

А. Г. ПИЛИПОСЯН, С. М. ДАВТЯН

О РЕАЛИЗАЦИИ ФУНКЦИЙ ИНДЕКСАЦИИ В ОПЕРАТОРНОМ ЯЗЫКЕ

В последних конструкциях электронных вычислительных машин для осуществления переадресации вводятся адресные регистры, называемые также индексными регистрами. Введение индексных регистров значительно упрощает программирование действий над адресами. В связи с этим представляет интерес разработка способов записи логических схем решаемой задачи для машин с индексными регистрами.

Данная заметка посвящена вопросу осуществления функций индексации в операторном языке, описанном авторами в работе [2].

§ 1. Оператор индексации и определения основных понятий операторного программирования

В настоящем параграфе в операторный язык вводится оператор нового типа—оператор индексации,—и даются определения только тех основных понятий операторного языка, данных в работе [2], которые подлежат изменению. Необходимость несколько изменить некоторые определения вызвана введением в операторный язык оператора индексации и исключением, вследствие этого, операторов формирования, восстановления, переадресации, изменения параметра и логического оператора проверки достижения параметром его конечного значения*.

В отличие от операторного языка, приведенного в работе [2] и называемого в дальнейшем операторным языком I, построенный в данной работе язык для описания алгоритмов решаемых задач будем называть операторным языком II.

Пусть i —параметр, $I = \{i^{(k)}\}$, $k=1, 2, \dots, N$ —множество значений параметра i , Δi —приращение этого параметра, $\{\Delta i^{(k)}\}$ —множество значений величины Δi и σ —некоторый определенный предикат,

* Определения указанных операторов даны в работе [2].

заданный на множестве натуральных чисел и ложный только на множестве I .

Пусть, далее, x — некоторая переменная величина и $\{0,1\}$ — множество значений величины x .

Отображение значений величин i и Δi на значения величин i и x такое, что $i^{(k+1)} = i^{(k)} + \Delta i^{(k)}$, $k = 1, 2, \dots, N$ и $x = 0$, если $i^{(k+1)} \in I$ (предикат σ ложен) и $x = 1$, если $i^{(k+1)} \notin I$ (предикат σ истинен) назовем оператором индексации по параметру i и будем обозначать через $Y(\Delta i, i, \sigma)$. Величину Δi назовем шагом индексации, i — параметром индексации.

Приведем определения некоторых основных понятий программирования в операторном языке II, отличающихся от соответствующих определений операторного языка I.

2. Пусть даны следующие множества:

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

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

в) множество операторов индексации $\{Y(\Delta i, i, \sigma)\}$;

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

д) множество пар стрелок произведения $\{\downarrow^I, \uparrow^I\}$;

е) символы H^{**} и „останов“.

Пусть

$$S_1 S_2 \dots S_n \quad (1)$$

последовательность, состоящая из конечного числа элементов, каждый из которых принадлежит одному из указанных множеств. На структуру последовательности (1) 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;

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

г) если в последовательности (1) элемент S_k — оператор индексации, то элемент S_{k+1} — выходящая стрелка управления;

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

е) если в последовательности (1) имеются стрелки произведения по некоторому параметру, то все параметрические операторы с символом этого параметра расположены между данными входящей и выходящей стрелками произведения.

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

** Символ H представляет собою символ нестандартного оператора.

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

е) если в последовательности (1) элемент S_k — параметрический оператор, не расположенный между стрелками произведения по параметрам этого параметрического оператора, то перед выполнением элемента S_k выполняются операторы, придающие значения параметрам данного параметрического оператора;

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

Последовательность (1), удовлетворяющую указанным условиям, назовем логической схемой.

Таким образом, данное определение логической схемы отличается от определения логической схемы операторного языка I ограничениями г) и е) на структуру последовательности (1).

С введением оператора индексации в определении выполнения логической схемы операторного языка I добавляется следующее правило: если выполняемый элемент логической схемы S_k — оператор индексации, то после его выполнения выполняется либо элемент S_{k+1} , если значение выходной величины x оператора индексации (см. определение оператора индексации) равно нулю, либо элемент S_{k+2} , если значение этой выходной величины равно единице.

3. Логическую схему

$$S_1 S_2 \dots S_n \quad (2)$$

назовем схемой счета, если из множеств, указанных в определении логической схемы, исключено множество $\{V(\Delta i, i, z)\}$ — множество операторов индексации, и всякий параметрический оператор находится между стрелками произведения по параметрам этого оператора.

Логическую схему (2), для которой выполнены следующие условия:

а) если элемент S_k — оператор засылки начального значения параметра, то элемент S_{k+1} — входящая стрелка управления;

б) если элемент S_k — оператор индексации, то элемент S_{k+1} — выходящая стрелка управления, соответствующая входящая стрелка которой расположена непосредственно за оператором засылки начального значения параметра индексации;

в) выходящая стрелка управления оператора индексации расположена правее от соответствующей ей входящей стрелки, — назовем схемой программы.

Как известно, схема программы представляет собой логическую схему, учитывающую некоторые особенности вычислительных машин для реализации на них схемы счета. Иными словами, схема программы может рассматриваться как промежуточный этап при переходе от схемы счета к языку машины. Ясно, что введенный оператор индексации, так же как и операторы II рода (операторы восстановления, формирования, переадресации, изменения параметра и логический опе-

ратор проверки достижения параметром его конечного значения) операторного языка I, обеспечивает реализацию схемы программы на вычислительной машине и может быть охарактеризован как оператор организации счета.

Заметим, что выполнение в операторном языке II схемы программы $\bar{L}$, соответствующей некоторой схеме счета и содержащей оператор индексации, сводится к выполнению в операторном языке I схемы программы L , соответствующей этой же схеме счета и содержащей операторы восстановления, формирования, переадресации, изменения параметра и логического оператора проверки достижения параметром его конечного значения.

Рассмотрим вид схемы $\bar{L}$ в операторном языке II в случае отсутствия в схеме L в произведении операторов переадресации, изменения параметра и логического оператора проверки достижения параметром его конечного значения одного из операторов этого произведения. Так, легко убедиться, что если в логической схеме L отсутствует оператор переадресации по некоторому параметру, то в логической схеме $\bar{L}$ нет параметрических операторов по этому параметру. Если в схеме L отсутствует оператор изменения некоторого параметра, то, следовательно, отсутствует и логический оператор проверки достижения параметром его конечного значения, и в логической схеме $\bar{L}$ в операторном языке II нет оператора индексации по этому параметру. При отсутствии в схеме L логического оператора проверки достижения параметром его конечного значения в соответствующем операторе индексации схемы $\bar{L}$ высказывание σ должно быть тождественно истинным (всегда выполняется элемент схемы $\bar{L}$, непосредственно следующий за выходящей стрелкой управления оператора индексации).

Кроме того, заметим, что в операторном языке II, в силу сохранения в схемах программ параметрических операторов, функций операторов восстановления, формирования и переадресации заменяются получением значений параметров.

§ 2. Построение схемы программы по заданной схеме счета

Каждой схеме счета алгоритма решения некоторой задачи соответствует схема программы этого же алгоритма, получаемая дополнением схемы счета операторами, организующими реализацию данной схемы счета на вычислительной машине. В операторном языке II такими операторами являются оператор индексации и оператор засылки начального значения параметра индексации.

При построении схемы программы по заданной схеме счета последовательно рассматриваются элементы схемы счета, и для каждого элемента выполняются соответствующие из следующих правил:

I. Если рассматриваемый элемент S_k — оператор арифметический, засылки, символ нестандартного оператора, не являющегося номером

некоторой подсхемы, стрелка управления или символ „останов“, то этот элемент передается в схему программы.

II. Если рассматриваемый элемент S_k — символ нестандартного оператора — номер стандартной подсхемы, то эта подсхема включается в схему программы*.

III. Если рассматриваемый элемент S_k — логический оператор, то он передается в схему программы; если же элемент S_k — особый логический оператор**, то этот элемент запоминается до выполнения всех построений, связанных с выходящей стрелкой произведения, которая всегда непосредственно следует за особым логическим оператором.

IV. Пусть рассматриваемый элемент S_k — входящая стрелка произведения по параметру i :

1) если $i_{нач}$ — переменная величина, то в схеме программы строится арифметический оператор, вычисляющий по формуле определения*** начальное значение параметра i ;

2) если $i_{кон}$ — переменная величина, формула определения которой не содержит величин, вычисляемых в произведении по параметру i , то в схеме программы строится арифметический оператор, вычисляющий величину $i_{кон}$ по ее формуле определения;

3) если Δi — переменная величина, формула определения которой содержит внешние параметры непересекающихся произведений и не содержит величин, вычисляемых в произведении по параметру i , или не содержит внешних параметров пересекающихся произведений и величин, вычисляемых в произведении по параметру i , то в схеме программы строится арифметический оператор, вычисляющий величину Δi по ее формуле определения;

4) в схеме программы строится оператор засылки начального значения параметра i ;

5) в схему программы передается входящая стрелка управления с номером, отличным как от номеров входящих стрелок управления данной схемы счета, так и от номеров входящих стрелок управления построенного отрезка схемы программы.

V. Пусть рассматриваемый элемент S_k — выходящая стрелка произведения по параметру i :

1) если Δi — переменная величина, формула определения которой содержит внешние параметры пересекающихся произведений, или содержит величины, вычисляемые в произведении по параметру i , то в схеме программы строится арифметический оператор, вычисляющий величину Δi по ее формуле определения;

* Правила включения подсхем описаны в § 2, гл. V работы [3].

** Особый логический оператор задается в схеме счета, перед выходящей стрелкой произведения по некоторому параметру, если конечное значение этого параметра неизвестно.

*** Формулы определения величин $i_{нач}$, Δi и $i_{кон}$ определяются в гл. II работы [2].

2) если $i_{\text{кон}}$ — переменная величина, формула определения которой содержит величины, вычисляемые в произведении по параметру i , то в схеме программы строится арифметический оператор, вычисляющий величину $i_{\text{кон}}$ по ее формуле определения;

3) в схеме программы строится оператор индексации по параметру i с шагом индексации, равным Δi , и с соответствующим высказыванием σ , обеспечивающим принятие параметром i всех значений из области его задания;

4) в схеме программы строится выходящая стрелка управления, соответствующая входящей стрелке, расположенной непосредственно после оператора засылки начального значения параметра i ;

5) если $i_{\text{кон}}$ — неизвестно, то в схему программы передается особый логический оператор и после него строится выходящая стрелка управления, соответствующая входящей стрелке, расположенной непосредственно после оператора засылки начального значения параметра i .

Заметим, что построение схемы программы по заданной схеме счета можно осуществить также другим способом, добавляя к построениям, описанным правилами I—V, следующие построения:

для вложенных произведений для старшей входящей стрелки вложенных произведений строится оператор засылки начального значения параметра для каждого параметра этого произведения;

для нестаршей выходящей стрелки произведения после всех построений, связанных с нею, строятся операторы индексации по параметрам данного произведения с тождественно истинным высказыванием и с шагом индексации, равным $i_{\text{нач}} - i_{\text{кон}}$.

В качестве примера, построим в операторном языке II схему программы алгоритма умножения треугольных матриц*.

Алгоритм получения треугольной матрицы C — произведения треугольных матриц A и B , — может быть записан в виде следующей схемы счета:

$$\downarrow^k \downarrow^l (0 \rightarrow d) \downarrow^l A_{kil} D \uparrow^l L_{kl} \uparrow^l \uparrow^k \text{ остан.},$$

$$k = 1, 2, \dots, n; l = 1, 2, \dots, k; i = l, l+1, \dots, k,$$

где оператор $A_{kil} : \{a_{ki}, b_{il}\} \rightarrow f$ вычисляет произведение величин a_{ki} и b_{il} и засылает его в величину f ; оператор $D : \{f, d\} \rightarrow d$ вычисляет сумму величин f и d и засылает ее в величину d ; оператор $L_{kl} : d \rightarrow c_{kl}$ величину d засылает в величину c_{kl} .

Согласно правилам преобразования схемы счета в схему программы, приведенной схеме счета соответствуют следующие схемы программ, построенные первым и вторым способами соответственно:

* Подробное описание логических схем алгоритма умножения треугольных матриц в операторном языке I дается в работе [2]. В данной работе пример рассматривается для иллюстрации правил преобразования схемы счета в схему программы и, далее, в рабочую схему программы.

1. $(k_{\text{нов}} \rightarrow k) \uparrow^1 (l_{\text{нов}} \rightarrow l) \uparrow^2 (0 \rightarrow d) (l \rightarrow i) \uparrow^3 A_{kl} DY(1, i, z_1) \uparrow^4 L_{kl} Y(1, l, z_2) \uparrow^5 Y(1, k, z_3) \uparrow^6 \text{останов},$
2. $(k_{\text{нов}} \rightarrow k)(l_{\text{нов}} \rightarrow l)(l \rightarrow i) \uparrow^1 \uparrow^2 (0 \rightarrow d) \uparrow^3 A_{kl} DY(1, i, z_1) \uparrow^4 Y(l - k, i, z_2) \uparrow^5 L_{kl} Y(1, i, z_3) \uparrow^6 Y(1, l, z_2) \uparrow^7 Y(-k, l, z_4) \uparrow^8 Y(-k, i, z_4) \uparrow^9 Y(1, k, z_3) \uparrow^{10} \text{останов},$

где z_1 — высказывание $i = k + 1$;

z_2 — высказывание $l = k + 1$;

z_3 — высказывание $k = n + 1$;

z_4 — тождественно истинное высказывание.

§ 3. Построение рабочей схемы программы

Переход к рабочей параметризации* и построение рабочей схемы программы является важным этапом построения схем алгоритма решения данной задачи.

В настоящем параграфе описываются преобразования схемы программы в операторном языке II, вызванные введением рабочей параметризации.

Переход к рабочим параметрам в операторном языке II отражается на параметрических операторах и операторах индексации и засылки начального значения параметра.

Построение схемы программы, соответствующей данной схеме счета, по второму способу (см. § 2 настоящей работы) представляет интерес с точки зрения дальнейших преобразований логических схем, и потому в данном параграфе для построения рабочей схемы программы в качестве исходной берется схема программы, построенная для заданной схемы счета по второму способу.

Пусть M — некоторое множество числовых величин данной схемы счета, параметризованное эквивалентными параметрами $i_1, i_2, \dots, i_n$, и пусть параметры $s_1, s_2, \dots, s_k$ осуществляют рабочую параметризацию множества величин M . Пусть Y_n и S_k — области определения множеств $(i_1, \dots, i_n)$ и $(s_1, \dots, s_k)$ и формулами

$$s_r = s_r(i_1, i_2, \dots, i_n), r = 1, 2, \dots, k,$$

$$i_t = i_t(s_1, s_2, \dots, s_k), t = 1, 2, \dots, n \quad (3)$$

устанавливается соответствие между множествами Y_n и S_k , при котором соответствующие наборы значений параметров $i_1, i_2, \dots, i_n$ и $s_1, s_2, \dots, s_k$ определяют одну и ту же величину множества M .

При параметризации множества M рабочими параметрами $s_1, s_2, \dots, s_k$ эквивалентные параметры параметрического оператора схемы заменяются на рабочие параметры тех множеств числовой инфор-

* Рабочей параметризацией множества величины называется параметризация этого множества посредством одного параметра, принимающего значения из некоторого отрезка натурального ряда.

мации, величины которых являются входными или выходными величинами данного параметрического оператора.

В силу (3), изменение каждого параметра i_{i_0} на некоторую величину Δi_{i_0} приводит к приращению параметров $s_1, s_2, \dots, s_k$, вычисляемого по формулам

$$\Delta i_{i_0} s_r = s_r (i_1, i_2, \dots, i_{i_0} + \Delta i_{i_0}, i_{i_0+1}, \dots, i_k) - s_r (i_1, \dots, i_{i_0}, \dots, i_k),$$

$$r = 1, \dots, k. \quad (4)$$

Вследствие этого, при переходе к рабочим параметрам в схеме программы перед каждым оператором индексации по эквивалентному параметру i_{i_0} строятся операторы индексации по тем рабочим параметрам, формулы которых содержат символ параметра i_{i_0} . Шаг индексации каждого оператора индексации по рабочему параметру равен приращению этого рабочего параметра, вычисляемого по формуле (4), и в операторе индексации выбрано тождественно истинное высказывание.

Чтобы не усложнять переход к рабочей схеме программы выходящим стрелкам управления построенных операторов индексации будем приписывать номер выходящей стрелки управления рассматриваемого оператора индексации.

Преобразование схемы программы в рабочую схему программы осуществляется по следующим правилам:

1. Если рассматриваемый элемент схемы программы—оператор засылки начального значения параметра, то:

1) в рабочей схеме программы строятся операторы засылки начального значения параметра для тех рабочих параметров, формулы которых содержат параметр рассматриваемого оператора засылки; при этом, для каждого рабочего параметра оператор засылки начального значения параметра строится только один раз;

2) в рабочую схему программы передается рассматриваемый оператор засылки начального значения параметра.

II. Если рассматриваемый элемент схемы программы—параметрический оператор или стрелка управления, то этот элемент без изменения передается в рабочую схему программы.

III. Если рассматриваемый элемент схемы программы—параметрический оператор, то параметры этого оператора заменяются на соответствующие рабочие параметры, и полученный элемент передается в рабочую схему программы.

IV. Если рассматриваемый элемент схемы программы—оператор индексации, то:

1) в рабочей схеме программы строятся операторы индексации с тождественно истинным высказыванием по тем рабочим параметрам формулы которых содержат параметр рассматриваемого оператора индексации; при этом шаг индексации построенных операторов индексации вычисляется по формулам (4);

2) после каждого оператора индексации в рабочую схему программы передается выходящая стрелка управления с номером выходящей стрелки управления рассматриваемого оператора индексации;

3) в рабочую схему программы передается рассматриваемый оператор индексации.

Таким образом, по правилам I—IV для каждой схемы программы осуществляется построение соответствующей рабочей схемы программы.

Так, построенной ранее второй схеме программы умножения треугольных матриц (см. § 2) при параметризации числовой информации рабочими параметрами

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

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

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

по правилам преобразования схемы программы в рабочую схему программы соответствует следующая рабочая схема:

$$\begin{aligned} & (r_{\text{нач}} \rightarrow r)(s_{\text{нач}} \rightarrow s)(t_{\text{нач}} \rightarrow t)(k_{\text{нач}} \rightarrow k) \uparrow^1 (l_{\text{нач}} \rightarrow l) \uparrow^2 (l \rightarrow i) (0 \rightarrow d) \uparrow^3 \\ & A_{r,s} Y(\Delta_l r, r, \sigma_4) \uparrow^3 Y(\Delta_l s, s, \sigma_4) \uparrow^3 Y(1, i, \sigma_1) \uparrow^3 Y(\Delta_1, r, \sigma_4) \uparrow^3 Y(\bar{\Delta}_2, s, \sigma_4) \uparrow^3 \\ & L_t Y(\Delta_l s, s, \sigma_4) \uparrow^2 Y(\Delta_l r, r, \sigma_4) \uparrow^2 Y(\Delta_l t, t, \sigma_4) \uparrow^2 Y(1, l, \sigma_2) \uparrow^2 Y(\bar{\Delta}_1, s, \sigma) \\ & \uparrow^2 Y(\bar{\Delta}_2, t, \sigma_4) \uparrow^2 Y(\Delta_3, r, \sigma_4) \uparrow^2 Y(\Delta_k r, r, \sigma_4) \uparrow^1 Y(\Delta_k t, t, \sigma_4) \uparrow^1 \\ & Y(1, k, \sigma_3) \uparrow^1 \text{останов.} \end{aligned}$$

В приведенной рабочей схеме

σ_4 — тождественно истинное высказывание,

σ_1 — высказывание $i = k + 1$; σ_2 — высказывание $l = k + 1$;

σ_3 — высказывание $k = n + 1$;

$$\Delta_l r = 1;$$

$$\Delta_l t = 1;$$

$$\Delta_l s = i;$$

$$\bar{\Delta}_1 = \frac{-k(2l - k + 1)}{2};$$

$$\Delta_1 = l - k - 1;$$

$$\bar{\Delta}_2 = -k;$$

$$\Delta_2 = \frac{(l - k - 1)(2l + l - k - 2)}{2};$$

$$\Delta_3 = -k;$$

$$\Delta_l s = l + 1;$$

$$\Delta_k r = k;$$

$$\Delta_l r = 1;$$

$$\Delta_k t = k.$$

При вычислении шагов индексации $\Delta_l s$, $\Delta_l r$, $\bar{\Delta}_1$ и Δ_3 учтена зависимость $i = l$.

Заметим, что в построенной рабочей схеме программы построение операторов индексации по рабочим параметрам привело к тому, что в рабочей схеме программы имеются операторы индексации по одному и тому же рабочему параметру с тождественно истинным высказыванием. Легко проверить, что произведение операторов индексации с тождественно истинным высказыванием может быть преобразовано по правилам, аналогичным правилам преобразования операторов переадресации [3].

§ 4. О реализации оператора индексации на электронной вычислительной машине (ЭВМ)

Программирование переадресаций на ЭВМ с индексными регистрами осуществляется при помощи набора операций, функции которых могут быть описаны следующим образом:

- 1) получение модифицированного адреса прибавлением содержимого индексного регистра к данному адресу (модификация адреса);
- 2) изменение содержимого индексного регистра прибавлением к нему некоторого приращения;
- 3) проверка на окончание сравнением содержимого индексного регистра с некоторой заданной константой сравнения;
- 4) засылка начального значения индекса в индексный регистр.

Функции второй и третьей группы обычно в ЭВМ реализуются так называемой, групповой операцией.

Предположим, что количество индексных регистров в ЭВМ практически неограничено. Сопоставим каждому параметру данной логической схемы некоторый индексный регистр, так чтобы разным параметрам были сопоставлены разные индексные регистры. Тогда, оператор засылки начального значения параметра можно в машине реализовать операцией засылки начального значения индекса в индексный регистр. Выполнение параметрического оператора может быть реализовано посредством модификации переменных адресов данного параметрического оператора, а оператор индексации реализуется групповой операцией. При этом для вычисления шага индексации желательно в наборе операций ЭВМ иметь операции и над содержимым индексных регистров.

Заметим, что в случае ограниченного количества индексных регистров индексные регистры, использованные при выполнении вложенных произведений, после выполнения произведения по старшему параметру могут быть вновь использованы. Кроме того, обычно среди операций машины имеется операция, дающая возможность запомнить содержимое индексного регистра для его использования в дальнейшем.

В силу сделанного замечания, практическая ограниченность количества индексных регистров в ЭВМ не существенна.

§ 5. О построении рабочей схемы программы в операторном языке II по рабочей схеме программы, заданной в операторном языке I

Вопрос о непосредственном построении рабочей схемы программы в операторном языке II по заданной в операторном языке I рабочей схеме программы важен при практическом использовании программирующих программ, составленных для ЭВМ, не имеющих индексных регистров, на ЭВМ с индексными регистрами (в частности для использования программирующей программы, описанной в работах [2, 3]).

Преобразование некоторой рабочей схемы программы L в операторном языке I в соответствующую ей в операторном языке II рабочую схему программы $\bar{L}$ выполняется по следующим правилам:

1) вместо оператора восстановления строится произведение операторов засылки начального значения рабочих параметров данного произведения;

2) вместо оператора формирования для всех рабочих параметров формирования строятся операторы вычисления и засылки значения этих параметров;

3) вместо каждого произведения оператора переадресации, оператора изменения параметра и логического оператора проверки достижения параметром его конечного значения строится оператор индексации по параметру переадресации. При этом, если в указанном произведении отсутствуют некоторые операторы, оператор индексации строится в соответствии со случаями, разобранными в § 1;

4) вместо каждого оператора, входные или выходные величины которого — параметрические, передается соответствующий параметрический оператор;

5) остальные элементы рабочей схемы программы L передаются без изменения.

После выполнения для каждого элемента схемы L соответствующего правила будет построена в операторном языке II рабочая схема программы $\bar{L}$.

Аналогично, могут быть описаны правила построения схемы L по рабочей схеме $\bar{L}$.

Հ. Գ. ՓԻՆՓԻՆՈՅԱՆ, Ս. Մ. ԴԱԺԵԱՆ

ԻՆՏԵՐՍՈՅՈՒՆԻ ՀՈՒՆԵՍԻԱՆԵՐԻ ԻՐԱԴՈՐՈՒՄԸ ՕՊԵՐԱՏՈՐՆԵՐ ԼԵՂՎՈՒՄ

Ա մ փ ո փ ո լ մ

Այս աշխատանքում տրվում են ինդեքսացման օպերատորի սահմանումը և համապատասխանորեն վերափոխում [2] աշխատանքում շարադրված օպերատորային լեզվի տրամաբանական, հաշվման և ծրագրման սխեմաների սահ-

մանումները: Այնուհետև տրվում է կանոնների մի սիստեմ, որի օգնությամբ տրված հաշվման սխեմայից կարող ենք կառուցել նրան համարժեք ծրագրման և աշխատանքային ծրագրման սխեմաները նոր սահմանված լեզվում: Հիմնավորվում է, որ ինդեքսացիայի օպերատորով ծրագրման սխեմաներն իրագործում են ժամանակակից էլեկտրոնային հաշվիչ մեքենաների ինդեքսացման ֆունկցիաներից ավելի լայն ֆունկցիաներ: Վերջին պարագրաֆում տրվում է մի ալգորիթմ, որի օգնությամբ առաջին օպերատորային լեզվի ամեն մի աշխատանքային ծրագրման սխեմայի համար կարելի է կառուցել նրան համարժեք աշխատանքային ծրագրման սխեմա II օպերատորային լեզվում:

Л И Т Е Р А Т У Р А

1. Блоу Дж. Э. Индексация и метод управляющих слов. Кибернетический сборник, 2, 1961.
2. Пилипосян А. Г., Давтян С. М. Описание универсальной программирующей программы, выполняющей некоторые преобразования логических схем. I, Труды Вычислительного центра АН АрмССР и ЕрГУ, вып. 3, 1965.
3. Пилипосян А. Г., Давтян С. М. Описание универсальной программирующей программы, выполняющей некоторые преобразования логических схем. II, Труды Вычислительного центра АН АрмССР и ЕрГУ, вып. 4, 1966.