автоматизации процесса программирования. Проблеме автоматизации программирования посвящены, например, работы [7, 8, 9]. Описанные в этих работах программирующие программы (ПП) полностью решают вопрос автоматического программирования операторной схемы, представляющей собой описание алгоритма счета данной задачи в терминах операторов.

Для записи алгоритма счета задачи применяются различные методы. Разнообразие задач, решаемых в Вычислительном центре Академии наук Армянской ССР и Ереванского государственного университета с применением операторного метода А. А. Ляпунова для записи алгоритмов счета, дало большой материал для изучения возможностей этого метода при автоматизации программирования.

Для каждого численного метода может быть построено несколько алгоритмов счета данной задачи и, следовательно, несколько программ.

Среди алгоритмов решения задачи особый интерес представляют те, программная реализация которых наиболее рациональна как с точки зрения машинного времени, так и с точки зрения занимаемого объема памяти. Поэтому важную роль играет вопрос преобразования алгоритмов к виду, дающему рациональную в указанном смысле программу. Некоторым таким преобразованиям посвящены работы Р. И. Подловченко [5, 6]. Одним из результатов этих работ является преобразование алгоритма решения данной задачи к виду с более рациональной машинной реализацией.

Автоматизация программирования может быть осуществлена наиболее употребительными методами—методом универсальных программирующих программ и методом библиотеки стандартных подпрограмм. В последнее время в ряде ПП, а также в описываемой ПП комбинируются оба метода автоматизации программирования.

Схема счёта представляет собою запись алгоритма счета, непосредственная машинная реализация которого часто приводит к нерациональным программам. Ввиду этого возникает необходимость построения промежуточных логических схем, учитывающих специфику современных вычислительных машин и содержащих возможность приведения к рациональному, с точки зрения машинной реализации, виду путем формальных преобразований.

Языком описываемой универсальной ПП является язык операторов А. А. Ляпунова [1, 2]. Вся необходимая информация для правильной работы ПП представляет собой код схемы счета с описанием операторов этой схемы и некоторые сведения о параметризации числовой информации задачи. Последние необходимы для формальных преобразований логических схем решаемой задачи. В отличие от известных ПП, предлагаемая ПП осуществляет автоматическое построение промежуточных логических схем с преобразованием перехода от одной параметризации к другой по принципам, разработанным Р. И. Подловченко [5]. Кроме того, ПП осуществляет формальные преобразования логических схем по определенным правилам, которые упрощают эти схемы с точки зрения их структуры и выполнения. В ПП предусмотрено автоматическое включение стандартных подсхем и стандартных подпрограмм. Работа ПП может быть разбита на две части. Первой частью работы ПП осуществляется построение промежуточных логических схем и их преобразование по следующим этапам:

1) переход от схемы счета к схеме программы с автоматическим включением необходимых подсхем;

2) переход от схемы программы к рабочей схеме программы;

3) формальные преобразования рабочей схемы программы.

Второй частью работы ПП осуществляется переход от преобразованной рабочей схемы к программе в истинных адресах с использованием некоторых возможностей оптимизации программы и с автоматическим включением необходимых стандартных программ.

Отметим некоторые особенности данной ПП.

1. ПП может начать работу с любого этапа первой части, а также со второй части, после задания информации, необходимой для работы данного этапа или данной части.

Класс задач, алгоритмы решения которых допускают описание в виде схемы программы, шире класса задач, алгоритмы решения которых описываются в виде схемы счета, не содержащей символа нестандартного оператора. В силу особенности 1, класс задач, программируемых при помощи ПП, не ограничивается последним*.

2. В логических схемах допускаются пересекающиеся произведения**.

3. ПП строит промежуточные логические схемы и осуществляет формальные преобразования схем для приведения к виду, более рациональному с точки зрения машинной реализации.

4. В разных частях и на разных этапах работы ПП предусматривается использование уже накопленной библиотеки стандартных подсхем и стандартных подпрограмм.

5. Пусть данная задача является общей по отношению к некоторой задаче. Преобразование схем при переходе от одной параметризации к другой, осуществляемое ПП, дает возможность автоматически перейти от схемы программы общей задачи к схеме программы частной задачи. Так, например, имея схему программы решения задачи Дирихле в квадратной области итерационным методом, можно получить схему программы той же задачи при симметричности граничных условий с запоминанием только тех значений функции, которые лежат на диагонали и ниже ее.

Описанные в данной работе алгоритмы построения логических схем и программирования некоторых операторов были запрограммированы и опробованы на модернизированной машине М-3 Вычислительного центра Академии наук Армянской ССР и Ереванского государственного университета в отдельности, ввиду ограниченности памяти машины.

Данная работа состоит из восьми глав. В первой главе приводится описание теоретических основ предлагаемой ПП. При этом в разработанной системе определений логическая схема некоторого алгоритма рассматривается в запоминающем устройстве машины. Во второй главе описывается класс схем, программируемых при помощи ПП. В третьей главе рассматриваются некоторые вопросы, связанные с построением и реализацией оператора формирования. В четвертой главе приводится общая структура ПП. Главы пятая, шестая и седьмая посвящены преобразованиям логических схем, а глава восьмая —

---

* Определение схемы счета и схемы программы приводится в § 1, гл. I.

** Определение пересекающихся произведений см. на стр. 22.

программированию некоторых операторов ПП. В настоящей статье публикуются первые три главы работы.

Авторы приносят искреннюю благодарность И. Д. Заславскому и Р. И. Подловченко за ряд ценных советов, высказанных при ознакомлении с рукописью.

ГЛАВА I

ОПИСАНИЕ ТЕОРЕТИЧЕСКИХ ОСНОВ ПП

§ 1. Определение основных понятий

Основные принципы этой главы разработаны в работах А. А. Ляпунова [1, 2] и Р. И. Подловченко [3, 5, 6]. В настоящем параграфе дается определение основных понятий, используемых в дальнейшем при описании ПП.

1. Всякую упорядоченную пару чисел (l, m), где l — натуральное, а m — рациональное, назовем величиной.

Пусть $T = \{m_k\}$, $k = 1, 2, \cdots, q$, — некоторое множество рациональных чисел, и x — символ переменной, значениями которой являются числа множества T. Пару (l, x), где l — натуральное число, а x — символ переменной, назовем переменной величиной. Множество T назовем областью определения переменной величины (l, x), а величины (l, m_k), $m_k \in T$, $k = 1, \cdots, q$, — значениями переменной величины (l, x).

В частности, если множество T состоит из одного рационального числа, величину (l, x) будем называть постоянной величиной. Если число m — целое, то величину (l, m) будем называть целочисленной величиной. Примером величины может служить номер ячейки запоминающего устройства ЭВМ вместе с ее содержимым.

2. Пусть $M = \{(l_r, x)\}$, $r = 1, 2, \cdots, n$, — множество переменных величин, $T_r = \{m^{(r)}_{k_r}\}$, $k_r = 1, 2, \cdots, t_r$, — область определения переменной величины $(l_r, x) \in M$, $r = 1, \cdots, n$. Пусть $T = \{(m^{(1)}_{k_1}, m^{(2)}_{k_2}, \cdots, m^{(n)}_{k_n})\}$, где $m^{(r)}_{k_r} \in T_r$, $r = 1, 2, \cdots, n$, — упорядоченное множество. Любое подмножество множества T назовем областью определения множества M, а каждый элемент множества T — $(m^{(1)}_{k_1}, m^{(2)}_{k_2}, \cdots, m^{(n)}_{k_n})$ - набором, определяющим значения переменных величин множества M. В дальнейшем указанный набор будем называть набором значений переменных величин множества M. Длиной данного конечного набора σ или конечного множества M назовем количество элементов в этом наборе или множестве и будем обозначать через $d(\sigma)$ или $d(M)$. Объединением наборов $(z_1, z_2, \cdots, z_n)$ и $(y_1, y_2, \cdots, y_q)$ назовем набор $(z_1, z_2, \cdots, z_n, y_1, y_2, \cdots, y_q)$.

Две переменные величины (l_1, x_1) и (l_2, x_2) с областями определения T_1 и T_2 соответственно называются **подобными** ($(l_1, x_1) \sim (l_2, x_2)$), если $l_1 = l_2$. Подобные величины называются **равными**, если множества T_1 и T_2 совпадают. В частном случае, из данных определений следуют определения подобия и равенства постоянных величин.

В дальнейшем будем предполагать, что для каждого рассматриваемого множества величин M задана некоторая область определения этого множества. Кроме того, будем предполагать, что каждое рассматриваемое множество M конечное и не содержит подобных величин.

3. **Отображением** G множества переменных величин M на множество переменных величин N $(G: M \to N)$ назовем некоторое правило, которое каждому набору значений переменных величин множества M сопоставляет набор значений переменных величин множества N, при этом всякий набор значений переменных величин множества N соответствует некоторому набору значений переменных величин множества M. Всякое отображение G множества M на множество $N (G: M \to N)$ назовем **оператором** над множеством M. Множество M назовем **входом** оператора, а множество N — **выходом** оператора; величины множества M назовем входными величинами, величины множества N — выходными величинами.

Пусть $O = \{m_k\}$, $k = 1, 2, \cdots, p$ — конечное фиксированное множество рациональных чисел и $A = \{A_i\}$, $i = 1, 2, \cdots$ — некоторое счетное множество операторов. Пусть M_{A_i} и N_{A_i} — вход и выход оператора $A_i \in A$. Каждому оператору A_i $(i = 1, 2, \cdots)$, представляющему отображение заданного множества M_{A_i} на N_{A_i}, сопоставим упорядоченную последовательность $(m_{k_1}^{(i)}, m_{k_2}^{(i)}, \cdots, m_{k_r}^{(i)}) = S^{(i)}$, где $m_{k_j}^{(i)} \in O$, $j = 1, 2, \cdots, r$, так, чтобы операторам A_i и A_l, $i \neq l$, были сопоставлены отличные друг от друга последовательности.

Пусть множества $M_{\overline{A}}$ и $N_{\overline{A}}$ — вход и выход оператора $\overline{A}$ и пусть между величинами множеств P и Q и некоторых множеств $\overline{P}$, $\overline{Q}$, $\overline{P} \subset M_{\overline{A}}$, $\overline{Q} \subset N_{\overline{A}}$, установлено взаимно-однозначное соответствие. По оператору $\overline{A}$ построим оператор B следующим образом: в множествах $M_{\overline{A}}$ и $N_{\overline{A}}$ величины множеств $\overline{P}$ и $\overline{Q}$ заменим на соответствующие им величины множеств P и Q.

Пусть $C_{\overline{A}}$ — множество операторов, полученных описанным образом. Будем предполагать, что операторам множества $C_{\overline{A}}$ сопоставлены разные последовательности одной и той же длины.

Числам m_{k_j}, $j = 1, \cdots, r$, сопоставим последовательные натуральные числа и построим величины (l_j, m_{k_j}), $m_{k_j} \in S$, $j = 1, \cdots, r$, которые составят последовательность величин $\overline{S}$. Величины (l_j, m_{k_j}), $j = 1, 2, \cdots, r$ будем называть **операторными величинами**.

В частном случае последовательность $\bar{S}$ может состоять из одной операторной величины.

В качестве множества O можно взять, например, перечень машинных команд. Тогда последовательность $\bar{S}$ представляет собой программную реализацию, а операторная величина — команду программы. Последовательность $\bar{S}$ будем называть **операторной последовательностью**.

Величину (l, m), которая не принадлежит ни одной операторной последовательности, сопоставленной операторам класса $\{A\}$, назовем **числовой величиной**.

Операторы будем обозначать заглавными буквами латинского алфавита — $A, B, C, \cdots$, величины — строчными — $a, b, c, \cdots$.

Выполнением отображения (оператора) назовем получение по заданному набору значений величин множества M сопоставленного ему набора значений величин множества N.

Пусть A и B представляют собою соответственно отображения $M_1 \to N_1$ и $M_2 \to N_2$ и пусть $\bar{N} \subset N_1$ — множество величин, для которых существуют в множестве M_2 подобные величины.

Пусть $M_1 = (x_1, x_2, \cdots, x_n)$, $N_1 = (\bar{x}_2, \cdots, \bar{x}_t, \bar{x}_{t+1}, \cdots, \bar{x}_{t+s})$

$$M_2 = (y_1, y_2, \cdots, y_m, y_{m+1}, \cdots, y_{m+s}), \quad N_2 = (\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_k)$$

$$\bar{x}_{t+j} \sim y_{m+j}, \; j = 1, 2, \cdots, s$$

и пусть $\bar{N} = (\bar{x}_{t+1}, \cdots, \bar{x}_{t+s})$ (без ограничений можно предположить, что подобные величины множеств N_1 и M_2 упорядочены указанным образом).

Произведением операторов A и B назовем оператор $C = AB$, который состоит в последовательном выполнении операторов A и B. Множество M — входных величин оператора C — составляется из величин множества M_1 и тех величин множества M_2, для которых нет подобных в множестве $\bar{N}$. Множество N — выходных величин оператора C — составляется из величин множества N_2 и тех величин множества N_1, для которых нет подобных величин в множестве N_2. Таким образом,

$$M = (x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_m),$$

$$N = (\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_k, \bar{x}_{i_1}, \bar{x}_{i_2}, \cdots, \bar{x}_{i_p})$$

($\bar{x}_{i_1}, \bar{x}_{i_2}, \cdots, \bar{x}_{i_p}$ — величины множества N_1, для которых в множестве N_2 нет подобных величин). Область определения множества M представляет собою множество наборов длины $n + m$ значений переменных величин $x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_m$. Каждый набор значений из области определения множества M представляет собою объединение таких наборов из области определения множества M_1 и наборов значений величин $y_1, y_2, \cdots, y_m$ из области определения множества M_2, которые удовлетворяют следующему условию: набор значений длины $m + s$, который образуется объединением набора значений величин

$y_1, y_2, \cdots, y_m$ из области определения множества M_2 и набора значений величин множества $\overline{N}$, сопоставленного заданному набору значений из области определения множества M_1 оператором A, принадлежит области определения множества M_2.

Область определения переменных величин множества N — выходных величин оператора C — представляет собою объединение наборов значений переменных величин $\overline{y}_1, \overline{y}_2, \cdots, \overline{y}_k$ и $\overline{x}_{l_1}, \cdots, \overline{x}_{l_p}$, которые получаются при выполнении оператора C по заданному набору из области определения множества M.

4. Пусть $(i_1, i_2, \cdots, i_m)$ — упорядоченное множество целочисленных переменных величин, I_m — область определения набора переменных величин $i_1, i_2, \cdots, i_m$.

а) Пусть $M = \{(l, m)\}$ — множество числовых величин, содержащее более одной величины. Соответствие между множествами I_m и M назовем **параметризацией** множества числовых величин параметрами $i_1, i_2, \cdots, i_m$, если каждому упорядоченному набору $(i_1^\circ, i_2^\circ, \cdots, i_m^\circ) \in I_m$ значений переменных величин $i_1, i_2, \cdots, i_m$ сопоставлено единственное число l такое, что $(l, m) \in M$ и, наоборот, для каждой величины $(l, m) \in M$ числу l сопоставлен единственный упорядоченный набор $(i_1^\circ, \cdots, i_m^\circ) \in I_m$. Фактически, параметризация множества M представляет собою функцию, которая по заданному набору $(i_1^\circ, \cdots, i_m^\circ) \in I_m$ определяет элементы множества M. Из определения параметризации ясно, что для функции параметризации существует обратная функция, т. е. функция параметризации — обратимая функция.

Функцию параметризации множества M будем записывать в виде $(l_{i_1 i_2 \cdots i_m}, m)$, где $i_1, i_2, \cdots, i_m$ — символы переменных, и обозначать через $a_{i_1 i_2 \cdots i_m}$. Элементы множества M определяются функцией $a_{i_1 \cdots i_m}$ по заданному набору $(i_1^\circ, i_2^\circ, \cdots, i_m^\circ) \in I_m$, а потому, вместо элементов параметризованного множества M, можно рассматривать его функцию параметризации. Функцию параметризации множества числовых величин M будем называть параметрической числовой величиной. Множество M будем называть параметризованным, величины $i_1, i_2, \cdots, i_m$ — параметрами множества M.

б) Пусть (l, x) — переменная операторная величина и L — область определения величины (l, x).

Область определения операторной переменной величины (l, x) назовем параметризованной, если каждому набору $(i_1^\circ, \cdots, i_m^\circ) \in I_m$ значений величин $i_1, i_2, \cdots, i_m$ соответствует единственное число m°, такое, что $m^\circ \in L$ и, наоборот, каждому $m^\circ \in L$ соответствует единственный упорядоченный набор $(i_1^\circ, i_2^\circ, \cdots, i_m^\circ) \in I_m$. Аналогично, параметризация множества L есть обратимая функция, которая по каждому набору $(i_1^\circ, \cdots, i_m^\circ) \in I_m$ определяет элемент множества L. Функцию параметризации области определения переменной операторной величины будем записывать в виде $(L, x_{i_1 i_2 \cdots i_m})$, где $i_1, i_2, \cdots, i_m$ — символы переменных. Функцию параметризации множества L будем называть параметрической операторной величиной.

В дальнейшем будем предполагать, что область определения каждой рассматриваемой переменной операторной величины параметризована.

в) Пусть N — множество числовых величин, параметризованное параметрами $i_1, i_2, \cdots, i_m$, I_m — область определения величин $i_1, i_2, \cdots, i_m$. Составим для каждого фиксированного набора значений параметров множество вида

$$M = M_1 \cup (l_{\bar{i}_1 \cdots \bar{i}_m}, \bar{m}),$$

где $(l_{\bar{i}_1 \cdots \bar{i}_m}, \bar{m}) \in N$, $(\bar{i}_1, \cdots, \bar{i}_m) \in I_m$. M_1 — некоторое фиксированное множество величин, не содержащее величин из множеств, параметризованных параметрами $i_1, i_2, \cdots, i_m$. Рассмотрим множество указанных множеств для всех наборов из области I_m и обозначим его через $\{M\}$. Очевидно, что каждому набору значений параметров $(\bar{i}_1, \bar{i}_2, \cdots, \bar{i}_m) \in I_m$ соответствует единственное множество $M \in \{M\}$ и, наоборот, каждому множеству $M \in \{M\}$ соответствует единственный набор $(\bar{i}_1, \bar{i}_2, \cdots, \bar{i}_m) \in I_m$. Указанное соответствие назовем параметризацией множества $\{M\}$ параметрами $i_1, i_2, \cdots, i_m$. Параметризация множества $\{M\}$ — обратимая функция, которая по каждому набору $(\bar{i}_1, \cdots, \bar{i}_m) \in I_m$ определяет $M \in \{M\}$. Функцию параметризации множества $\{M\}$ будем обозначать через $M_{i_1 i_2 \cdots i_m}$, где $i_1, i_2, \cdots, i_m$ — символы переменных. Параметризованному множеству сопоставим его функцию параметризации $M_{i_1 i_2 \cdots i_m}$.

Если некоторое множество $\overline{M}$ содержит величины из нескольких параметризованных множеств $N_1, N_2, \cdots, N_k$, то, аналогично, можно ввести параметризацию множества $\{\overline{M}\}$ посредством множества параметров, представляющего собой теоретико-множественную сумму множеств параметров* множеств $N_1, N_2, \cdots, N_k$, и функцию параметризации множества $\{\overline{M}\}$. Так как элемент множества $\{\overline{M}\}$ определяется по заданному набору параметров функцией параметризации этого множества, то в дальнейшем, вместо элементов множества $\{\overline{M}\}$ будем рассматривать его функцию параметризации, которую назовем параметрическим множеством. Параметрическое множество для каждого набора значений его параметров представляет собою множество $\overline{M} \in \{\overline{M}\}$, соответствующее этому набору.

---

* Предполагается, что отличные друг от друга параметры обозначены различными символами переменных, а равные параметры—одним и тем же символом. Параметры каждого множества $N_1, N_2, \cdots, N_k$ рассматриваются как множество символов, операция теоретико-множественной суммы выполняется над этими множествами.

Пусть $M_{i_1 i_2 \cdots i_m}$ и $N_{j_1 j_2 \cdots j_k}$ — параметрические множества, $(i_1, \cdots, i_m)$ и $(j_1, \cdots, j_k)$ — упорядоченные множества отличных друг от друга параметров с областями определения I_m и J_k соответственно. Пусть, далее, параметрические множества $M_{i_1 \cdots i_m}$ и $N_{j_1 \cdots j_k}$ для каждых наборов $(i_1^\circ, \cdots, i_m^\circ) \in I_m$ и $(j_1^\circ, \cdots, j_k^\circ) \in J_k$ представляют собою множества входных и выходных величин нижеследующих операторов. При этом наборы значений входных и выходных величин операторов сопоставляются одним и тем же правилом.

$$
\begin{aligned}
&M_{i_1^{(1)} \cdots i_m^{(1)}} \to N_{j_1^{(1)} \cdots j_k^{(1)}}\\
&M_{i_1^{(2)} \cdots i_m^{(2)}} \to N_{j_1^{(2)} \cdots j_k^{(2)}}\\
&\cdots\cdots\cdots\cdots\\
&M_{i_1^{(\bar{p})} \cdots i_m^{(\bar{p})}} \to N_{j_1^{(\bar{p})} \cdots j_k^{(\bar{p})}}
\end{aligned}
\qquad (1)
$$

$$(i_1^{(t)}, \cdots, i_m^{(t)}) \in I_m, \quad (j_1^{(t)}, \cdots, j_k^{(t)}) \in J_k, \quad t = 1, 2, \cdots, \bar{p}.$$

Обозначим это множество операторов через $\{A_i\}$, $i = 1, 2, \cdots, \bar{p}$.

Множество операторов $\{A_i\}$, $i = 1, \cdots, \bar{p}$ будем называть параметризованным параметрами $i_1, i_2 \cdots, i_m, j_1, j_2, \cdots, j_k$, так как каждым наборам $(i_1^\circ, i_2^\circ, \cdots, i_m^\circ) \in I_m$ и $(j_1^\circ, \cdots, j_k^\circ) \in J_k$ соответствует единственный оператор $A \in \{A_i\}$ и, наоборот, каждому оператору $A \in \{A_i\}$ соответствует единственная пара наборов $(i_1^\circ, i_2^\circ, \cdots, i_m^\circ) \in I_m$ и $(j_1^\circ, j_2^\circ, \cdots, j_k^\circ) \in J_k$. Функцию параметризации множества операторов $\{A_i\}$ обозначим через $A_{i_1 \cdots i_m j_1 \cdots j_k}$. Элементы множества $\{A_i\}$ определяются функцией $A_{i_1 \cdots i_m j_1 \cdots j_k}$ по заданным наборам $(i_1^\circ, \cdots, i_m^\circ) \in I_m$ и $(j_1^\circ, \cdots, j_k^\circ) \in J_k$, а потому в дальнейшем вместо элементов параметризованного множества операторов будем рассматривать его функцию параметризации и называть ее параметрическим оператором.

В ы п о л н е н и е п а р а м е т р и ч е с к о г о о п е р а т о р а для определенного набора значений параметров означает выполнение соответствующего ему оператора. Заметим, что если множества $(i_1, i_2, \cdots, i_m)$ и $(j_1, j_2, \cdots, j_k)$ содержат равные параметры, то параметризация множества операторов $\{A_i\}$, $i = 1, 2, \cdots, \bar{p}$, осуществляется параметрами, представляющими собою теоретико-множественную сумму параметров $i_1, i_2, \cdots, i_m$ и $j_1, j_2, \cdots, j_k$. Очевидно, что каждому отображению из (1) будут сопоставлены различные упорядоченные последовательности одной и той же длины:

$$S_i = (m_1^{(i)}, m_2^{(i)}, \cdots, m_q^{(i)}), \; i = 1, 2, \cdots, \bar{p},$$

$$\text{где } m_k^{(i)} \in O^*, \; i = 1, 2, \cdots, \bar{p}, \; k = 1, 2, \cdots, q.$$

Пусть $l_1, l_2, \cdots, l_q$ — последовательные натуральные числа. Построим следующие последовательности операторных величин:

* Описание множества O см. стр. 9.

$$\{(l_1, m_1^{(i)}), (l_2, m_2^{(i)}), \cdots, (l_q, m_q^{(i)})\} \quad i=1, 2, \cdots, \overline{p}.$$

Построенные последовательности можно рассматривать как значения последовательности переменных операторных величин

$$\{(l_1, m_1), (l_2, m_2), \cdots, (l_q, m_q)\}$$

с областью определения

$$\{(m_1^{(i)}, m_2^{(i)}, \cdots, m_q^{(i)})\}, \; i=1, 2, \cdots, \overline{p}.$$

Эту последовательность сопоставим параметрическому оператору $A_{l_1 \cdots l_m l_1 \cdots l_k}$. Таким образом, ясно, что в операторной последовательности, сопоставленной параметрическому оператору, есть хоть одна параметрическая операторная величина.

5. Пусть N — множество величин, и предикат з определен на области определения множества N. Пусть (p, x) — переменная величина, где $x = \{0, 1\}$. Отображение $N \to (p, x)$ такое, что $x=1$, если предикат з истинен на заданном наборе значений величин множества N, и $x=0$, если он ложен, назовем **логическим оператором** P над множеством N.

Опишем некоторые символы, используемые в дальнейшем. Символы $\downarrow$ и $\uparrow$ будем называть **стрелками**, стрелку вида $\downarrow$ — входящей стрелкой, а стрелку вида $\uparrow$ — выходящей стрелкой. Стрелке приписывается натуральное число, называемое номером стрелки.

Стрелки с номером записываются в виде символов $\downarrow^N$, $\uparrow^N$ и называются **стрелками управления**. Символы $\downarrow^i$, $\uparrow^i$, где стрелкам приписан некоторый параметр i, будем называть **стрелками произведения** по параметру i.

6. Пусть даны следующие множества:

а) множество операторов $\{A\}$, заданных над множеством величин $\{M_A\}$ соответственно*;

б) множество логических операторов $\{P\}$, заданных над множествами величин $\{N_P\}$ соответственно*;

в) множество пар стрелок управления $\{\downarrow^N, \uparrow^N\}$;

г) множество пар стрелок произведения $\{\downarrow^i, \uparrow^i\}$;

д) символы H** и „останов".

Пусть

$$S_1, S_2, \cdots, S_n \tag{2}$$

последовательность, состоящая из конечного числа элементов, каждый из которых принадлежит одному из указанных множеств. На структуру последовательности (2) наложим следующие ограничения:

а) каждой выходящей стрелке произведения соответствует одна и только одна входящая стрелка произведения с тем же параметром и наоборот;

б) каждой выходящей стрелке управления соответствует одна и только одна входящая стрелка управления с тем же номером;

---

* Множества $\{A\}$ и $\{P\}$ могут содержать параметрические операторы.

** Символ H представляет собой символ нестандартного оператора, описываемого на стр. 20.

в) если в последовательности (2) элемент S_k — логический оператор, то элемент S_{k-1} — выходящая стрелка произведения или управления;

г) каждая выходящая стрелка произведения стоит вправо от соответствующей ей входящей стрелки произведения*;

д) всякий параметрический оператор находится между стрелками произведения по его параметрам;

е) для каждой входящей стрелки управления, расположенной между стрелками произведения, соответствующая ей выходящая стрелка управления расположена между этими же стрелками произведения.

Последовательность (2), удовлетворяющую указанным условиям, назовем **логической схемой**.

Если S_k — выходящая стрелка управления, а S_{k-1} — логический оператор, то выходящую стрелку S_k и соответствующую ей входящую стрелку будем называть стрелками условного управления, если же S_{k-1} некоторый, отличный от логического оператора элемент, то указанные стрелки будем называть стрелками безусловного управления.

Отрезок логической схемы, заключенный между входящей и выходящей стрелками произведения по некоторому параметру i, вместе со стрелками произведения будем называть **произведением по параметру** i. Начало произведения изображается входящей стрелкой произведения, а конец — выходящей стрелкой произведения.

Пусть задано множество $\overline{T}$ конечных упорядоченных последовательностей целочисленных значений параметра i:

$$\overline{T} = \{i_1^{(j)}, i_2^{(j)}, \cdots, i_{k_j}^{(j)}\}, \quad j = 1, 2, \cdots, p.$$

Рассмотрим две переменные величины i_1 и i_k с областями определения $T_1 = \{ i_1^{(j)} \}$ и $T_k = \{ i_{k_j}^{(j)} \}$, $j = 1, 2, \cdots, p$. Величины i_1 и i_k назовем соответственно начальным и конечным значением параметра i и будем в дальнейшем обозначать символами $i_{нач}$ и $i_{кон}$.

Переменную величину с областью определения

$$T = \{ \Delta i_r^{(j)} \}, \quad \text{где } \Delta i_r^{(j)} = i_r^{(j)} - i_{r-1}^{(j)}; \; r = 2, 3, \cdots, k_j; \; j = 1, 2, \cdots, p$$

назовем приращением параметра i и будем обозначать символом Δi.

Пусть $\overline{T} = \{(\overline{i}_1^{(j)}, \overline{i}_2^{(j)}, \cdots, \overline{i}_k^{(j)}, \cdots)\}$ — множество бесконечных для всех $j = 1, 2, \cdots, p$ упорядоченных последовательностей значений параметра i. Тогда величины $i_{нач}$ и Δi определяются аналогично. Величина $i_{кон}$ аналогично не может быть определена. В этом случае будем говорить, что $i_{кон}$ — неизвестно.

В данной работе предполагается, что для любого параметра выполнено следующее условие: если в множестве $\overline{T}$ имеется хотя бы

* Соответствующими называем стрелки с одним и тем же номером или параметром.

одна конечная последовательность, то все последовательности множества T — конечны. Множество $\overline{T}$ назовем областью задания параметра i. Очевидно, что область задания параметра i может быть построена, если заданы величины $i_{нач}$, $i_{кон}$, Δi с областями определения T_1, T_2, T соответственно. (Заметим, что области T_1, T_2, T упорядочены соответственным образом). В случае, если $i_{кон}$ — неизвестно, в логической схеме перед выходящей стрелкой произведения по параметру i должен быть логический оператор, обеспечивающий выделение конечной подпоследовательности из данной бесконечной последовательности множества $\overline{T}$. В дальнейшем такой логический оператор будем называть **особым логическим оператором.**

Очевидно, что если для множества $\overline{T} = \{(i_1^{(j)}, i_2^{(j)}, \cdots, i_{k_j}^{(j)})\}$, $j = 1, 2, \cdots, p$, $p = 1$, то $i_{нач}$ и $i_{кон}$ — постоянные величины. Если $i_{нач}$, $i_{кон}$, Δi (предполагается, что $i_{кон}$ не является неизвестным) — переменные величины, то к схеме задаются функции, определяющие значения величин $i_{нач}$, $i_{кон}$, Δi. Указанные функции будем называть **функциями определения величин** $i_{нач}$, $i_{кон}$, Δi. Функции определения могут зависеть от переменных и параметрических числовых величин. При этом функции определения могут зависеть только от тех величин, значения которых заданы перед выполнением входящей и выходящей стрелки произведения по данному параметру*. Ясно, что функции определения могут быть реализованы операторами, входными и выходными величинами которых являются соответственно аргументы и значения функций.

Пусть L — логическая схема и M — множество входных величин всех операторов схемы (в том числе и нестандартных операторов, символы которых имеются в данной схеме) и операторов, реализующих функции определения параметров данной схемы, с областью определения T. Выполнение логической схемы над некоторым набором $S \in T$ состоит в последовательном выполнении элементов схемы L. Выполнение элементов логической схемы заключается в изменении в данном наборе S значений некоторых величин и в выборе следующего выполняемого элемента. При выполнении оператора по заданному набору $S \in T$ составляется набор значений входных величин выполняемого оператора. В наборе $S \in T$ значениям величин множества M, подобных выходным величинам оператора, придаются значения, полученные после выполнения этого оператора. **Выполнение** элементов логической схемы L над набором $S \in T$ определяется следующим образом:

I. Выполнение логической схемы начинается с выполнения элемента S_1;

II. а) если выполняемый элемент S_k — оператор, отличный от логического, то после его выполнения** выполняется элемент S_{k+1};

---

* Выполнение входящей и выходящей стрелки произведения см. на стр. 17.

** Выполнение оператора и параметрического оператора см. на стр. 10 и 13.

б) если выполняемый элемент S_k — выходящая стрелка управления, то выполняется соответствующая входящая стрелка управления;

в) если выполняемый элемент S_k — входящая стрелка управления, то выполняется элемент S_{k+1};

г) если выполняемый элемент S_k — логический оператор, то после его выполнения выполняется либо элемент S_{k+1}, если значение выходной величины логического оператора равно нулю, либо элемент S_{k+2}, если значение выходной величины логического оператора равно единице;

д) если выполняемый элемент S_k — входящая стрелка произведения по некоторому параметру i, то параметру i придается значение $i_{нач}$; если $i_{нач}$ — переменная величина, то ее значение определяется по функции определения и придается параметру i;

е) если выполняемый элемент S_k — выходящая стрелка произведения по некоторому параметру i, то по функциям определения вычисляются значения Δi и параметра i. После чего либо выполняется элемент S_{k+1}, если полученное значение параметра не принадлежит последовательности значений области его определения, либо, в противном случае, выполняется элемент, следующий за соответствующей входящей стрелкой произведения;

ж) если выполняемый элемент S_k — символ H, то после выполнения соответствующего нестандартного оператора выполняется элемент S_{k+1}.

III. Если выполняемый элемент—символ „останов“, то выполнение логической схемы окончено.

Примечание: Рассматриваются только те логические схемы, для которых символ „останов“ становится выполняемым элементом через конечное число применений правил I—III при любых $S \in T$.

Пусть L — логическая схема, M — множество входных величин всех операторов схемы (в том числе и нестандартных операторов, символы которых имеются в данной схеме) с областью определения T, и пусть I — множество всех параметров логической схемы. Пусть для каждого параметра множества I задана область задания этого параметра. Пусть A — оператор схемы L.

Входную величину оператора A назовем независимой, если эта входная величина не является выходной величиной никакого другого оператора логической схемы L, выполненного до первого выполнения оператора A при выполнении схемы L хотя бы над одним набором из области T. Вход оператора A назовем независимым, если все входные величины этого оператора независимые. Очевидно, что вход оператора схемы, выполняемого первым при каком-либо наборе из T, независим для данной схемы.

Входом логической схемы L назовем множество независимых входных величин операторов этой схемы (в том числе и нестандартных операторов, символы которых имеются в данной схе-

2—254

ме) и независимых входных величин операторов, реализующих функции определения параметров схемы.

Выходную величину оператора A назовем **независимой**, если она не является выходной величиной никакого другого оператора логической схемы L, выполняемого после выполнения оператора A при выполнении схемы L хотя бы над одним набором из области T.

Выходом логической схемы L назовем все независимые выходные величины операторов этой схемы (в том числе и нестандартных операторов, символы которых имеются в данной схеме) и независимые выходные величины операторов, реализующих функции определения параметров схемы.

Легко видеть, что выполнение логической схемы реализует отображение множества значений входных величин M на множество значений выходных величин N, где M и N — вход и выход логической схемы. Иными словами, выполнение логической схемы реализует выполнение некоторого оператора над набором значений входных величин схемы.

7. Пусть M_1 и M_2 — множества переменных величин, T_1 и T_2 — области определения множеств M_1 и M_2 соответственно. Будем говорить, что множество M_1 содержит множество M_2 ($M_1 \supseteq M_2$), если для каждой величины множества M_2 существует подобная ей величина множества M_1, и область определения множества подобных величин, рассматриваемых в M_1 (обозначим ее через T), содержит область определения множества M_2, т. е. $T \supseteq T_2$. Очевидно, что T_2 есть область определения множества подобных величин, рассматриваемых в множестве M_2. Если $M_1 \supseteq M_2$, то ясно, что для каждого набора $S \in T_2$ существует хотя бы один набор $\overline{S} \in T_1$, такой, что значения подобных величин, определяемые наборами S и $\overline{S}$, равны. Набор $\overline{S}$ будем называть **расширением** набора S. Ясно, что для каждого набора области T_2 может быть несколько расширений в области T_1.

Пусть R_1 и R_2 — логические схемы, выполняемые над областями определения множеств M_1 и M_2 и реализующие отображения $M_1 \to N_1$ и $M_2 \to N_2$ соответственно. Пусть $M_1 \supseteq M_2$ и $N_1 \supseteq N_2$. Пусть, далее, $P^{(2)}$ — произвольный набор области определения множества M_2, и пусть $p^{(2)}$ — набор области определения множества N_2, сопоставленный набору $P^{(2)}$ логической схемой R_2.

Если существует хотя бы одно расширение набора $P^{(2)}$ в области определения множества M_1 такое, что набор $p^{(1)}$, сопоставленный набору $P^{(1)}$ логической схемой R_1, представляет собою расширение набора $p^{(2)}$ в области определения множества N_1, то будем говорить, что логическая схема R_1 **охватывает** логическую схему R_2, и обозначать $R_1 \supseteq R_2$.

Логические схемы R_1 и R_2 назовем **эквивалентными** (и будем обозначать $R_1 \sim R_2$), если $R_1 \supseteq R_2$ и $R_2 \supseteq R_1$. Эквивалентность логических схем R_1 и R_2 означает, что $M_1 = M_2$, $N_1 = N_2$ и что отображения, реализуемые логическими схемами R_1 и R_2, каждому набору области определения множества величин, над которым выполняются схемы, сопоставляют один и тот же набор области определения множества выходных величин. Заметим, что при установлении последнего факта условие о существовании расширенного набора в определении охватываемости логических схем используется только в одну сторону.

8. Типы операторов.

I. Отображение

$$\{(l_1, m_1), (l_2, m_2)\} \to (l_2, m_1)$$

назовем элементарным оператором засылки. Произведение конечного числа элементарных операторов засылки назовем оператором засылки.

II. Элементарным арифметическим оператором назовем отображение

$$\{(l_1, m_1), (l_2, m_2)\} \to (l_3, m_3),$$

где $m_3 = m_1 \boxed{\times} m_2$, $\boxed{\times}$ — символ арифметической операции сложения, вычитания, умножения, деления. Произведение конечного числа элементарных арифметических операторов и операторов засылки назовем арифметическим оператором.

III. Логический оператор — определение логического оператора приведено на стр. 14.

IV. Пусть (l, m) — параметрическая операторная величина, L — область определения величины (l, m), параметризованная параметрами $i_1, i_2, \cdots, i_n$. Элементарным оператором формирования по параметрам $i_1, i_2, \cdots, i_n$ назовем отображение, которое по заданному набору значений параметров — $(i_1^0, i_2^0, \cdots, i_n^0)$ — сопоставляет соответствующее ему значение операторной величины (l, m_1), где $m_1 \in L$. Элементарный оператор формирования по параметрам $i_1, i_2, \cdots, i_n$ обозначим через $\Phi((l, m), i_1, i_2, \cdots, i_m)$.

Пусть имеем некоторое множество $M = \{(l_1, m_1), \cdots, (l_t, m_t)\}$ переменных операторных величин, области определения которых параметризованы параметрами $i_1, \cdots, i_n$. Отображение, которое по каждому набору значений параметров $i_1^0, i_2^0, \cdots, i_n^0$ сопоставляет соответствующие ему значения операторных величин множества M, назовем оператором формирования по параметрам $i_1, i_2, \cdots, i_m$ и обозначим через $\Phi(M, i_1, i_2, \cdots, i_n)$. Очевидно, что

$$\Phi(M, i_1, \cdots, i_n) = \prod_{j=1}^{t} \Phi((l_j, m_j), i_1, i_2, \cdots, i_n),$$

где $m_j \in L_j$, $j = 1, 2, \cdots, t$, L_j — область определения операторной величины (l_j, m_j).

V. Пусть (l, m) — переменная операторная величина, область определения которой L параметризована параметрами $i_1, i_2, \cdots, i_n$,

$(l, m^{(1)})$, $m^{(1)} \in L$ — значение операторной величины (l, m), соответст-
вующее набору $(i_1^{\circ}, \cdots, i_{k-1}^{\circ}, i_k^{\circ}, i_{k+1}^{\circ}, \cdots, i_n^{\circ})$. Элементарным
оператором переадресации по параметру i_k назовем отобра-
жение значения $(l, m^{(1)})$ операторной величины (l, m) и величины Δ
на значение операторной величины (l, m), соответствующее новому
набору $(i_1^{\circ}, \cdots, i_{k-1}^{\circ}, i_k^{(1)}, i_{k+1}^{\circ}, \cdots, i_n^{\circ})$, где $i_k^{(1)} = i_k^{\circ} + \Delta$. Элементарный
оператор переадресации по параметру i_k будем обозначать через
$F = ((l, m), \Delta, i_k)$.

Пусть имеем некоторое множество

$$M = \{(l_1, m_1), (l_2, m_2), \cdots, (l_t, m_t)\}$$

переменных операторных величин, области определения которых L_j,
$j = 1, 2, \cdots, t$ параметризованы параметрами $i_1, i_2, \cdots, i_n$, и пусть
$(l_j, m_j^{(1)})$, $j = 1, 2, \cdots, t$, $m_j^{(1)} \in L_j$ — значения операторных величин
множества M, соответствующие набору $(i_1^{\circ}, \cdots, i_{k-1}^{\circ}, i_k^{\circ}, i_{k+1}^{\circ}, \cdots, i_n^{\circ})$.
Произведение $\prod_{j=1}^{t} F((l_j, m_j), \Delta, i_k)$ элементарных операторов пере-
адресации по параметру i_k назовем оператором переадреса-
ции по параметру i_k и будем обозначать через $F(M, \Delta, i_k)$. Вели-
чину Δ назовем шагом переадресации.

VI. Пусть начальные значения параметров $i_1, i_2, \cdots, i_n$ — посто-
янные величины. Элементарный оператор формирования по парамет-
рам $i_1, i_2, \cdots, i_n$, в котором заданный набор значений параметров
составлен из их начальных значений, назовем элементарным опе-
ратором восстановления по параметрам $i_1, \cdots, i_n$ и обозначим
через $R((l, m), i_1, i_2, \cdots, i_n)$, где (l, m) — параметрическая оператор-
ная величина. Аналогично определяется оператор восстановления по
параметрам $i_1, \cdots, i_n$ — $R(M, i_1, \cdots, i_n)$, где M — некоторое множество
переменных операторных величин. Несмотря на то, что оператор вос-
становления определяется как частный случай оператора формирова-
ния, в дальнейшем оператором формирования будем называть только
те операторы формирования, которые не являются операторами вос-
становления.

VII. Пусть $\{(l, i_l^{(k)})\}$, $k = 1, \cdots, t$ — множество значений пара-
метра i. Элементарный арифметический оператор, представляющий
собой отображение

$$\{(l, i^{(k-1)}), (l_1, m)\} \rightarrow (l, i^{(k)}),$$

где $i^{(k)} = i^{(k-1)} + m$, назовем оператором изменения пара-
метра i и будем обозначать через $f(i)$.

VIII. Нестандартный оператор есть некоторый оператор, который
задан либо соответствующей ему последовательностью операторных
величин, либо представляет собою номер некоторой логической схемы,
которая в дальнейшем должна быть включена в заданную логичес-
кую схему. Символ нестандартного оператора будем обозначать че-
рез H.

9. Схема счета и схема программы.

Логическую схему

$$S_1\ S_2 \cdots S_n$$

назовем **схемой счета**, если множество операторов $\{A\}$, указанное в определении логической схемы, состоит только из операторов засылки и арифметических операторов, исключая операторы изменения параметра.

Пусть

$$S_1\ S_2 \cdots S_n$$

логическая схема, удовлетворяющая следующим условиям:

а) множество операторов $\{A\}$ состоит из операторов типа I—VIII;

б) если элемент S_k — оператор засылки $(i_{нач} \to i)$, где i—параметр (в дальнейшем указанный оператор будем называть оператором засылки начального значения параметра и обозначать через $L(i_{нач})$), а элемент S_{k+1} не является оператором восстановления или формирования, то элемент S_{k+1} — входящая стрелка произведения;

в) если элемент S_k — оператор восстановления, а элемент S_{k+1} не является оператором формирования, то элемент S_{k+1} — входящая стрелка произведения;

г) если элемент S_k — оператор формирования, то элемент S_{k+1} — входящая стрелка произведения;

д) если элемент S_k — логический оператор, осуществляющий проверку достижения параметром его конечного значения (например, логический оператор, который истинен, если $i = i_{кон}$, и ложен в противном случае (в дальнейшем указанный оператор будем называть логическим оператором проверки достижения параметром его конечного значения и обозначать через $P(i_{кон})$), или особый логический оператор, то элемент S_{k+1} — выходящая стрелка произведения;

е) если элемент S_k — оператор переадресации, то либо элемент S_{k+1} — логический оператор проверки достижения параметром его конечного значения или особый логический оператор, либо элемент S_{k+1} — выходящая стрелка произведения;

ж) если элемент S_k — оператор изменения параметра, а элемент S_{k+1} не является оператором переадресации, то элемент S_{k+1} — логический оператор проверки достижения параметром его конечного значения или особый логический оператор.

Класс логических схем, удовлетворяющих описанным условиям, обозначим через R. Каждую логическую схему $L \in R$, в которой: 1) стрелки произведения заменены стрелками управления так, что стрелки произведения с различными параметрами заменены стрелками управления с различными номерами, отличными от номеров стрелок управления схемы L;

2) каждый параметрический оператор заменен произвольным непараметрическим оператором Z^*, — назовем **схемой программы**.

В дальнейшем, чтобы каждый раз при построении схем программ не выполнять в логических схемах класса R указанную замену, в схемах программ стрелки с символом параметра будут сохраняться и рассматриваться как стрелки управления с номером, равным условному числу этого параметра**.

В дальнейшем будем предполагать, что в схемах счета значение параметра изменяется только при выполнении стрелок произведения, а в схемах программ — только при выполнении оператора засылки начального значения параметра и оператора изменения параметра.

Заметим, что выполнение операторов восстановления, формирования, переадресации приводит к изменению некоторых операторных последовательностей, вследствие чего при выполнении схемы программы меняются и соответствующие операторы схемы. Иными словами, схемы программ могут перерабатываться в процессе их выполнения.

10. Произведение по параметру k назовем **вложенным** произведением по отношению к параметру i, если входящая и выходящая стрелки произведения по параметру k расположены между входящей и выходящей стрелками произведения по параметру i. Произведение по параметру i назовем **внешним** произведением, параметр i — внешним параметром, а произведение по параметру k — внутренним произведением (в определении вложенного произведения параметры i и k равноправны).

Произведения по параметрам i и k назовем **пересекающимися**, если между $\downarrow^i$ и $\uparrow^i$ имеется такая $\uparrow^k$ (или $\downarrow^k$), для которой соответствующая $\downarrow^k$ находится левее $\downarrow^i$ (или $\uparrow^k$ находится правее $\downarrow^i$). Взаимное расположение стрелок пересекающихся произведений имеет вид:

$$\downarrow^k \ldots \downarrow^i \ldots \uparrow^k \ldots \uparrow^i \quad \text{или} \quad \downarrow^i \ldots \downarrow^k \ldots \uparrow^i \ldots \uparrow^k.$$

Старшими стрелками и старшим параметром будем называть:

а) для невложенного произведения по параметру i, в котором между $\downarrow^i$ и $\uparrow^i$ нет других стрелок произведения, — стрелки и параметр этого произведения;

б) для вложенных произведений, не содержащихся между входящей и выходящей стрелками произведения по некоторому парамет-

---

* Заметим, что при выполнении схемы программы перед выполнением оператора Z операторами формирования или восстановления оператор Z будет заменен определенным оператором, представляющим собой значение соответствующего параметрического оператора схемы счета.

** Условным числом величины (l, m) называется некоторое натуральное число, по которому однозначно определяется число l данной величины (l, m). Условным числом параметрической величины называется натуральное число, поставленное в соответствие этой величине.

ру — крайнюю левую входящую и крайнюю правую выходящую стрелки и их параметр;

в) для пересекающихся произведений — крайнюю левую входящую стрелку и соответствующий ей параметр.

В дальнейшем будем предполагать, что каждому оператору логической схемы и символу нестандартного оператора приписано некоторое натуральное число, называемое номером или **условным числом** оператора. Всем операторам и символам нестандартных операторов логической схемы приписываются отличные друг от друга номера, таким образом, что по условному числу однозначно определяется число l первой величины операторной последовательности, сопоставленной данному оператору.

В связи с этим, в операторах формирования, восстановления и переадресации вместо множества переменных операторных величин будут указываться номера соответствующих параметрических операторов. Кроме того, в схемах программ оператору Z будет приписываться номер заменяемого им параметрического оператора.

Далее, в § 2 и 3 в терминах введенных выше понятий кратко излагаются результаты Р. И. Подловченко [5], используемые в данной работе.

§ 2. Эквивалентные параметры и преобразование параметров

Решение задачи на машине начинается с составления схемы счета. Вся числовая информация задачи разбивается на множества величин, которые затем параметризуются. Ясно, что разбиение информации на множества величин и их параметризация связаны с алгоритмом решения задачи и с удобством составления схемы. В терминах введенных параметров составляются схема счета и схема программы данного алгоритма. Параметры, параметризующие числовую информацию данной задачи, будем называть **эквивалентными** относительно данной задачи.

В качестве примера рассмотрим умножение треугольных матриц n-го порядка. Числовую информацию составляют матрицы-сомножители A, B и матрица-произведение — C. Сопоставим элементам матриц A, B, C отличные друг от друга натуральные числа. Тогда каждую треугольную матрицу можно рассматривать как множество величин $\{(l, m)\}$, где m — элемент матрицы. В качестве параметров, параметризующих каждую матрицу, возьмем индексы строки и столбца матрицы. Матрицу A параметризуем параметрами k, i, $k=1,\cdots, n$; $i = 1,\cdots, k$, матрицу B — параметрами i, l, $i=1,\cdots, n$; $l=1,\cdots, i$, матрицу C — параметрами k, l, $k=1, 2,\cdots, n$; $l=1,\cdots, k$. Так как элемент матрицы определяется функцией параметризации этой матрицы по заданным значениям параметров, то вместо элементов матрицы будем рассматривать соответствующие функции параметризации. Иными словами, матрицам A, B, C сопоставим функции параметризации a_{il}, b_{il}, c_{il}. Тогда

$$c_{kl}=\sum_{i=1}^{n} a_{ki}\, b_{il}; \quad k, l=1, 2, \cdots, n.$$

Введем операторы:

$A_{kil}:\{a_{ki}, b_{il}\}\to f$ — вычисляет произведение значений функций a_{ki} и b_{il} и засылает его в величину f;

$D:\{f, d\}\to d$ — вычисляет сумму величин f и d и засылает ее в величину d;

$L_{kl}: d\to c_{kl}$ — величину d засылает в значение функции c_{kl}.

Алгоритм умножения треугольных матриц может быть записан в виде следующей схемы счета:

$$\downarrow^k \downarrow^l (0\to d)^{(N_1)} \downarrow^i A_{kil}^{(N_2)} D^{(N_3)} \uparrow^i L_{kl}^{(N_4)} \uparrow^l \uparrow^k \text{ ост.}$$

$$1\to k_{\text{нач}},\quad 1\to \Delta k,\quad n\to k_{\text{кон}},$$
$$1\to l_{\text{нач}},\quad 1\to \Delta l,\quad k\to l_{\text{кон}},$$
$$1\to i_{\text{нач}},\quad 1\to \Delta i,\quad k\to i_{\text{кон}},$$

где N_1, N_2, N_3, N_4 — номера операторов $(0\to d)$, A_{kil}, D, L_{kl}, или в виде следующей схемы программы, соответствующей указанной схеме счета:

$$R_1(N_2, k, i, l)\, R_2(N_4, k, l)\,(1\to k)\downarrow^k (1\to l)\downarrow^l (l\to i)\times$$
$$\times (0\to d)^{(N_1)}\downarrow^i Z^{(N_2)} D^{(N_3)} f(\Delta i, i)\, F(N_2, \Delta i, i)\, P(i>k)\uparrow^k (l-k\to \Delta_1)\times$$
$$\times F(N_2, \Delta_1, i)\, Z^{(N_4)} f(\Delta l, l)\, F(N_2, N_4, \Delta l, l)\, P(l>k)\uparrow^l (-k\to \Delta_2)\times$$
$$\times F(N_2, N_4, \Delta_2, l)\, f(\Delta k, k)\, F(N_2, N_4, \Delta k, k)\, P(k>n)\uparrow^k \text{ ост.},$$

где R_1 и R_2 — операторы восстановления параметрических операторов схемы счета A_{kil} и L_{kl}, $\downarrow^k$, $\uparrow^k$; $\downarrow^i$, $\uparrow^i$; $\downarrow^l$, $\uparrow^l$ — стрелки управления с номерами k, i, l соответственно.

Ясно, что параметризация множеств величин, составляющих числовую информацию задачи, не единственная. Так, в приведенном примере параметризация множеств A, B, C может быть осуществлена также параметрами k_1, i_1, k_2, i_2, k_3, i_3, где k_1, k_2, k_3 и i_1, i_2, i_3 — индексы строк и столбцов матриц A, B и C соответственно. Возникает вопрос о выборе системы параметров более удобной как с точки зрения записи алгоритма в данном языке, так и с точки зрения его машинной реализации. В связи с этим, большой интерес представляют преобразования при переходе от одной параметризации к другой.

Пусть вся информация задачи разбита на множества величин, которые параметризованы. Рассмотрим некоторое множество числовых величин M, параметризация которого осуществлена системой параметров $i_1, i_2, \cdots, i_n$. Пусть система параметров $j_1, j_2, \cdots, j_k$ осуществляет некоторую другую параметризацию множества величин M. Пусть I_n и J_k — области определения множеств $(i_1, i_2, \cdots, i_n)$ и $(j_1,$

$j_2, \cdots, j_k)$ соответственно. Между множествами I_n и J_k установим такое взаимно-однозначное соответствие, при котором соответствующие наборы значений параметров $(i_1, i_2, \cdots, i_n)$ и $(j_1, j_2, \cdots, j_k)$ определяют одну и ту же величину множества M. Пусть указанное соответствие выражается формулами

$$j_r = j_r(i_1, i_2, \cdots, i_n),\ r = 1, 2, \cdots, k,$$

$$i_t = i_t(j_1, j_2, \cdots, j_k),\ \ t = 1, 2, \cdots, n.$$

Рассмотрим, как отражается переход от одной параметризации к другой на элементах логической схемы. Ясно, что этот переход отражается только на параметрических операторах и на операторах над множеством операторных величин.

Пусть входная или выходная величина некоторого оператора логической схемы взята из параметризованного множества M. Тогда в логической схеме имеется оператор $A_{i_1 \cdots i_n i_{n+1} \cdots i_m}$ $m > n$. При другой параметризации множество M параметризовано параметрами $j_1, j_2, \cdots, j_k$, а потому вместо параметрического оператора $A_{i_1 \cdots i_n i_{n+1} \cdots i_m}$ в новой системе параметров будем иметь параметрический оператор $A_{j_1 \cdots j_k i_{n+1} \cdots i_m}$. Таким образом, параметры $i_1, \cdots, i_n$ в параметрическом операторе $A_{i_1 \cdots i_n \cdots i_m}$ $(m > n)$ заменяются параметрами $j_1, \cdots, j_k$, что приводит к замене в логической схеме параметрического оператора $A_{i_1 \cdots i_m}$ параметрическим оператором $A_{j_1 \cdots j_k i_{n+1} \cdots i_m}$.

Переход от одной параметризации к другой отражается на операторах переадресации, формирования, восстановления.

Рассмотрим оператор переадресации $F(N, \Delta i_{l_0}, i_{l_0})$, где N—номер переадресуемого оператора, l_0—целое число. Так как

$$j_r = j_r(i_1, i_2, \cdots, i_n), \quad r = 1, 2, \cdots, k,$$

то изменение параметра i_{l_0} на величину Δi_{l_0} приводит к приращению параметров $j_1, j_2, \cdots, j_k$ на величины, которые вычисляются как разности величин $j_r(i_1, \cdots, i_{l_0} + \Delta i_{l_0}, \cdots, i_n)$ и $j_r(i_1, \cdots, i_{l_0}, \cdots, i_n)$ по формулам

$$\Delta_{i_{l_0}} j_r = j_r(i_1, \cdots, i_{l_0} + \Delta i_{l_0}, \cdots, i_n) - j_r(i_1, \cdots, i_{l_0}, \cdots, i_n). \tag{1}$$

Из (1) и из определения оператора переадресации следует, что при переходе от параметров $i_1, \cdots, i_n$ к параметрам $j_1, \cdots, j_k$ оператор переадресации преобразуется по следующей формуле:

$$F(N, \Delta i_{l_0}, i_{l_0}) = \prod_{r=1}^{k} F(N, \Delta i_{l_0} j_r, j_r) \tag{2}$$

и, следовательно, при замене параметров достаточно вычислить величины $\Delta j_r\ (r = 1, 2, \cdots, k)$ по формуле (1) и произвести в схеме программы замену операторов переадресации по формуле (2).

Рассмотрим оператор формирования по параметрам $i_1, \cdots, i_m$ — $\Phi(N, i_1, \cdots, i_m)$, где N — номер формируемого оператора. При переходе от параметров $i_1, \cdots, i_n$ к параметрам $j_1, \cdots, j_k$ в силу формул

$$j_r^{(i)} = j_r(i_1, \cdots, i_n), \quad r = 1, 2, \cdots, k,$$

оператор формирования $\Phi(N, i_1, \cdots, i_m)$ заменяется оператором формирования $\Phi(N, j_1, \cdots, j_k, i_{n+1}, \cdots, i_m)$, $m > n$. В операторе восстановления при таком переходе параметры $i_1, \cdots, i_n$ заменяются параметрами $j_1, \cdots, j_k$.

Заметим, что новая параметризация может относиться сразу к нескольким множествам величин или ко всей информации задачи. В этом случае переход от одной параметризации к другой осуществляется так же, как и в случае введения новых параметров для одного множества. Можно проверить, что если в схеме программы произвести преобразования, связанные с переходом от одной параметризации к другой, то получим логическую схему, реализующую тот же оператор, что и исходная схема.

§ 3. Рабочая параметризация и рабочая схема программы

Пусть вся информация задачи разбита на множества числовых величин, которые параметризованы, и L—схема программы этой задачи. Рассмотрим конечное множество величин $M = \{(l, m)\}$.

Параметризацию множества M посредством одного параметра t, принимающего значения из отрезка натурального ряда, начиная с некоторого t_0, будем называть **рабочей** параметризацией.

Пусть схема программы L задана над множествами величин $\{M\}$. Для каждого параметризованного множества величин M введем рабочую параметризацию. Осуществляя в схеме L преобразования, связанные с переходом от одной параметризации к другой (с переходом от эквивалентных параметров к рабочим), получим логическую схему L', которую будем называть **рабочей схемой** программы.

Переход к рабочей схеме программы проиллюстрируем на примере умножения треугольных матриц. Пусть параметризация каждого множества величин A, B и C осуществляется посредством рабочих параметров r, s и t соответственно. Пусть заданы следующие формулы перехода от параметров i, k, l к параметрам r, s, t:

$$r = \frac{k(k-1)}{2} + i,$$

$$s = \frac{i(i-1)}{2} + l,$$

$$t = \frac{k(k-1)}{2} + l.$$

$$k = 1, 2, \cdots, n; \qquad l = 1, 2, \cdots, k; \quad i = l, \cdots, k.$$

Следовательно,

$$A = \{a_r\}, \quad B = \{b_s\}, \quad C = \{c_t\}.$$

В параметрическом операторе A_{kil} параметры k, i, l заменяются параметрами r, s, а в операторе L_{kl} параметры k, l заменяются параметром t. Переход к рабочим параметрам отражается также на операторах $R_1(N_2, k, i, l)$, $R_2(N_4, k, l)$ и операторах переадресации $F(N_2, \Delta i, i)$, $F(N_2, \Delta_1, i)$, $F(N_2, N_4, \Delta l, l)$, $F(N_2, N_4, \Delta_2, l)$, $F(N_2, N_4, \Delta k, k)$.

Согласно изложенному в § 2, произведем соответствующие преобразования этих операторов. Начальные значения параметров r, s, t определяются из соотношений:

$$r_{\text{нач}} = \frac{k(k-1)}{2} + i \Big|_{k=1,\ i=1} = 1,$$

$$s_{\text{нач}} = \frac{i(i-1)}{2} + l \Big|_{i=1,\ l=1} = 1,$$

$$t_{\text{нач}} = \frac{k(k-1)}{2} + l \Big|_{k=1,\ l=1} = 1.$$

Далее, согласно (2) из § 2

$$F(N, \Delta i, i) = F(N, \Delta_i r, r)\, F(N, \Delta_i s, s),$$

$$F(N, \Delta k, k) = F(N, \Delta_k r, r)\, F(N, \Delta_k t, t),$$

$$F(N, \Delta l, l) = F(\mathrm{N}, \Delta_l s, s)\, F(N, \Delta_l t, t),$$

где Δr, Δs, Δt вычисляются по формулам (1) из § 2.

Например,

$$\Delta_i s = s(i + \Delta i, l) - s(i, l) \Big|_{\Delta i = 1} = \frac{i(i+1)}{2} + $$

$$+ l - \frac{i(i-1)}{2} - l = i \to \overline{\Delta}_1,$$

$$\Delta_i r = r(k, i + \Delta i) - r(k, i) \Big|_{\Delta i = 1} = \frac{k(k-1)}{2} + $$

$$+ i + 1 - \frac{k(k-1)}{2} - i = 1 \to \overline{\Delta}_2$$

и

$$F(N_2, \Delta i, i) = F(N_2, \overline{\Delta}_1, r)\, F(N_2, \overline{\Delta}_2, s),$$

аналогично

$$F(N_2, \Delta_1, i) = F(N_2, \overline{\Delta}_3, r)\, F(N_2, \overline{\Delta}_4, s),$$

$$F(N_2, N_4, \Delta l, l) = F(N_2, N_4, \overline{\Delta}_5, s)\, F(N_2, N_4, \overline{\Delta}_6, t),$$

$$F(N_2, N_4, \Delta_2, l) = F(N_2, N_4, \overline{\Delta}_7, s)\, F(N_2, N_4, \overline{\Delta}_8, t),$$

$$F(N_2, N_4, \Delta k, k) = F(N_2, N_4, \overline{\Delta}_9, r)\, F(N_2, N_4, \overline{\Delta}_{10}, t),$$

$$\text{где} \quad (l-k) \to \overline{\Delta}_3; \quad \frac{(l-k)(2i+l-k-1)}{2} \to \overline{\Delta}_4; \; l \to \overline{\Delta}_5;$$

$$1 \to \overline{\Delta}_6; \quad \frac{k(k-2)}{2} \to \overline{\Delta}_7; \quad (-k) \to \overline{\Delta}_8; \quad k \to \overline{\Delta}_9); \quad k \to \overline{\Delta}_{10}.$$

Эти преобразования приводят к следующей рабочей схеме программы для умножения треугольных матриц:

$$R_1(N_2, r, s)\, R_2(N_4, t)\,(1 \to k) \downarrow^k (1 \to l) \downarrow^l (l \to i)\,(0 \to d)^{(N_3)} \downarrow^i Z^{(N_3)} \times$$

$$\times D^{(N_3)} f(\Delta i, i)\, F(N_2, \overline{\Delta}_2, r)\,(i \to \overline{\Delta}_1)\, F(N_2, \overline{\Delta}_1, s) \times$$

$$\times P(i > k) \uparrow^k (l-k \to \overline{\Delta}_3)\, F(N_2, \overline{\Delta}_3, r) \left(\frac{(l-k)(2i+l-k-1)}{2} \to \overline{\Delta}_4 \right) \times$$

$$\times F(N_2, \overline{\Delta}_4, s)\, Z^{(N_3)} f(\Delta l, l)\,(l \to \overline{\Delta}_5)\, F(N_2, N_4, \overline{\Delta}_5, s)\, F(N_2, N_4, \overline{\Delta}_6, t) \times$$

$$\times P(l > k) \uparrow^l \left(\frac{k(k-2)}{2} \to \overline{\Delta}_7 \right) F(N_2, N_4, \overline{\Delta}_7, s)\,(-k \to \overline{\Delta}_8) \times$$

$$\times F(N_2, N_4, \overline{\Delta}_8, t)\, f(\Delta k, k)\,(k \to \overline{\Delta}_9)\, F(N_2, N_4, \overline{\Delta}_9, r)\,(k \to \overline{\Delta}_{10}) \times$$

$$\times F(N_2, N_4, \overline{\Delta}_{10}, t)\, P(k > n) \uparrow^k \text{ост.}$$

Заметим, что если несколько рабочих параметров имеют одинаковые начальные значения и получают одинаковые приращения при одних и тех же приращениях одних и тех же эквивалентных параметров, то для простоты записи рабочей схемы программы, совокупности указанных рабочих параметров можно сопоставить один рабочий параметр. Вышесказанное легко проверить на примере задачи Дирихле, рассмотренной в работе [3].

Необходимость перехода к рабочим параметрам и к рабочей схеме программы обуславливается следующими фактами:

а) рабочая параметризация соответствует линейному характеру памяти современных вычислительных машин;

б) из определения рабочего параметра и рабочей схемы программы следует, что все параметрические величины, содержащие символ одного и того же рабочего параметра, при переадресации по этому параметру имеют один и тот же шаг переадресации, тогда как параметрические величины, содержащие символ одного и того же эквивалентного параметра, при переадресации по этому параметру могут иметь различные шаги переадресации;

в) рабочая схема программы удобнее для выполнения преобразований логических схем, в результате которых получается эквивалентная логическая схема, реализующая более рационально алгоритм решения данной задачи.

Таким образом, построение логических схем, связанных с решением конкретной задачи, состоит из следующих этапов:

1) выбор численного метода решения данной задачи, параметризация числовой информации и построение схемы счета, реализующей этот метод;

2) построение схемы программы, соответствующей схеме счета;

3) введение рабочей параметризации и построение рабочей схемы программы;

4) преобразование рабочей схемы программы (построение эквивалентной логической схемы).

Глава II

ВХОДНАЯ ИНФОРМАЦИЯ ПП

§ 1. Описание входной информации ПП

В настоящем параграфе приводится подробное описание входной информации, обеспечивающей работу всех блоков ПП. В дальнейшем, при описании ПП, каждой логической схеме $\overline{S}_1\ \overline{S}_2 \cdots \overline{S}_n$ будем сопоставлять последовательность $S_1\ S_2 \cdots S_n$, в которой вместо каждого оператора схемы записано обозначение (символ) этого оператора. Последовательность $S_1\ S_2 \cdots S_n$ назовем кодом логической схемы (соответственно — кодом схемы счета, схемы программы).

Пусть $\overline{S}_1\ \overline{S}_2 \cdots \overline{S}_n$ — схема счета алгоритма решения данной задачи. Согласно определению схемы счета, каждый элемент $\overline{S}_i$, $i = 1, 2, \cdots, n$ может быть оператором типа арифметического, логического, стрелкой управления или произведения, символом H или „останов". Входная информация к ПП состоит из кода схемы счета — $S_1\ S_2 \cdots S_n$. К каждому элементу S_i, $i = 1, \cdots, n$, представляющему собой символ оператора, задается описание этого оператора. Если элемент S_i кода данной схемы счета представляет собой стрелку произведения по некоторому параметру, то к нему задаются формулы определения этого параметра. Для параметризованных множеств числовой информации данной задачи задается таблица символов параметрических величин и соответствующих рабочих параметров этих множеств. Для каждой параметрической величины задается связь между эквивалентными параметрами и рабочим параметром (формула рабочего параметра).

В качестве кода исходной схемы ПП может быть задан код схемы программы с информацией, аналогичной описанной выше. Информация к символам операторов восстановления, формирования, переадресации и изменения счетчика подробно описывается в § 2 настоящей главы.

Заметим, что в схеме программы функции определения параметров схемы реализуются операторами схемы программы.

Перейдем к определению описания оператора и к описанию клас-

са допустимых в данной ПП функций определения параметра, формул
рабочих параметров и таблицы параметрических величин.

Предварительно определим формулу счета. Введем:

1) элемент-символы постоянных, переменных и параметрических числовых величин*;

2) $\boxed{\times}$ — символ двуместной арифметической операции сложения, вычитания, умножения, деления;

3) Θ_i — символ i — местной псевдооперации (под i — местной псевдооперацией понимается вычисление некоторой i — местной функции, например, тригонометрических, корня, детерминанта ограниченного порядка и т. п.).

Формула счета определяется следующим образом:

1) элемент есть формула счета;

2) если U_1, U_2 — формулы счета, то $(U_1 \boxed{+} U_2)$ — формула счета;

3) если $U_1, U_2, \cdots, U_n$ — формулы счета, то

$$\Theta_i(U_{k_1}, \cdots, U_{k_i}), \ 1 \leqslant k_j \leqslant n; \ \ j = 1, 2, \cdots, i; \ \ i = 1, 2, \cdots, n$$

— формула счета.

Пусть $U_1, U_2, \cdots, U_k$ — формулы счета. Выражение вида

$$U_1 \to a^{(1)},$$
$$U_2 \to a^{(2)},$$
$$\cdots\cdots$$
$$U_k \to a^{(k)},$$

где $a^{(i)}$ — числовые величины, $\to$ — символ операции засылки, назовем описанием арифметического оператора, реализующего формулы счета $U_1, U_2, \cdots, U_k$. В частности, если U_i, $i = 1, 2, \cdots, k$ — элементы, то получаем описание оператора засылки.

Пусть U — формула счета. Выражение вида

$$U \boxed{+} 0,$$

где $\boxed{+}$ — логический знак (логическим знаком называем один из знаков $>, \geqslant, <, \leqslant, =, \neq$), 0 — нулевая величина, назовем описанием логического оператора.

Описание нестандартного оператора представляет собою либо последовательность операторных величин (машинных команд), сопоставленную данному нестандартному оператору, либо номер некоторой подсхемы**.

Пусть f_1, f_2, f_3 — формулы счета, в которых:

1) элементом могут быть только символы постоянных и переменных числовых величин;

---

* Класс допускаемых параметрических величин описывается на стр. 32 при описании таблицы параметрических величин.

** Номер подсхемы см. в гл. V.

2) символом операции может быть только символ двуместной арифметической операции сложения, вычитания, умножения;

3) результаты вычислений по формулам счета f_1, f_2, f_3 должны быть целочисленными величинами при любых исходных данных. Тогда

$$f_1 \to i_{\text{нач}},$$
$$f_2 \to \Delta i$$
$$f_3 \to i_{\text{кон}},$$

где $i_{\text{нач}}$, $i_{\text{кон}}$, Δi — символы переменных величин, именуемых в дальнейшем символами начального, конечного значения параметра и приращения соответственно, назовем **формулами определения эквивалентного параметра** i, или формулами определения величин $i_{\text{нач}}$, $i_{\text{кон}}$, Δi соответственно.

Опишем допустимые зависимости рабочих параметров от эквивалентных. Выражение вида

$$f \pm a_1 x_1^{(1)} x_2^{(1)} \cdots x_{k_1}^{(1)} P(A_1 \boxed{\pm} 0) \pm \cdots \pm a_n x_1^{(n)} x_2^{(n)} \cdots x_{k_n}^{(n)} \times$$
$$\times P(A_n \boxed{\pm} 0) \to s,$$

где f — формула счета, удовлетворяющая условиям 1) и 2) определения описания формул определения параметра, A_k, $k = 1, 2, \cdots, n$ — формулы счета, содержащие символ только двуместной арифметической операции сложения, вычитания, умножения, $x_1^{(p)}, \cdots, x_k^{(p)}$, $p = 1, 2, \cdots, n$ — символы переменных величин, a_p — символ постоянной величины, назовем **формулой рабочего параметра** s. При этом результат вычисления по формуле рабочего параметра должен быть целочисленной величиной.

В дальнейшем сомножитель вида $P(A \boxed{\pm} 0)$ в формуле рабочего параметра будем называть логическим условием, A — выражением логического условия. Формулы рабочего параметра и определения параметров будем приводить к виду, в котором для указания порядка выполнения операций скобки не используются, и задавать в полученном виде. Например, формула рабочего параметра

$$j(a_1 i + a_2 j - a_3 ik) \to s$$

задается в виде

$$a_1 ij + a_2 j^2 - a_3 ijk \to s.$$

Ограничения, наложенные на формулы определения эквивалентного параметра и формулы рабочих параметров, вызваны тем, что при практическом решении задач подавляющее большинство указанных формул имеет описанный вид. В связи с этим в данной работе рассматриваются только формулы описанного вида.

К схеме счета задается список символов параметрических величин, входящих в описание хотя бы одного оператора этой схемы или хотя бы в одно выражение логического условия формул рабочих параметров схемы. Для каждого символа параметрической величины списка указывается рабочий параметр и натуральное число, представляющее собой длину множества числовой информации, определяемого этой параметрической величиной. Индексами параметрических величин могут быть не только символы параметров, но и формулы счета вида $ai+b$, где a и b — символы постоянных величин, i — символ параметра. В дальнейшем список параметрических величин будем называть **таблицей параметрических величин.**

§ 2. О способе задания информации к ПП

В настоящем параграфе описывается способ задания информации ПП. В приводимом описании под величинами подразумеваются их условные числа*.

Код схемы счета $S_1 S_2 \cdots S_n$ задается в виде последовательности строк, описывающих каждый элемент схемы счета следующим образом:

I. Строка информации о символе каждого оператора схемы счета (такими могут быть только операторы арифметический, логический, засылки и нестандартный) содержит следующие указания:

1) указание о типе оператора;

2) номер оператора в схеме счета (по номеру оператора однозначно определяется описание этого оператора);

3) список параметров, если оператор параметрический;

4) указание о принадлежности оператора к схеме счета (последнее необходимо для типов операторов, которые могут быть не только элементами схемы счета, но и добавлены при построении схемы программы по заданной схеме счета. Такими операторами могут быть только оператор засылки и логический оператор);

5) для логического оператора — указание логического знака.

II. Строки о стрелках содержат следующую информацию:

1) тип стрелки (в схеме счета имеем стрелки двух типов — управления и произведения);

2) вид стрелки (различаем стрелки трех видов — входящие, выходящие и стрелки с возвратом**);

3) номер стрелки, если она является стрелкой управления, или параметр стрелки, если она является стрелкой произведения.

Для стрелок произведения дополнительно задается:

4) для входящей стрелки произведения—символ начального значения параметра;

5) для выходящей стрелки произведения — символы приращения и конечного значения параметра; если конечное значение параметра неизвестно, то символ конечного значения параметра не указывается.

---

* См. примечание к стр. 22.

** Стрелка управления с возвратом будет описана в главе V.

Описание арифметического оператора схемы счета и формулы счета описания логического оператора задаются так же, как и в работе [8].

Таблица параметрических величин представляет собой последовательность строк, в каждой из которых записаны эквивалентные параметры данной параметрической величины, рабочий параметр и натуральное число, представляющее собой длину множества числовых величин, определяемого этой параметрической величиной. Каждой параметрической величине отводится одна строка. Например, для параметрической величины a_{ijk}, определяющей множество числовых величин, параметризованное эквивалентными параметрами i, j, k и рабочим параметром s, строка в таблице параметрических величин имеет вид

$$<a_{ijk}> \; i \; j \; k \; s \; d,$$

где $<a_{ijk}>$ — условное число параметрической величины a_{ijk}, а i, j, k, s, d — условные числа параметров и длины соответственно.

Если индекс параметрической величины представляет собой формулу счета вида $a \cdot i + b$, то в строке, соответствующей этой параметрической величине, вместо индекса указывается некоторое условное число. Последним определяется информация о формуле счета. Информация о формуле счета задается так же, как и информация о формуле рабочего параметра. Например, для параметрической величины $c_{i\; aj+b}$ строка в таблице параметрических величин имеет вид

$$<c_{i\; aj+b}> \; i \; r \; s \; d,$$

где по условному числу r определяется информация о формуле $aj + b$.

Информация о формуле рабочего параметра задается в виде строк, в первой из которых записывается условное число этого рабочего параметра, а в остальных — члены формулы. В последних для каждого члена формулы записаны последовательно коэффициент (коэффициентом называем постоянную величину, входящую в данный член формулы) и переменные величины. Заметим, что условным числом параметра задается либо сам параметр, либо некоторая формула, поставленная в соответствие этому параметру. Если в формуле рабочего параметра содержится n-я степень некоторой величины, то эта величина записывается в строке n раз. Для каждого члена формулы рабочего параметра отводится одна строка, знак члена приписывается строке. Информация о формуле рабочего параметра содержит признак окончания информации о рабочем параметре. Если член формулы рабочего параметра содержит логическое условие, то информация о логическом условии записывается в некоторой группе строк в следующем виде: в первой строке информации о логическом условии записываются последовательно коэффициент, переменные при логическом условии и логический знак данного логического условия. В следующих строках записывается выражение логического условия. Для каждого члена выражения отводится одна строка. Строки информации о выражении логического условия составляются так же, как и строки информации о формуле рабочего параметра. При этом, если член выражения логического условия содержит параметрическую величину, то в строке информации об этом члене записывается условное число этой параметрической величины. В указанной группе строк информации содержится признак окончания информации о логическом условии.

Например, информация о формулах рабочих параметров s_1 и s_2

$$j(a_1 i + a_2 j - a_3 i k) \to s_1,$$
$$a[jP(i + j - n = 0) + bi^3] \to s_2,$$

заданных в виде

$$a_1 ij + a_2 j^2 - a_3 ijk \to s_1,$$
$$ajP(i + j - n = 0) + abi^3 \to s_2,$$

записывается в виде следующих строк информации:

$$\begin{array}{l} \quad\; s_1 \\ + \langle a_1 \rangle\; i\; j \\ + \langle a_2 \rangle\; j\; j \\ - \langle a_3 \rangle\; i\; k\; j \\ \quad\; \Omega \\[6pt] \quad\; s_2 \\ + \langle a \rangle\; j\; = \\ + \langle 1 \rangle\; i \\ + \langle 1 \rangle\; j \\ - \langle n \rangle \\ \quad\; W \\ + \langle ab \rangle\; i\; i\; i \\ \quad\; \Omega \end{array}$$

где Ω — признак окончания информации о рабочем параметре, W — признак окончания информации о логическом условии, символом $\langle a \rangle$ обозначено условное число величины a.

Формулы определения некоторого параметра записываются так же, как и формула рабочего параметра.

Если в качестве исходной схемы для работы ПП взята схема программы, то к описанной информации добавляется информация о символах операторов засылки начального значения параметра, восстановления, переадресации, формирования, изменения параметра и логического оператора проверки достижения параметром его конечного значения. Для символов указанных операторов задается указание о типе оператора, после чего задается следующая информация:

Для символа оператора засылки начального значения параметра:

1) параметр, начальное значение которого засылается;

2) символ начального значения этого параметра;

3) признак принадлежности к схеме программы.

Для символа оператора восстановления:

1) номера восстанавливаемых параметрических операторов;

2) список параметров.

Для символа оператора формирования:

1) номера формируемых параметрических операторов;

2) список параметров формирования.

Для символа логического оператора проверки достижения параметром его конечного значения:

1) параметр, конечное значение которого проверяется;

2) символ конечного значения этого параметра;

3) признак принадлежности к схеме программы.

Для символа оператора переадресации:

1) номера переадресуемых по этому параметру параметрических операторов;

2) шаг переадресации;

3) параметр переадресации.

Для символа оператора изменения параметра:

1) параметр, значение которого изменяется;

2) приращение этого параметра.

Ясно, что описанная информация о символах операторов схемы программы может быть построена по информации об элементах схемы счета.

Глава III

НЕКОТОРЫЕ ВОПРОСЫ ПОСТРОЕНИЯ И РЕАЛИЗАЦИИ ОПЕРАТОРА ФОРМИРОВАНИЯ

В данной ПП построение оператора формирования осуществляется в случаях, если начальное значение параметра—переменная величина. Однако, если начальное значение параметра—переменная величина, функция определения которой зависит только и только от внешних относительно данного параметров пересекающихся произведений, то оператор формирования можно не строить, так как его работу может выполнить оператор переадресации, если при вычислении шагов переадресации по соответствующим рабочим параметрам учесть эту зависимость.

Проиллюстрируем сказанное на примере следующей схемы счета:

$$\downarrow^{i} \downarrow^{j} A_{ij}^{(N_1)} \uparrow^{j} \uparrow^{i} \text{ ост.}$$

$$1 \to i_{\text{нач}}, \quad 1 \to \Delta i, \quad i \to j_{\text{нач}}, \quad 1 \to \Delta j,$$

$$n \to j_{\text{кон}}, \quad n \to i_{\text{кон}},$$

где N_1 — номер оператора A_{ij}.

Описание оператора A_{ij} имеет вид:

$$a_{ij} \times b_{ij} \to c_{ij}.$$

Таблица параметрических величин состоит из следующих строк:

$$\langle a_{ij} \rangle \; i \; j \; s \; d,$$
$$\langle b_{ij} \rangle \; i \; j \; s \; d,$$
$$\langle c_{ij} \rangle \; i \; j \; s \; d.$$

Начальное значение параметра j — переменная величина, формула определения которой содержит только внешний параметр i непересекающегося произведения, а потому по данной схеме счета можно построить следующую схему программы:

$$R(N_1, i, j)(1 \to i) \downarrow^{i} (i \to j) \downarrow^{j} A_{ij}^{(N_1)} f(\Delta j, j) F(N_1, \Delta j, j) P(j > n) \uparrow^{j} \times$$
$$\times [-(n+1-i) \to \Delta_1] F(N_1, \Delta_1, j) f(\Delta i, i) F(N_1, \Delta i, i) P(i > n) \uparrow^{i} \text{ ост.},$$

где оператор $R(N_1, i, j)$ осуществляет восстановление начального вида оператора A_{ij}, стрелки $\downarrow^{i}$, $\uparrow^{i}$, $\downarrow^{j}$, $\uparrow^{j}$ — стрелки управления, символ A_{ij} рассматривается как символ непараметрического оператора, на место которого оператор $R(N_1, i, j)$ при выполнении схемы передает начальный вид параметрического оператора A_{ij}. Оператор переадресации $F(N_1, \Delta_1, j)$ при выполнении схемы для каждого значения параметра i подготавливает вид оператора A_{ij} для выполнения произведения по параметру j. Легко проверить, что построенная схема программы не эквивалентна данной схеме счета, так как в ней не учтена зависимость $j_{нач} = f(i)$. Эту зависимость можно непосредственно учесть при переходе к рабочим параметрам. Перед вычислением приращения $\Delta_i s$ подставим в формулу рабочего параметра $j = i$. Тогда получим

$$s = (i-1)n + i,$$

после чего получим

$$\Delta_i s = s(i + \Delta i) - s(i) = (i - 1 + 1)n + i + 1 - (i-1)n - i = n + 1$$

и следующую рабочую схему программы:

$$R(N_1, s)(1 \to i) \downarrow^{i} (i \to j) \downarrow^{j} A_{ij}^{(N_1)} f(\Delta j, j)(1 \to \overline{\Delta}_1) F(N_1, \overline{\Delta}_1, s) P(j > n) \uparrow^{j} \times$$
$$\times [-(n+1-i) \to \Delta_1] F(N_1, \Delta_1, s) f(\Delta i, i)(n + 1 \to \overline{\Delta}_2) F(N_1, \overline{\Delta}_2, s) \times$$
$$\times P(i > n) \uparrow^{i} \quad \text{ост.}$$

Таким образом, не строя оператор формирования, но учитывая при переходе к рабочим параметрам зависимость $j = f(i)$, получаем рабочую схему программы, эквивалентную данной схеме счета.

Опишем построение оператора формирования для вложенных и пересекающихся произведений.

Пусть $i_1, i_2, \cdots, i_n$ — параметры, I_n — область определения величин $i_1, i_2, \cdots, i_n$. Пусть начальные значения некоторых параметров в некоторой схеме счета — переменные величины и пусть $A_{i_1 \cdots i_n}$ — параметрический оператор. Пусть для начальных значений параметров, являющихся переменными величинами, условие зависимости функций определения только и только от внешних параметров непересекающихся произведений не выполняется. Из начальных значений параметров $i_1, i_2, \cdots, i_n$ построим набор $(i_1^\circ, \cdots, i_n^\circ)$, причем, если начальное значение параметра — переменная величина, то берется произвольное значение из ее области определения. Зафиксируем этот набор. Для построенного набора $(i_1^\circ, \cdots, i_n^\circ) \in I_n$ по функции параметризации можно определить последовательность значений операторных величин S_0, сопоставленную оператору $A_{i_1^\circ \ldots i_n^\circ}$. Выбранные набор $(i_1^\circ, \cdots, i_n^\circ)$ и последовательность S_0 назовем соответственно **стандартным набором значений параметров** и **стандартным видом параметрического оператора** $A_{i_1 \cdots i_n}$.

Согласно определению схемы счета, параметрический оператор $A_{i_1 \cdots i_n}$ находится между $\downarrow^{i_k}$ и $\uparrow^{i_k}$ для всех $k = 1, 2, \cdots, n$. Так как при выполнении схемы счета начальное значение каждого параметра i_k задается при выполнении $\downarrow^{i_k}$, то, очевидно, что полный набор значений параметров $i_1, \cdots, i_n$ будет известен только после выполнения $\downarrow^{i_k}$ для всех $k = 1, 2, \cdots, n$. Следовательно, оператор формирования может быть построен после задания параметрам их начальных значений. Так, например, если начальные значения параметров $i_1, \cdots, i_n$ — переменные величины, то отрезку схемы счета, представляющему собой систему вложенных произведений по параметрам i_k, $k = 1, 2, \cdots, n$,

$$\downarrow^{i_1} A_{i_1}^{(N_1)} \downarrow^{i_2} A_{i_1 i_2}^{(N_2)} \cdots \downarrow^{i_n} A_{i_1 i_2 \cdots i_n}^{(N_n)} \uparrow^{i_n} \cdots \uparrow^{i_2} \uparrow^{i_1}$$

сопоставляется следующий, эквивалентный ему отрезок схемы программы:

$$L^{(1)} \Phi (N_1, i_1) \downarrow^{i_1} A_{i_1}^{(N_1)} L^{(2)} \Phi (N_2, i_1, i_2) \downarrow^{i_2} A_{i_1 i_2}^{(N_2)} \cdots L^{(n)} \Phi (N_n, i_1, \cdots, i_n) \times$$
$$\times \downarrow^{i_n} A_{i_1 i_2 \cdots i_n}^{(Nn)} f(\Delta_n, i_n) \tag{3}$$
$$F(N_n, \Delta_n, i_n) P_n \uparrow^{i_n} (c_n \to \overline{\Delta}_n) F(N_n, \overline{\Delta}_n, i_n) \cdots \uparrow^{i_2} (c_2 \to \overline{\Delta}_2) \times$$
$$\times F(N_2, \cdots, N_n, \overline{\Delta}_2, i_2) f(\Delta_1, i_1) F(N_1, \cdots, N_n, \Delta_1, i_1) P_1 \uparrow^{i_1},$$

где операторы $L^{(k)}$, $k = 1, 2, \cdots, n$ — осуществляют вычисление и задание начального значения соответствующего параметра i_k, $k=1, 2, \cdots, n$, $\downarrow^{i_k}$ и $\uparrow^{i_k}$, $k=1, 2, \cdots, n$ — стрелки управления, символы $A_{i_1}, A_{i_1 i_2}, \cdots, A_{i_1 \cdots i_n}$ — рассматриваются как символы непараметрических операторов, $N_j \neq 1, 2, \cdots, n$, $j = 1, 2, \cdots, n$.

Пусть имеем теперь схему счета с пересекающимися произведениями

$$\cdots A^{(N_1)} \downarrow^{i} B_i^{(N_2)} \downarrow^{k} C_{ik}^{(N_3)} \uparrow^{i} D_k^{(N_4)} \uparrow^{k} E^{(N_5)} \cdots$$

Тогда построение оператора формирования осуществляется как перед выполнением произведения, так и в самом произведении. Если $k_{нач}$ — переменная величина, формула определения которой не содержит параметр i и величины, являющиеся выходными величинами оператора $C^{(N_3)}_{ik}$, то оператор формирования строится перед входящей стрелкой произведения по параметру k для всех операторов с параметром k, расположенных между входящей и выходящей стрелками произведения по этому параметру. Если же $k_{нач}$ — переменная величина, формула определения которой содержит параметр i или выходные величины оператора $C^{(N_3)}_{ik}$, то перед входящей срелкой произведения по параметру k строится оператор формирования по параметру k для всех параметрических операторов с параметром k, заключенных между входящей стрелкой произведения по этому параметру и выходящей стрелкой произведения по параметру i. После выходящей стрелки произведения по параметру i строится оператор формирования по параметру k для параметрических операторов с параметром k, расположенных между входящей и выходящей стрелками произведения по параметру k.

Так, в случае, если $k_{нач}$ — переменная величина, формула определения которой содержит параметр i или выходные величины оператора $C^{(N_3)}_{ik}$, указанному выше отрезку схемы счета с пересекающимися произведениями сопоставляется следующий, эквивалентный ему, отрезок схемы программы:

$$\cdots A^{(N_1)}(i_{нач} \to i)\, R(N_2,\ i) \downarrow^{i} B^{(N_2)}_{i}(k_{нач} \to k)\, \Phi(N_3,\ i,\ k) \downarrow^{k} C^{(N_3)}_{ik} \times$$

$$\times f(\Delta i,\ i)\, F(N_3,\ \Delta i,\ i) \times$$

$$\times P(i_{кон}) \uparrow^{i} \Phi(N_3,\ N_4,\ k,\ i)\, D^{(N_4)}_{k} f(\Delta k,\ k)\, F(N_3,\ N_4,\ \Delta k,\ k) \times$$

$$\times P(k_{кон}) \uparrow^{k} E^{(N_5)}.$$

Перейдем к рассмотрению вопроса о программной реализации оператора формирования. Пусть задана, например, схема программы вида (3). В силу того, что:

1) оператор $A_{i_1 \cdots i_n}$ расположен между стрелками $\downarrow^{i_k}$ и $\uparrow^{i_k}$, $k = 1,\ 2, \cdots,\ n$;

2) формула определения начального значения каждого параметра i_k, $k = 1, \cdots,\ n$ может содержать параметры только внешние для данного параметра;

3) значения каждого параметра i_k задаются при выполнении входящей стрелки произведения по этому параметру и не меняются в отрезке логической схемы, заключенном между $\downarrow^{i_k}$ и $\uparrow^{i_k}$, $k = 1, 2, \cdots,\ n$.

Схеме (3) можно сопоставить схему вида:

$$R(N_1,\ N_2, \cdots,\ N_n,\ i_1, \cdots,\ i_n)\, \overline{L}^{(1)} F(N_1, \cdots, N_n,\ \Delta i_1,\ i_1) \downarrow^{i_1} A^{(N_1)}_{i_1} \overline{L}^{(2)} \times$$

$$\times F(N_2, \cdots, N_n, \Delta i_2,\ i_2) \downarrow^{i_2} A^{(N_2)}_{i_1 i_2} \cdots\ \overline{L}^{(n)} F(N_n,\ \Delta i_n,\ i_n) \downarrow^{i_n} A^{(N_n)}_{i_1 \cdots i_n} \times$$

$$\times f(\overline{\Delta}_n,\ i_n)\, F(N_n,\ \overline{\Delta}_n,\ i_n)\, P(i_n) \uparrow^{i_n} (c_n \to \overline{\Delta}_n) \times \qquad (4)$$

$$\times F(N_n, \widetilde{\Delta}_n, i_n) \cdots \uparrow^{i_2} (c_2 \to \widetilde{\Delta}_2) F(N_2, \cdots, N_1, \widetilde{\Delta}_2, i_2) \times$$

$$\times f(\overline{\Delta}_1, i_1) F(N_1, \cdots, N_n, \overline{\Delta}_1, i_1) P(i_1) \uparrow^{i_1},$$

где $\downarrow^{i_k}$, $\uparrow^{i_k}$, $k=1, 2, \cdots, n$ — стрелки управления, символы A_{i_1}, $A_{i_1 i_2}, \cdots, A_{i_1 \cdots i_n}$ рассматриваются как символы непараметрических операторов, оператор R осуществляет задание стандартного вида операторов A_{i_1}, $A_{i_1 i_2}, \cdots, A_{i_1 \cdots i_n}$, операторы $\overline{L}^{(k)}$, $k = 1, 2, \cdots, n$ осуществляют вычисление величин $i_k^{(1)}$ и $i_k^{(1)} - i_k^{(0)} \to \Delta i_k$, где $i_k^{(0)}$ — значение параметра i_k из стандартного набора, $\widetilde{\Delta i}_k$ — шаги переадресации, обеспечивающие получение вида операторов, указанных в операторе $F(N, \widetilde{\Delta i}_k, i_k)$, соответствующего значению параметра i_k из стандартного набора.

Полученная схема не эквивалентна в определенном в данной работе смысле схеме (3), так как во вход схемы (4) кроме входных величин схемы (3) входят также постоянные величины стандартного набора, сопоставленные параметрам с переменным начальным значением. Легко убедиться, что выход схемы (4) не зависит от выбора стандартного набора, в силу чего для любого набора значений входных величин схемы (3), независимо от выбора стандартного набора, схемы (4) и (3) выдают один и тот же набор значений выходных величин. Аналогично, для каждой схемы с пересекающимися произведениями, содержащей оператор формирования, может быть сопоставлена схема, содержащая операторы восстановления и переадресации, и реализующая тот же оператор, что и данная схема.

Таким образом, оператор формирования, содержащийся в схеме программы, можно реализовать по следующей схеме:

1) задание стандартного вида формируемых операторов, соответствующего выбранному и зафиксированному стандартному набору параметров (задание указанного вида операторов осуществляется перед выполнением старшей стрелки произведения);

2) вычисление формул $i_k^{(1)} - i_k^{(0)} \to \Delta i_k$ перед выполнением стрелки управления $\downarrow^{i_k}$, где $i_k^{(1)}$ — текущее значение параметра i_k, а $i_k^{(0)}$ — значение этого параметра из стандартного набора;

3) выполнение оператора переадресации по параметру i_k с шагом переадресации Δi_k перед выполнением стрелки управления $\downarrow^{i_k}$ для всех формируемых операторов с параметром k.

По описанной схеме оператор формирования будет реализовываться при программировании оператора формирования. В записи схем программ будет использоваться символ оператора формирования.

Հ. Գ. ՓԻԼԻՊՈՍՅԱՆ, Ս. Մ. ԴԱՎԹՅԱՆ

ՏՐԱՄԱԲԱՆԱԿԱՆ ՍԽԵՄԱՆԵՐԻ ՄԻ ՔԱՆԻ ՁԵՎԱՓՈԽՈՒԹՅՈՒՆՆԵՐ ԿԱՏԱՐՈՂ ՈՒՆԻՎԵՐՍԱԼ ԾՐԱԳՐՈՂ ԾՐԱԳՐԻ ՆԿԱՐԱԳՐՈՒԹՅՈՒՆԸ

Ամփոփում

Այս աշխատության մեջ տրվում է ծրագրման Ա. Ա. Լյապունովի օպերատորային լեզվի վրա հիմնված ծրագրող ծրագրի նկարագրությունը։ Ի տարբերություն մինչև այժմ հայտնի ծրագրող ծրագրերից, վերոհիշյալը կատարում է տրամաբանական սխեմաների որոշ ձևափոխություններ, որոնց հետևանքով ստացվում է տրված տրամաբանական սխեմային համապատասխանող համեմատաբար ռացիոնալ ծրագիր։

Այս ժողովածուի մեջ տպագրվում է այս աշխատության առաջին երեք գլուխները։

Առաջին գլխում շարադրվում են ծրագրող ծրագրի տեսական հիմունքները։ Երկրորդ գլխում նկարագրվում է ծրագրող ծրագրի մուտքային լեզուն և որոշվում ծրագրող ծրագրով ծրագրվող տրամաբանական սխեմաների դասը։ Երրորդ գլխում տրվում է կազմավորման օպերատորի կառուցման գծերը և իրագործման ձևը։

ЛИТЕРАТУРА

1. *Ляпунов А. А.* О логических схемах программ, сб. „Проблемы кибернетики", вып. I, М., Физматгиз, 1958.
2. *Ляпунов А. А.* К алгебраической трактовке программирования, сб. „Проблемы кибернетики", вып. 8, М., Физматгиз, 1962.
3. *Подлоеченко Р. И.* Об основных понятиях программирования I, сб. „Проблемы кибернетики", вып. I, М., Физматгиз, 1958.
4. *Янов Ю. И.* О логических схемах алгоритмов, сб. „Проблемы кибернетики", вып. I, М. Физматгиз, 1958.
5. *Подловченко Р. И.* Об основных понятиях программирования II, сб. „Проблемы кибернетики", вып. 3, М., Физматгиз, 1960.
6. *Подловченко Р И.* О преобразованиях схем программ и их применении в программировании, сб. „Проблемы кибернетики", вып. 7, М., Физматгиз, 1962.
7. *Ершов А. П.* Программирующая программа для быстродействующей электронной счетной машины, Изд. АН СССР, 1958.
8. Система автоматизации программирования, сб. под ред. Н. П. Трифонова, М. Р. Шура-Бура, М., Физматгиз, 1961.
9. *Камынин С. С., Любимский Э. З., Шура-Бура М. Р.* Об автоматизации программирования при помощи программирующей программы, сб. „Проблемы кибернетики", вып. 1, М., Физматгиз, 1958.