

УДК 681.3.06:51

СХЕМНО-БЛОЧНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

Л.Т.Петрова, Н.Н.Пивкина

В работе описывается язык программирования и приводится пример его использования. Естественно, что разработка языка проводилась с учетом и под влиянием накопленного опыта в области автоматизации программирования. Язык разработан в духе идей школы [1]. Ему свойственны крупноблочность и система косвенных ссылок на уже накопленную информацию. Здесь можно упомянуть работу [2]. Из работ других направлений сошлемся на [3], [4], [5]. Этот список, конечно, может быть продолжен. Схемно-блочный язык ориентирован на математико-экономические задачи с блочной структурой данных и опробован на алгоритмах из [6]. Характер информации в этих задачах, естественно, повлиял на выбор основных объектов языка. В частности, это неправильные массивы — отдельные или организованные в схемы (деревья), и сами схемы, являющиеся полезным инструментом для анализа и выявления связей между блоками. Вычислительный характер задач обусловил выбор средств в языке, позволяющих учитывать дополнительную информацию для создания качественной программы. Эта информация касается, например, способа подачи данных или частичного планирования памяти. Перейдем к синтаксису языка. Синтаксис его задан с помощью бэкусовых формул.

§ 1. Символы

$$\langle \text{символ} \rangle ::= \langle \text{буква} \rangle \mid \langle \text{цифра} \rangle \mid \langle \text{ограничитель} \rangle$$
$$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

<ограничитель> ::= <скобка> | <операция> | <разделитель> |
 <описатель> | <характеристика> | <указатель>
 <скобка> ::= начало { [[(()]] } конец
 <операция> ::= <арифметическая операция> | <схемная опера-
 ция> | <операция отношения>
 <арифметическая операция> ::= + | - | x | / | \odot
 <операция отношения> ::= > | < | = | \neq | \geq | \leq | \in | \notin | лист
 <схемная операция> ::= \cap | \uparrow | прав: | лев: | прав | лев
 <разделитель> ::= . | , | : | ; | := | =: | :: | * | ? | да | для |
цикл | шаг | пока | на
 <описатель> ::= массив | массив-уровень | массив-цепь |
набор | схема | область | лента | барабан | цепь | процедура |
дисперсия | моменты
 <характеристика> ::= справка | адрес | значение | индекс |
признак | вес | координаты
 <указатель> ::= разместить | включить | исключить |
настроить | очередной

§ 2. Имена и числа

Синтаксис

<имя общее> ::= <имя> | <специальное имя> | <полное имя>
 <имя> ::= <буква> | <имя> <буква или цифра>
 <буква или цифра> ::= <цифра> | <буква>
 <специальное имя> ::= <имя> *
 <полное имя> ::= <специальное имя> <имя или целое>
 <имя или целое> ::= <имя> | <целое>
 <целое> ::= <цифра> | <целое> <цифра>
 <число> ::= <целое> | <вещественное>
 <вещественное> ::= <целое> . | <целое> . <целое>
 <число с признаком> ::= <целое> : <число>

	Примеры	
<имя>	<спец.имя>	<полное имя>
t	$A*$	$A*10$
$A15$	$B1*$	$B*m$
BC	$abc*$	$ab*r1$

<целое>	<вещ.>	<число с признаком>
I25	I25.	15 : 10 . 5
64.	10 . 025	2 : 54
I	0 . 62	1 : 12

Семантика

<Имя> используется для обозначения отдельных чисел (констант или скаляров), пар чисел (констант или скаляров с признаком), различных числовых массивов (сюда относятся наборы чисел, массивы, области), схем, процедур.

<Специальное имя> обозначает область, в которой размещается набор массивов.

<Полное имя> обозначает отдельный массив или отдельную область из набора. Так, $A * 10$ обозначает десятый массив, размещенный в области $A *$. $B * m1$ обозначает тот массив из области $B *$, порядковый номер которого определяется значением переменной $m1$.

<Число> и <число с признаком> может служить значением константы, скаляра, элемента массива или схемы (§ 3). Условимся, что имена объектов с целым значением начинаются с одной из следующих букв: $i, j, k, l, m, n, L, M, N$.

§ 3. Объекты

<объект> ::= <числовой объект> | <схемный объект> | <справочный объект>

3.1. Числовые объекты

Синтаксис

<числовой объект> ::= <константа> | <скаляр> | <компонента массива>

<константа> ::= <число> | <имя константы>

<имя константы> ::= <имя>

<скаляр> ::= <имя скаляра> ::= <имя>

<компонента массива> ::= <имя массива> [<индексный ряд>]

<имя массива> ::= <имя общее>

<индексный ряд> ::= <индекс или признак> | <индексный ряд>

<индексный ряд>

<индекс или признак> ::= <индекс> | <признак>

<индекс> ::= <константа> | <скаляр> | *

<признак> ::= <индекс> :

Примеры

25

1.84

t

$A[5, n]$

$B * k[n, m]$

$C[n:]$

$D[* , k]$

$F * I[n : , *]$

$A I[* , *]$

$u * [i]$

Семантика

С каждым числовым объектом связывается его числовое значение - число (с признаком или без него) или упорядоченная совокупность чисел. Значением констант и скаляров являются отдельные числа. Значением компоненты массива может быть отдельное число или некоторый подмассив. При помощи числовых объектов организуется доступ к отдельным элементам массива или к его подмассивам. Так, значением компоненты $A[n, m]$ является m -тый элемент в n -ой строке матрицы A , $A[5, *]$ - пятая строка, $A[* , k]$ - k -тый столбец, $A[* , *]$ - весь массив A .

Выше уже говорилось, что значением элемента массива может быть число с признаком. Признаками могут снабжаться также и целые компоненты массива по той или иной размерности. В этом случае компоненты могут определяться не только их индексами, но равно и признаками. Например, объект $C[n:]$ означает элемент с признаком n из вектора C . В $[n : , m]$ означает m -тый элемент в строке, имеющей признак n . В $[k : , *]$ - строка с признаком k и т.п. В такой форме осуществляется ассоциативный поиск информации. Значения индексов и признаков предполагаются целыми.

3.2. Схемные объекты

Синтаксис

<схемный объект> ::= <вершина> | <схема> | <путь>

<вершина> ::= <имя схемы> [\langle индекс или признак \rangle]

<схема> ::= <имя схемы> <скобка {(> <индекс или признак>

<скобка > } >

<путь> ::= <имя схемы> <скобка [(> <индекс или признак>, <индекс или признак > <скобка)] >

<имя схемы> ::= <имя>

<скобка { (> ::= { | (

<скобка) } > ::=) | }

<скобка [(> ::= [(| (

<скобка)] > ::=) |]

	Примеры	
<вершина>	<схема>	<путь>
$Q[n]$	$P(n1)$	$Q[k;n]$
$D[m:]$	$P\{n2:\}$	$Q[k,*]$

Семантика

Схема - это ориентированный граф без контуров, имеющий не более одной входящей дуги в каждой вершине (т.е. дерево). Длина пути в конечном дереве ограничена, поэтому существуют максимальные пути, не являющиеся строгими подпутями никаких более длинных путей. Начальная и конечная вершины любого максимального пути называются, соответственно, корнем и листом. Принимается, что корень находится на первом уровне, и вообще, вершина, лежащая в конце пути длиной j от корня, находится на j -том уровне. Удобными характеристиками схемы являются вектор моментов и вектор дисперсии. Компоненты вектора моментов определяют число вершин на каждом уровне. Для схемы, изображенной на рис. 1, вектором моментов является вектор $M1 = (1, 3, 7, 2)$. Ту же размерность имеет вектор дисперсии $D0 = (d1, d2, \dots, d\upsilon)$, определяемый следующим образом: его первая компонента $d1$ равна числу корней в дереве, а для $i = 2, 3, \dots, \upsilon$ di равно максимальной полустепени исхода среди всех вершин $(i-1)$ -го уровня. В нашем примере $D0 = (1, 3, 3, 2)$. Каждая вершина схемы может быть помечена признаком так, как это показано на рис. 1 [3].

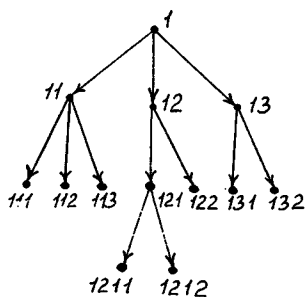


Рис. I. Схема.

Признаки вершин удобно сделать одинаковыми по количеству символов в них. Если значащие символы признака сдвинуть вправо, а слева пополнить признак нулями, то будем иметь правый признак. При сдвиге влево имеем левые признаки вершин (см. табл. I). Схема полностью определяется списком признаков ее вершин. Рассмотрим два специальных упорядочения схемы. Признаки вершин будем считать целыми числами. Пусть список правых признаков упорядочен по их возрастанию. В этом случае говорят, что имеется правая схема (схема Q из таблицы I). В случае упорядочения списка левых признаков по их возрастанию получается левая схема (схема P из таблицы I).

Таблица I. Правая и левая схемы для графа, изображенного на рис. I.

Индексы	правая схема Q	левая схема P	нагруженная левая схема PQ
1	0001	1000	1000 : 1
2	0011	1100	1100 : 2
3	0012	1110	1110 : 5
4	0013	1120	1120 : 6
5	0111	1130	1130 : 7
6	0112	1200	1200 : 3
7	0113	1210	1210 : 8
8	0121	1211	1211 : 12
9	0122	1212	1212 : 13
10	0131	1220	1220 : 9
11	0132	1300	1300 : 4
12	1211	1310	1310 : 10
13	1212	1320	1320 : 11

В правой схеме вершины упорядочены по уровням: вершина 0001 лежит на первом уровне, следующие за ней вершины 0011, 0012, 0013 - на втором уровне и т.д. В левой схеме вершины упорядочены по поддеревьям: следом за каждой вершиной выписаны (если они имеются) вершины порожденного ею поддерева. Все признаки таких вершин содержат в себе признак порождающей вершины. Так, за вершиной I200 непосредственно следуют вершины I210, I211, I212, I220, образующие поддерево с корнем в вершине I200. Для разных целей оказывается более удобным одно или другое упорядочение схемы. Поэтому для одного и того же графа могут быть заданы обе схемы - левая и правая.

Кроме того, схема может быть нагружена, т.е. каждой вершине ставится в соответствие числовое значение типа \langle число с признаком \rangle . Здесь признаком является признак вершины, а числом - некоторый весовой коэффициент данной вершины. В частности, бывает удобно в левой схеме иметь в качестве весов порядковые номера вершин в правой схеме (схема PQ из табл. I), и наоборот.

Значением схемы является набор числовых значений ее вершин. В нагруженной схеме это признаки и веса вершин, в ненагруженной - только признаки. Значением схемного объекта $S[n]$ является числовое значение n -ой по порядку вершины в схеме S . Значением $S[n:]$ является числовое значение вершины с признаком n в схеме S . В частности, в ненагруженной схеме значением $S[n:]$ является значение n . При этом если схема S левая (правая), то и результатом является левый (правый) признак. Это позволяет преобразовывать левые признаки в правые и обратно. В таблице 2 приводятся схемные объекты типа \langle вершина \rangle из нашего примера (рис. I, табл. I), а также значения этих объектов.

Таблица 2. Схемные объекты типа \langle вершина \rangle и их значения.

Объекты	Значение объекта
Q[4]	0013
P[4]	I120
PQ[4]	I120 : 6
n	I200
Q[n:]	0012
P[n:]	I200
PQ[n:]	I200 : 3

Каждая вершина схемы порождает подсхему с корнем в этой вершине. Значением объекта $S\{n\}$ является подсхема с корнем в n -ой по порядку вершине схемы S . Значением $S\{n:\}$ является подсхема с корнем в вершине, имеющей признак n . Замена левой фигурной скобки на круглую означает, что корень не включается в результирующую подсхему. Замена правой фигурной скобки на круглую исключает из подсхемы все ее листы. В таблице 3 приведены схемные объекты типа <схема> и их значения.

Таблица 3. Схемные объекты типа <схема> и их значения.

Объект	Значение	Объект	Значение	Объект	Значение
$P\{6\}$ или $P\{1200:\}$	I200 I210 I211 I212 I220	$P(6)$	I210 I211 I212 I220	$P\{6\}$	I200 I210
$Q\{3\}$ или $Q\{0012:\}$	0012 0121 0122 I211 I212	$Q(3)$	0121 0122 I211 I212	$Q\{3\}$	0012 0121
$PQ\{7\}$ или $PQ\{1210:\}$	I210 : 8 I211 : 12 I212 : 13	$PQ(7)$	I211 : 12 I212 : 13	$PQ\{7\}$	\emptyset

Признак вершины содержит в себе информацию о том единственном пути, который ведет из данной вершины в корень дерева. Этот путь очевидным способом и легко восстанавливается по признаку. Так, путь из вершины I212 проходит через вершины I210, I200, I000 (в левой схеме) или через вершины 0121, 0012, 0001 (в правой схеме). Путь из одной вершины в другую (единственный, если он существует) определяется признаком исходной вершины и числом вершин на этом пути (если число вершин равно нулю, ука-

занного пути не существует). Так, пара чисел (1212 : 3) определяет путь из вершины 1212 в вершину 1200 (в левой схеме) или в вершину 0012 (в правой схеме).

Значением схемного объекта $Q[k, n]$ является информация о пути p в схеме Q , ведущем из k -ой в n -ую вершину. Эта информация представляется значением типа <число с признаком>, т.е. парой чисел $(p:q)$, где p - признак k -той вершины в Q , а вес q - число вершин на указанном пути. Замена в объекте $Q[k, n]$ левой скобки на круглую означает, что из указанного пути исключается начальная вершина. Замена правой скобки исключает конечную вершину. $Q[k, *]$ означает путь из k -той вершины в корень дерева Q . В таблице 4 приведены схемные объекты типа <путь> и их значения.

Таблица 4. Схемные объекты типа <путь> и их значения.

Объект	Значение
$Q[12, 3]$	1211 : 3
$P[12, 3]$	1310 : 0
$Q[12, 3]$	0121 : 1
$Q(7, *)$	0011 : 2
$Q[132 :, *]$	0132 : 3
$P[10, *]$	1220 : 2
$P[9, *]$	1212 : 4
$P[6, 10]$	1200 : 0
$PQ[10, *]$	1220 : 3

3.3. Справочные объекты

Синтаксис

<справочный объект> ::= <ряд характеристик> <величина>
 <величина> ::= <числовой объект> | <схемный объект>
 <ряд характеристик> ::= <характеристика> | <характеристика> <цифра> | <ряд характеристик> <ряд характеристик>

Примеры

индекс $P[k:]$
признак $2BE_k, *$
индекс 2 справка $\Lambda * K[n:, *]$
вес $Q[k, n]$
координаты $M[n]$
признак z
индекс значение $g[m:]$
признак 2 признак 1 $B[k, m]$

Семантика

Каждому массиву, набору или схеме, согласно их описанию, при трансляции соотносится ряд служебных характеристик, т.е. та информация с помощью которой и организуется доступ к значению этих объектов или их отдельных компонент. К таким характеристикам относятся, в частности, начальный адрес размещения значений объекта в памяти, размеры объекта по всем измерениям, а также соответствующие шаги, которые отделяют в памяти одну компоненту данной размерности от другой. Эти служебные характеристики составляют справку объекта. При помощи справочных объектов справки и их элементы становятся доступными для программиста. Так, справочный объект справка $A[*,*]$ означает справку о матрице A . Кроме того, можно формировать справки об отдельных компонентах: справка $\Lambda * K[n:, *]$ означает справку о строке с признаком n в матрице $\Lambda * K$. По желанию формируются и присваиваются справки вновь вычисляемым объектам (5.1). Таким образом, характеристики справка и адрес имеют очевидный смысл и допустимы для всех типов величин (это отмечено знаком + в табл. 5). Характеристика значение используется только в том случае, когда справочный объект содержит наряду с ней и другие характеристики. Поясним смысл остальных характеристик для различных типов величин. Характеристики признак и вес связаны с числовыми значениями элементов типа <число с признаком> следующим образом. Если значением скаляра z является число q с признаком p ($p:q$), то значение объекта признак z есть p , а объекта вес z - q . Аналогичный смысл имеют

эти характеристики для элементов массивов или схем, с той особенностью, что значением объекта типа признак $A[n:]$ является значение n . На базе нагруженной схемы PQ справочный объект признак $PQ\{n\}$ определяет ненагруженную подсхему с корнем в n -ой вершине, а объект вес $PQ\{n\}$ - набор весов, отвечающих этой же подсхеме. Для ненагруженной величины типа <схема> характеристики признак и вес обычно не используются. Если Q - схема, то объект вес $Q[k, n]$ означает число вершин на пути $Q[k, n]$. В частности, значением объекта вес $Q[m, *]$ является номер уровня, на котором лежит вершина m в схеме Q . Значением объекта признак $Q[k, n]$ является набор признаков вершин, лежащих на пути $Q[k, n]$.

В многомерных массивах характеристика вес по-прежнему относится к элементам, т.е. к первому измерению. А характеристика признак может иметь дополнение <цифра>, т.е. принимать вид признак 1, признак 2 и т.д., и выполнять несколько другую функцию, не зависящую от типа числовых значений элементов. В этом случае, например, объект признак $2B[k, *]$ (равно как и объект признак $2B[k, m:]$) означает признак (по второму измерению в массиве B) k -той строки. В этой второй функции характеристика признак сходна с характеристикой индекс. Так, объект индекс $A[n:]$ определяет индекс (порядковый номер) элемента с признаком n в векторе A , а объект индекс $2B[k, *]$ - индекс строки с признаком k в матрице B . Для величины типа <скаляр> значение характеристики индекс не определено (этот факт отмечается знаком - в табл. 5). Объект индекс $S[n:]$ означает индекс вершины n в схеме S . Справочный объект индекс $S\{n\}$ означает набор индексов тех вершин из схемы S , которые входят в подсхему $S\{n\}$. Объект индекс $Q[k, n]$ означает набор индексов тех вершин из схемы Q , которые образуют путь $Q[k, n]$. Наконец, рассмотрим последнюю характеристику координаты. Объект коорд $M[n]$ означает пару чисел (i, k) таких, что $M[i-1] < n \leq M[i]$, $k = n - M[i-1]$, (обычно M - набор целых чисел, упорядоченных по их возрастанию).

Справочный объект индекс 2 справка $\Lambda * k[n, *]$ содержит, по существу, два объекта: индекс (по второму измерению в массиве $\Lambda * k$) величины $\Lambda * k[n, *]$ и справку той же величины. Справочный объект может состоять и из большего числа

Таблица 5. Справочные объекты и их значения

Тип величины	Величина	справ-ка	адрес	зна-че-ние	ин-декс	прим-знак	вес	коорд.
<скаляр>	Z1	+	+	84 . 5	-	-	-	-
	Z2	+	+	0 : 55.	-	0	55.	-
<компо- нента масси- ва >	A[*]	+	+	10 : 22. 11 : 33. 12 : 44.	-	10 11 12	22. 33. 44.	-
	A[2]	+	+	11 : 33.	2	11	33.	-
	A[10:]	+	+	10 : 22.	1	10	22.	-
<вершина>	Q[5]	+	+	0111	5	-	-	-
	P[1200:]	+	+	1200	6	-	-	-
	PQ[5]	+	+	1130 : 7	5	1130	7	-
<схема>	Q{4}	+	+	0013 0131 0132	4 10 11	-	-	-
	PQ{11}	+	+	1300 : 4 1310 : 10 1320 : 11	11 12 13	1300 1310 1320	4 10 11	-
<путь>	P[7,*]	+	+	1210 : 3	7 6 1	1210 1200 1000	3	-
<компо- нента масси- ва >	M[*]	+	+	2 4 6 ... 100	-	-	-	-
	M[5]	+	+	10	5	-	-	3,1

отдельных объектов, соответственно количеству характеристик в нем.

§ 4. Выражения

Синтаксис

$\langle \text{выражение} \rangle ::= \langle \text{арифметическое выражение} \rangle \mid \langle \text{схемное выражение} \rangle \mid \langle \text{справочное выражение} \rangle$
 $\langle \text{арифметическое выражение} \rangle ::= \langle \text{числовой объект} \rangle \mid (\langle \text{арифметическое выражение} \rangle) \mid \langle \text{арифметическое выражение} \rangle \langle \text{арифметическая операция} \rangle \langle \text{арифметическое выражение} \rangle$
 $\langle \text{схемное выражение} \rangle ::= \langle \text{схемный объект} \rangle \mid \langle \text{путь} \rangle \cap \langle \text{путь} \rangle \mid \langle \text{операция I} \rangle \langle \text{схема} \rangle$
 $\langle \text{операция I} \rangle ::= \text{лев} \mid \text{прав} \mid \uparrow \mid \text{лев} : \mid \text{прав} :$
 $\langle \text{справочное выражение} \rangle ::= \langle \text{справочный объект} \rangle \mid \langle \text{очередной} \rangle \langle \text{справочный объект} \rangle$
 $\langle \text{очередной} \rangle ::= \text{очередной} \mid \text{очередной} \langle \text{цифра} \rangle$

Примеры

$\langle \text{арифметическое выражение} \rangle$

$u * n[*] \circ B[*, *]$
 $a + t[m, *] \circ f[*]$
 $(p + q) - \Lambda * k[i, n]$

$\langle \text{схемное выражение} \rangle$

$\text{лев } Q\{*\}$
 $\uparrow P\{m\}$
 $Q[k, n] \cap Q[m, *]$

$\langle \text{справочное выражение} \rangle$

очередной 2 признак справка $\Lambda * k[*, *]$
очередной значение индекс $B[*, *]$
индекс $P[k, *]$

Семантика

$\langle \text{Арифметическое выражение} \rangle$ описывает правило вычисления

значения числового объекта. В арифметическом выражении могут фигурировать как отдельные числа, так и целые компоненты. В этом последнем случае арифметические операции (+, -, \times , /) выполняются поэлементно, а операция \odot понимается как скалярное произведение двух векторов или выполняется по обычным правилам умножения матрицы на вектор, вектора на матрицу и т.п. Операция \odot обладает наивысшим приоритетом. Размеры компонент, фигурирующих в одном и том же выражении, должны быть согласованы по одноименным измерениям в следующем смысле. Те компоненты, элементы которых имеют своими значениями числа без признаков (правильные компоненты), должны иметь одинаковые размеры по соответствующим измерениям. Неправильные компоненты (значениями их элементов являются числа с признаками) играют в выражениях специальную роль. Будем считать, что в выражении не более одной неправильной компоненты. Вычисления организуются следующим образом. На каждом шаге выбирается очередное значение элемента неправильной компоненты, и по этому значению определяются соответствующие ему элементы правильных компонент. А именно, признак выбранного неправильного элемента служит индексом соответствующих ему элементов из правильных компонент. Для определенных таким способом значений выполняются вычисления по формуле. Таким образом, неправильная компонента служит шкалой, направляющей выборку соответствующих элементов из компонент, участвующих в выражении. В качестве примера рассмотрим вычисление выражения $B[l, *] \odot C[*]$, где B - правильная матрица $n \times n$, а C - неправильный вектор из m элементов ($m < n$). Пусть перед началом вычислений $a = 0$. k -тый шаг условно представим в виде:

$$\text{шаг } k \begin{cases} p := \text{признак } C[k] \\ q := \text{вес } C[k] \\ a := a + B[l, p] \times q \end{cases}$$

Через m шагов a принимает значение скалярного произведения $B[l, *] \odot C[*]$.

<Схемное выражение> описывает правило вычисления значения схемного объекта. Поясним смысл схемных операций. Выражение лев $Q\{*\}$ является указанием о построении левой схемы по заданной схеме Q (т.е. речь идет о преобразовании признаков вершин схемы Q в левые признаки и о специальном их упо-

рядочении). Операция лев: строит нагруженную схему. Весом вершины является ее порядковый номер в исходной схеме. Выражение $\uparrow P(m)$ является указанием о записи числовых значений вершин схемы $P(m)$ в обратном порядке, от последнего к первому. Операция пересечения (\cap) двух путей понимается обычным образом. Пусть для некоторых путей $Q[k_1, n_1]$ и $Q[k_2, n_2]$ определен путь $Q[n, m]$, являющийся их пересечением:

$$Q[k_1, n_1] \cap Q[k_2, n_2] = Q[n, m].$$

Здесь вершина n - начало, а вершина m - конец общей части двух рассматриваемых путей.

<Справочное выражение> описывает правило вычисления значений ряда объектов. Значения отдельных справочных объектов описаны в 3.3 и проиллюстрированы в табл. 5. Указатель очередной с дополнением <цифра>, фигурирующий в справочном выражении, определяет специальный циклический режим выборки очередной (по данному измерению) компоненты (и связанных с ней справочных объектов). За последней компонентой снова выбирается первая и т.д. Этот режим используется, в частности, в операторах присваивания (5.1).

§ 5. Операторы

<оператор> ::= <оператор присваивания> | <оператор перехода> | <условный оператор> | <оператор настройки> | <оператор цикла> | <оператор размещения> | <оператор процедуры> | <стоп> | <оператор ввода> | <оператор вывода>

В тексте операторы отделяются друг от друга точкой с запятой. Операторы ввода и вывода на данной ступени конкретизации языка не рассматриваются.

5.1. Операторы присваивания (ОП)

<оператор присваивания> ::= <ОП значения> | <ОП формы> | <ОП значения и формы>
 <ОП значения> ::= <ряд объектов> := <выражение \mathcal{D} >
 <ряд объектов> ::= <объект А> | <ряд объектов>, <объект А>

$\langle \text{объект } A \rangle ::= \langle \text{величина} \rangle \mid \underline{\text{признак}} \langle \text{величина} \rangle \mid \underline{\text{вес}}$
 $\langle \text{величина} \rangle$
 $\langle \text{выражение } \mathcal{D} \rangle ::= \langle \text{выражение} \rangle \mid \langle \text{данные} \rangle$
 $\langle \text{данные} \rangle ::= (\langle \text{ряд чисел} \rangle) \mid \langle \text{данные} \rangle, \langle \text{данные} \rangle$
 $\langle \text{ряд чисел} \rangle ::= \langle \text{число} \rangle \mid \langle \text{число с признаком} \rangle \mid \langle \text{ряд}$
 $\text{чисел} \rangle \langle \text{запятая} \rangle \langle \text{ряд чисел} \rangle$
 $\langle \text{запятая} \rangle ::= , \mid \langle \text{звездочка} \rangle$
 $\langle \text{звездочка} \rangle ::= * \mid \langle \text{звездочка} \rangle \langle \text{звездочка} \rangle$
 $\langle \text{ОП формы} \rangle ::= \langle \text{имя} \rangle =: \langle \text{величина} \rangle$
 $\langle \text{ОП значения и формы} \rangle ::= \langle \text{величина} \rangle =: \langle \text{выражение } \mathcal{D} \rangle$

Примеры

$t[m] := a$
 $a := t[m]$
 $A := C[*]$
 $A[*] := B[k, *] + \Lambda * 5[m, *]$
 $g[i] := a + B[k, *] \circ C[*]$
 $m, f := \underline{\text{индекс 2 справка}} B[n, *]$
 $j, a := \underline{\text{очередной индекс значение}} C[*]$
 $i, l := \underline{\text{коорд}} M[k]$
 $S[*] := \uparrow P(m)$
 $n0 := Q[k, n]$
 $n1 := \underline{\text{вес}} n0$
 $N[*] := \underline{\text{индекс}} Q[k, n]$
 $Q[n_2, m] := Q[n_1, m_1] \cap Q[n_2, *]$
 $\underline{\text{вес}} \mathcal{D}[m] := a$
 $\underline{\text{признак}} 2 \Lambda * k[m, *] := n$

$A * 12[*] := (1:6, 3:5, 4:1*2:4, 4:2*1:7, 2:8)$

Семантика

$\langle \text{ОП значения} \rangle$ имеет общепринятый смысл, с той разницей, что в качестве результатов слева от знака присваивания ($:=$) может быть указано несколько объектов. В этом случае справа имеем справочное выражение с несколькими характеристиками или данные. Например, оператор $m, f := \underline{\text{индекс 2 справка}} B[n, *]$ означает, что скаляру m присваивается значение

индекса (по второму измерению в массиве В) строки с признаком n , а значением скаляра f становится справка о той же строке. После этого к f можно обращаться как к вектору, например, $f[1] := a$, что равносильно $B[n:, 1] := a$ или $B[m, 1] := a$.

<ОП формы> присваивает динамическому массиву, указанному слева от знака ($=:$), форму (т.е. размеры по соответствующим измерениям) величины, указанной справа от знака. Так, оператор $A =: C[*]$ означает, что динамический массив А приобретает размеры вектора С. Отметим, что значение массива А размещается в отведенной этому массиву памяти и не связано, вообще говоря, со значением массива С. Массивы А и С имеют только одинаковые размеры.

<ОП значения и формы> объединяет функции двух предыдущих операторов. Здесь вычисляется и значение, и форма результата. Последняя определяется формой величин, фигурирующих в выражении, или данными. После выполнения оператора

$A * 12[*] := (1:6, 3:5, 4:1 * 2:4, 4:2 * 1:7, 2:8)$

значением неправильного массива $A * 12[*]$ становится сокращенная запись матрицы

$$A12 = \begin{pmatrix} 6 & 0 & 5 & 1 \\ 0 & 4 & 0 & 2 \\ 7 & 8 & 0 & 0 \end{pmatrix},$$

при этом организуется цепное хранение неправильных компонент. Неправильные векторы в данных отделяются друг от друга одной звездочкой, матрицы - двумя звездочками и т.д.

5.2. Оператор перехода

<оператор перехода> ::= на <метка>
<метка> ::= <имя>

Примеры

на α

на ω

5.3. Условный оператор

$\langle \text{условный оператор} \rangle ::= \langle \text{отношение} \rangle ? \underline{\text{да}} \langle \text{метка} \rangle$
 $\langle \text{отношение} \rangle ::= \langle \text{объект или пусто} \rangle \langle \text{операция отношения} \rangle$
 $\langle \text{объект} \rangle$
 $\langle \text{объект или пусто} \rangle ::= \langle \text{объект} \rangle | \langle \text{пусто} \rangle$
 $\langle \text{пусто} \rangle ::=$

Примеры

$\tau \neq S \quad ? \underline{\text{да}} \quad \omega$
 $Q[n] \in Q[k, *] \quad ? \underline{\text{да}} \quad \omega 1$
 $\underline{\text{лист}} P[m] \quad ? \underline{\text{да}} \quad \omega 2$

Семантика

$\langle \text{Условный оператор} \rangle$ передает управление на указанную в нем метку в том случае, когда значение данного отношения есть истина.

5.4. Оператор настройки

Синтаксис

$\langle \text{Оператор настройки} \rangle ::= \langle \text{настройка} \rangle | \langle \text{настройка} \rangle \langle \text{оператор перехода} \rangle$
 $\langle \text{настройка} \rangle ::= \underline{\text{настроить}} \langle \text{очередной} \rangle \langle \text{срезка} \rangle$
 $\langle \text{срезка} \rangle ::= \langle \text{имя общее} \rangle [\langle \text{граничный ряд} \rangle]$
 $\langle \text{граничный ряд} \rangle ::= \langle \text{индексный ряд} \rangle | \langle \text{граничная пара} \rangle |$
 $\langle \text{граничный ряд} \rangle, \langle \text{граничный ряд} \rangle$
 $\langle \text{граничная пара} \rangle ::= \langle \text{константа или скаляр} \rangle : \langle \text{индекс} \rangle$
 $\langle \text{константа или скаляр} \rangle ::= \langle \text{константа} \rangle | \langle \text{скаляр} \rangle$

Примеры

$\underline{\text{настроить}} \underline{\text{очередной}} \ 2 \ A1 * K[* , *]$
 $\underline{\text{настроить}} \underline{\text{очередной}} \ 2 \ B[n : * , *] \ \underline{\text{на}} \ \alpha$
 $\underline{\text{настроить}} \underline{\text{очередной}} \ S[2 : m]$

Семантика

$\langle \text{Оператор настройки} \rangle$ определяет начало и конец циклической процедуры выборки очередной компоненты данного массива и настраивает эту процедуру на выборку указанной начальной

компоненты. После исчерпания указанных компонент выполняется (если он указан) <оператор перехода>. Так, в первом примере оператор настраивает процедуру выборки строк из массива $A1 \times K$ на выборку первой строки, а во втором - на выборку n -ой строки из матрицы B . При следующих обращениях к очередной компоненте матрицы B будут выбираться строки $n, n+1$ и т.д. После того, как будет выбрана последняя строка из B , результатом обращения к очередной компоненте B будет служить переход на α .

5.5. Оператор цикла

Синтаксис

<оператор цикла> ::= <заголовок цикла> { <тело цикла> }
 <заголовок цикла> ::= для <элемент цикла> цикл
 <элемент цикла> ::= <ЭЛ 1> | <ЭЛ 2> | <ЭЛ 3>
 <ЭЛ 1> ::= <скаляр> := <правая часть>
 <правая часть> ::= <граничная пара> | <граничная пара>
 <шаг>
 <шаг> ::= шаг <константа или скаляр>
 <ЭЛ 2> ::= <ЭЛ 1> пока <отношение>
 <ЭЛ 3> ::= <ряд скаляров> := <очередной элемент>
 <ряд скаляров> ::= <скаляр> | <ряд скаляров>, <скаляр>
 <очередной элемент> ::= <очередной> <справочный элемент>
 <справочный элемент> ::= <справочный объект> | <справочная срезка>
 <справочная срезка> ::= <ряд характеристик> <срезка>

Примеры

<ЭЛ 1> для $i := 1 : m$ цикл
 для $i := n : m$ шаг k цикл
 <ЭЛ 2> для $i := 1 : m$ пока $a \neq b$ цикл
 для $i := n : *$ шаг k пока $a > b$ цикл
 <ЭЛ 3> для q, f := очередной 2 признак справка
 $\Lambda * P [*, *]$ цикл
 для i, j := очередной индекс значение $J [*, *]$
 цикл
 для i, j := очередной индекс значение $J [r, *]$
 цикл

для $p :=$ очередной $S[2:K]$ цикл

Семантика

<Оператор цикла> организует многократное циклическое выполнение операторов, указанных в <теле цикла>. Оператор с элементом типа <ЭЛ 1> и <ЭЛ 2> традиционен. Оператор цикла с элементом типа <ЭЛ 3> совмещает в себе функции <оператора настройки> (5.4) и <оператора присваивания> (5.1).

5.6. Операторы размещения

Синтаксис

<оператор размещения> ::= <OP1> | <OP2>
<OP1> ::= <указатель размещения> <компонента массива> |
<указатель размещения> <оператор присваивания>
<указатель размещения> ::= включить | разместить
<OP2> ::= исключить <компонента массива>

Примеры

<OP1> разместить $Y * n[*]$::= $C[*] \circ BO[*], [*]$
включить $A * q[m:,*]$::= $C[*]$
разместить $A \uparrow * n[*,*]$::= $A * K[*,*]$
<OP2> исключить $A * q[n:,*]$

Семантика

Области, статические массивы, наборы, схемы, скаляры и константы размещаются в памяти статическим образом. Эти объекты становятся активными сразу после их описания (т.е. их элементы доступны для выборки или присвоения им значения). Хранением же динамических массивов с изменяющимися размерами управляют <операторы размещения>. В сплошной области динамические массивы размещаются в специальном режиме магазина. После описания динамические массивы остаются пассивными, а область, в которой они размещаются, считается свободной. Если оператор размещения с указателем разместить относится к пассивному массиву, этому массиву отводится (согласно его форме в этот момент) часть свободной памяти в области, а сам массив ста-

новится активным. Свободная часть области соответственно уменьшается. Если оператор разместить относится к активному массиву, его значение хранится на прежнем месте, а все расположенные за размещаемой компонентой данные становятся пассивными. Оператор разместить, отнесенный к области, делает ее целиком свободной, а все размещенные в ней массивы — пассивными.

В цепной области организуется цепное хранение компонент динамических массивов, управляемое оператором размещения с указателем включить или исключить. Область организуется в цепь из сплошных компонент заданной размерности. Массив активируется первым применением к нему оператора включить. Включаемой компоненте отводится свободная компонента области из ее цепи. Оператор исключить возвращает компоненту (или массив) в свободную область. Оператор включить отнесенный к области, возвращает ее к исходному виду, делая пассивными все массивы, размещенные в ней.

5.7. Оператор процедуры

$\langle \text{оператор процедуры} \rangle ::= \langle \text{имя процедуры} \rangle \mid \langle \text{имя процедуры} \rangle$
 $(\langle \text{параметры} \rangle)$
 $\langle \text{имя процедуры} \rangle ::= \langle \text{имя} \rangle$
 $\langle \text{параметры} \rangle ::= \langle \text{объект} \rangle \mid \langle \text{параметры} \rangle, \langle \text{объект} \rangle$

Примеры

$T(i, k)$
 $\Lambda \lambda 1$
 $ABC(D[*], f)$

Семантика

$\langle \text{Оператор процедуры} \rangle$ имеет традиционную интерпретацию. Предполагается, что скалярные величины передаются в процедуру значением, а более сложные — ссылкой.

§ 6. Описания

$\langle \text{описание} \rangle ::= \langle \text{описание набора} \rangle \mid \langle \text{описание области} \rangle \mid$
 $\langle \text{описание ленты} \rangle \mid \langle \text{описание массива} \rangle \mid \langle \text{описание сложного}$
 $\text{массива} \rangle \mid \langle \text{описание схемы} \rangle \mid \langle \text{описание процедуры} \rangle \mid \langle \text{описание}$

начальных данных>

6.1. Описание набора, области и ленты

Синтаксис

<описание набора> ::= набор <сегмент набора>
<сегмент набора> ::= <ряд имён> [$\langle \text{длина} \rangle, \langle \text{разрядность} \rangle$]
<сегмент набора>, <сегмент набора>
<ряд имён> ::= <имя> | <ряд имен>, <имя>
<длина> ::= <индекс>
<разрядность> ::= <константа>
<описание области> ::= <область> | <область> цепь <цифра>
<область> ::= область <имя области> [$\langle \text{размеры} \rangle$]
<имя области> ::= <имя общее>
<размеры> ::= <константа> | <имя набора> | <размеры>,
<размеры>
<описание ленты> ::= лента <цифра> [$\langle \text{имя набора} \rangle$]

Примеры

набор $S[p, 6], J, K[m, 12]$
область $t[10, 15]$
область $\Lambda O * [F, F1]$ цепь 2
лента I [M]

Семантика

Наборы - это векторы, элементы которых плотно упакованы в памяти [5]. Набор $S[p, n]$ состоит из p элементов, значение каждого элемента размещается на n разрядах машинного слова. Семантические пояснения <описания области> имеются в 5.6 и 6.2.

6.2. Описание массива

Синтаксис

<описание массива> ::= <массив> | <массив> <место>
<массив> ::= массив <сегмент массива>
<сегмент массива> ::= <ряд имен массива> [$\langle \text{форма} \rangle$]
<сегмент массива>, <сегмент массива>

$\langle \text{ряд имен массива} \rangle ::= \langle \text{имя общее} \rangle \mid \langle \text{ряд имен массива} \rangle,$
 $\langle \text{ряд имен массива} \rangle$
 $\langle \text{форма} \rangle ::= \langle \text{индексный ряд} \rangle \mid \langle \text{имя набора} \rangle \mid \langle \text{форма} \rangle,$
 $\langle \text{форма} \rangle$
 $\langle \text{место} \rangle ::= \underline{\text{лента}} \langle \text{цифра} \rangle \mid \underline{\text{барабан}} \langle \text{цифра} \rangle \mid \underline{\text{область}}$
 $\langle \text{имя области} \rangle$

Примеры

массив $x, g[m], AO, CC[m_0], BO[m_0, m_0]$
массив $B*[M, M]$ барабан I
массив $t*[M], u*[F^2]$
массив $A*[N, *:]$ лента I

Семантика

$\langle \text{Описание массива} \rangle$ определяет форму массива и способ размещения в памяти. Описание относится либо к одному массиву, либо к ряду отдельных независимых массивов, либо, наконец, к набору массивов, образующих отдельную область. Последнему случаю соответствует специальное имя в описании массива. Так, описание массив $t*[M]$ означает, что в области t^* размещаются массивы t^*1, t^*2, \dots, t^*k , размеры которых определяются набором $M = (m_1, m_2, \dots, m_k)$. Описание массив $t*[k, M]$ определило бы неправильную матрицу t^* из k строк с длинами m_1, m_2, \dots, m_k соответственно. Описание массив $t*[n, M]$ определяет в отличие от предыдущего набор из k матриц с размерами $n \times m_1, n \times m_2, \dots, n \times m_k$. Описатель места типа лента или область определяет размеры памяти, отводимой данному набору массивов. Если такого описателя нет, массивы обязаны быть статическими (т.е. не иметь в индексном ряду своего описания символа $*$), а размеры области складываются из размеров массивов, образующих набор. При этом сплошные массивы размещаются подряд один за другим. Неправильные компоненты каждого массива организуются в свою цепь, т.е. в начале каждой компоненты имеется ссылка на следующую. Помимо доступа к элементам отдельных массивов возможен доступ к элементам области как единого целого. Так, описание массив $t*[M]$ порождает наряду с объектами вида $t^*m[i]$ также и объекты вида $t^*[i]$.

Разделитель $*$ в индексном ряду описания массива указывает на то, что размеры массива по данному измерению могут изменяться по ходу дела, т.е. массив является динамическим. Разделитель $:$ в индексном ряду указывает на то, что компоненты массива по данному измерению снабжаются признаками. Так, из описания массива $A*[N, *:]$ ленты I следует, что область $A*$ составляют матрицы $A*k$, число строк в которых задается набором N , а длина строк может быть переменной. При этом значении элементов являются числа с признаками. Общий объем необходимой памяти определяется описанием ленты I. Рассмотрим другой пример. Из описания

массив $A*[*: , M]$ область AO

следует, что динамические матрицы $A*k$ с переменным числом строк размещаются в области AO , при этом каждая строка снабжается признаком. Если описание области содержит дополнение цепь, т.е. имеет вид

область $AO[m, mO]$ цепь 2,

то это значит, что область организована в цепь из m строк длины mO , и из этой цепи поставляется необходимая память под строки матриц $A*k$, т.е. строки матриц $A*k$ хранятся попеременно в области AO , и для каждой матрицы $A*k$ организуется своя цепь строк. Распределение памяти в такой цепной области организуется операторами включить и исключить. В силовой области, описание которой не содержит дополнения цепь, размещение динамических массивов организуется оператором разместить.

6.3. Описание сложного массива

Синтаксис

\langle описание сложного массива $\rangle ::=$ массив-уровень $\langle CM \rangle |$
массив-цепь $\langle CM \rangle$
 $\langle CM \rangle ::= \langle CM1 \rangle | \langle CM1 \rangle \langle$ место \rangle
 $\langle CM1 \rangle ::= \langle$ специальное имя $\rangle [\langle$ форма $\rangle]$ схема \langle имя
схемы \rangle

Примеры

массив-уровень $\Lambda^*[*, M]$ схема Q область ΛO^*
массив-цепь $Y^*[M]$ схема Q
массив-цепь $A1^*:[MN]$ схема Q

Семантика

Сложные массивы представляют собой специальные области и наборы массивов, связанные со схемами. Пусть с некоторой схемой Q связывается набор массивов $Y^*[M]$ так, что массив Y^*_k отвечает вершине $Q[k]$. Тогда описание массив-цепь $Y^*[M]$ схема Q определяет область Y^* , предназначенную для одновременного размещения массивов, лежащих на любом пути схемы Q . Максимальное число массивов, хранящихся одновременно в области Y^* , равно числу уровней схемы Q . При этом имя Y^*_1 относится к тому массиву из пути, который лежит на первом уровне, Y^*_2 относится к массиву второго уровня и т.д. Размеры области определяются по схеме Q и набору M . Естественно, что массивы Y^*_k являются динамическими, их хранением управляет оператор разместить. Символ : в описании массив-цепь $A1^*:[MN]$ схема Q указывает на то, что каждый массив $A1^*_k$ из области снабжается признаком. Таким естественным признаком может служить, например, признак вершины, отвечающей данному массиву из цепи. Перейдем теперь к другому сложному массиву.

Описание массив-уровень $\Lambda 1^*[M, M]$ схема Q определяет набор областей, по числу уровней схемы Q , в которых размещаются массивы $\Lambda 1^*i[m_i, m_i]$, отвечающие схеме Q : в каждой области размещаются массивы соответствующего ей уровня, нумерация массивов сквозная, т.е. массив $\Lambda 1^*i$ соответствует вершине $Q[i]$. Если в описании массива-уровня нет описателя места, размеры областей определяются по схеме Q и наборам индексного ряда. В этом случае имеем статические массивы. Описатель места обязателен в случае динамических массивов или цепного размещения их компонент. Так, описание

массив-уровень $\Lambda^*[*, M]$ схема Q область ΛO^* и область $\Lambda O^*[F, F]$ цепь 2

определяют следующее размещение информации для схемы Q на

рис. 1. Размеры областей $\Lambda 0*1, \dots, \Lambda 0*4$ задаются наборами F и F^I . Каждая из этих областей организована в цепь равнодлинных строк. В области $\Lambda 0*1$ хранятся строки матрицы $\Lambda *1$, снабженные признаками. В следующей области $\Lambda 0*2$ размещаются массивы $\Lambda *2, \Lambda *3, \Lambda *4$, лежащие на втором уровне. Строки этих массивов, снабженные признаками, вперемежку хранятся в области $\Lambda 0*2$, и т.д.

6.4. Описание схемы

Синтаксис

```

<описание схемы> ::= схема <сегмент схемы> <дисперсия>
<дополнения>
<сегмент схемы> ::= <определение> <схема> | <сегмент
схемы>, <сегмент схемы>
<определение> ::= левая | правая | левая: | правая:
<схема> ::= <имя схемы> | <имя схемы> [ <индекс> ]
<имя схемы> ::= <имя>
<дисперсия> ::= дисперсия <имя набора>
<дополнения> ::= <пусто> | <моменты> | <вес> | <моменты>
<вес>
<моменты> ::= моменты <имя набора>
<вес> ::= вес <константа>

```

Пример

схема правая Q левая P , левая: $PO[r]$ дисперсия
 20 моменты $M1$ вес 4

Семантика

В описании схемы определения правая и левая относятся к ненагруженным схемам, а определения правая: и левая: - к нагруженным. Кодировка признаков вершин схемы определяется конструкцией <дисперсия>. Разрядность веса вершины в нагруженной схеме определяется конструкцией <вес>. Конструкция <моменты> содержит дополнительную информацию, которая используется в правых схемах и может быть получена из анализа схемы. <Индекс> в описании схемы указывает число вершин.

6.5. Описание процедуры

Синтаксис

$\langle \text{описание процедуры} \rangle ::= \langle \text{заголовок процедуры} \rangle$ начало
 $\langle \text{тело процедуры} \rangle$ конец
 $\langle \text{заголовок процедуры} \rangle ::= \langle \text{процедура} \rangle \mid \langle \text{процедура} \rangle$
 $(\langle \text{формальные параметры} \rangle)$
 $\langle \text{процедура} \rangle ::=$ процедура $\langle \text{имя процедуры} \rangle$
 $\langle \text{формальные параметры} \rangle ::= \langle \text{скаляр} \rangle \mid \langle \text{формальные пара-}$
 $\text{метры} \rangle, \langle \text{скаляр} \rangle$
 $\langle \text{тело процедуры} \rangle ::= \langle \text{оператор} \rangle \mid \langle \text{тело процедуры} \rangle ;$
 $\langle \text{оператор} \rangle$

Примеры

процедура $T(i, k)$ начало $\langle \text{тело процедуры} \rangle$ конец
процедура $\Lambda \lambda 1$
начало $SS[*] := \Lambda * \text{ко} [i I: , *] ;$
 $SS[0] := SS[0] - I ;$
для $i, f :=$ очередной 2 признак справка $\Lambda * \text{ко} [* , *]$
цикл
 $\{ d := u[*i] ; d = 0? \text{ да } \omega 6 ;$
 $f[*] := f[*] + d \times SS[*] ; \omega 6 : \}$ конец

6.6. Описание начальных данных

Синтаксис

$\langle \text{описание начальных данных} \rangle ::= \langle \text{основное описание} \rangle =$
 $\langle \text{данные} \rangle$
 $\langle \text{основное описание} \rangle ::= \langle \text{описание константы} \rangle \mid \langle \text{описание}$
 $\text{набора} \rangle \mid \langle \text{описание схемы} \rangle \mid \langle \text{описание массива} \rangle$
 $\langle \text{описание константы} \rangle ::=$ константа $\langle \text{ряд имен} \rangle$

Примеры

константа $p_0, p_0, m, n, m_0 = (13, 4, 32, 40, 4);$
набор $D_0, M_1[4, 3] = (1, 3, 3, 2), (1, 3, 7, 2);$
схема правая $Q[ro]$ дисперсия D_0 моменты $M_1 = (0001,$
 $0011, 0012, 0013, 0111, 0112, 0113, 0121, 0122, 0131, 0132,$
 $1211, 1212);$

§ 7. Пример

Для иллюстрации предлагаемого языка приведем запись на этом языке алгоритма из [6]. Имена объектов в записи алгоритма для облегчения ориентации приближены к обозначениям в работе [6]. Так, исходной матрице $A[M, N]$ отвечает набор массивов $A*[N, *:]$, а отдельным блокам $A[M_k, N_k]$ отвечают массивы $A**k[*:, *]$. Матрице $B[M, M]$ соответствует набор массивов $B*[M, M]$, матрице $\Lambda[\mathcal{J}, \mathcal{J}]$ - массив-уровень $\Lambda*[*:, M]$. Преобразованию $T[\mathcal{J}, \mathcal{J}]$ соответствует процедура T , векторам x, y, \mathcal{J} - массивы x, y, \mathcal{J} и т.п. В качестве примера выбрана матрица A , блочная структура которой показана на рис. 2. Порядок в множестве номеров блоков исходной матрицы задается схемой, изображенной на рис. 1. Наборы массивов упорядочены в соответствии с правой схемой Q (табл. I). Вид исходной информации проиллюстрирован на примере записи блока A 12 в 5.1. Структура других объектов частично также рассматривалась ранее в семантических пояснениях. На рис. 2 условно показаны области $A*, t*, u*, B*, AO*1, AO*2, AO*3$ и $AO*4$.

ОПИСАНИЯ:

константа $p_0, n_0, m, n, m_0, m_1 = (13, 4, 32, 40, 4, 28)$;
набор $M, N, \mathcal{D}, \mathcal{D}1, F, F1, F2, MN[*:, 6] =$
 $(2, 3, 4, 3, 2, 2, 2, 4, 2, 1, 2, 3, 2), (40, 12, 17, 11, 4,$
 $2, 6, 13, 4, 5, 6, 4, 9), (2, 5, 9, 12, 14, 16, 18, 22, 24, 25, 27, 30,$
 $32) (0, 0, 12, 29, 0, 4, 6, 12, 25, 29, 34, 12, 16), (0, 4, 4, 3), (0, 3,$
 $13, 28), (2, 3, 4, 3, 2, 2, 2, 4, 2, 1, 2), (...)$;
схема правая $Q[p_0]$ дисперсия $\mathcal{D}0$ моменты $M1 = (0001,$
 $0011, 0012, 0013, 0111, 0112, 0113, 0121, 0122, 0131, 0132,$
 $1211, 1212)$;

лента $I [MN]$;

набор $S[p_0, 8], \mathcal{J}, k[\mathcal{J}, 6], PQ[p_0, 12]$;

массив $S[n_0, m], C[n], x, y[m] AO, CSEm_0,$

$BO[m_0, m_0]$

массив $A*[N, *:]$: лента I ;

массив $B*[M, M]$ барабан I ;

область $AO*[F, F1]$ цепь 2 ;

массив-уровень $\Lambda*[*:, M]$ схема Q область $AO*$;

массив-цепь $Y*[M]$ схема Q ;

МАССИВ-ЦЕПЬ $A1^* : [M, N]$ СХЕМА Q ;
МАССИВ $t^*[M]$, $u^*[F2]$;

ПРОЦЕДУРЫ:

ПРОЦЕДУРА $T(\varphi, \eta)$

начало $t^*[*] := 0$; $t^*[\eta] := 1$; ЛИСТ $Q[\varphi]$? ДА Ω ;
 $\sigma := Q[\varphi]$ $S[*] := \text{вЕС } \uparrow PQ(\sigma)$;
 $s := S[1]$; $\tau, \ell := \text{КООРД } D[\eta]$; $\tau \neq s$? ДА ω ;

ДЛЯ $q, f := \text{очередной } 2 \text{ признак значение } \Lambda^*S[*], \ell$

ЦИКЛ

$\{t^*[q] := -f\}$;

ω : ДЛЯ $p := \text{очередной } S[2: *]$ ЦИКЛ
 $\{ \text{ДЛЯ } q, f := \text{очередной } 2 \text{ признак справка } \Lambda^*p[*], * \}$

ЦИКЛ

$\{t^*[q] := t^*[q] + t^*[p[*]] \circ f[*]\}$;

Ω : конец T

ПРОЦЕДУРА $ВИ(\varphi, \zeta)$

начало $ВОГ[*], * := В * \varphi[*], * ;$
 $\gamma_1 := В[\zeta]$; $\gamma := 1/\gamma_1$; $В[\zeta] := \gamma_1 - 1$;
 $СС[*] := ВОГ\zeta, *$;

ДЛЯ $j, f := \text{очередной } 2 \text{ индекс справка } ВОГ^*, * \text{ ЦИКЛ}$
 $\{ f[*] := f[*] - \gamma \times В[\zeta] \times СС[*] \}$;
 $В * \varphi[*], * := ВОГ^*, * ;$ конец

ПРОЦЕДУРА $ВИ1(\xi)$

начало $k_1, \ell_1 := \text{КООРД } D[\xi]$; $ВОГ^*, * := В * k_1[\xi], * ;$
 $СС[*] := u * k_1[\xi] \circ ВОГ^*, * ;$
 $\gamma := 1 + СС[\ell_1]$; $\gamma := 1/\gamma$;
 $ВОГ[\ell_1], * := ВОГ[\ell_1, *] - \gamma \times СС[*]$;
 $В * k_1[\xi], * := ВОГ^*, * ;$ конец

ПРОЦЕДУРА $\Lambda \lambda 1$

начало $СС[*] := \Lambda * КО[\lambda 1, *]$; $СС[\ell_0] := СС[\ell_0] - 1$;
ДЛЯ $i, f := \text{очередной } 2 \text{ признак справка } \Lambda * КО[*], * \text{ ЦИКЛ}$
 $\{ do := u * [\lambda i] ; do = 0 ? \text{ да } \omega 6 ;$

$f[*] := f[*] + d0 \times CC[*]$; $\omega 6 :$ } конец

процедура $\Lambda \lambda 2$

начало $CC[*] := \Lambda * \pi[i1, *]$;

для $i, d0 :=$ очередной индекс значение $u[*]$ цикл
{ $d0 = 0?$ да $\omega 7; 7; \Lambda * \pi[i, *] := \Lambda * \pi[i, *] +$
 $+ d0 \times CC[*]$; $\omega 7 :$ } конец

процедура $\Lambda \lambda 3$

начало для $i, f :=$ очередной 2 индекс справка $\Lambda * \kappa 1[*]$, $*$

цикл { $d0 := u[*i] - u * \kappa 1[*] \circ f[*]$;
 $f[i1] := f[i1] + a * d0$; } конец

АЛГОРИТМ:

начало $PQ\{*\} :=$ лев: $Q\{*\}$;

Вычисление $Y :$

$\alpha :$ начало для $k :=$ очередной вес $PQ[*]$ цикл

{ $\gamma :=$ вес $Q[k, *]$; для $i := 1 : m$ цикл { $CC := t * k[*]$;
 $CC[*] := 0$; $T(k, i)$; $CC[*] := CC[*] + C[\gamma, i] \times t * k[*]$; };

$VO[*] := V * k[*]$; разместить $Y * \gamma[*] := CC[*] \circ VO[*]$;

признак $A1 * \gamma = k ?$ да $\omega 0$;

разместить $A1 * \gamma[*]$ $:= A * k[*]$; признак $A1 * \gamma := t$.

$\omega 0 :$ лист $Q[k]$? да $\omega 2$;

Вычисление $C :$

для $i, j :=$ очередной индекс значение $J[*]$ цикл

{ $kj := k J[i]$; $Q[k] \neq Q[kj, *]$? да $\omega 1$;

$j := j - D1[k]$; $C[\gamma - 1, i] := C[\gamma, i] - Y * \gamma[*] \circ A1 * \gamma[j, *]$;

$\omega 1 :$ } ; на $\omega 3$;

Проверка $C < 0$:

$\omega 2 :$ $v[*] :=$ индекс $Q[k, *]$; для $\tau :=$ очередной $v[*]$

цикл { $\mu :=$ вес $Q[\tau, *]$; $n1 := D1[k] - D1[\tau]$;

настроить очередной 2 $A1 * \mu[n1, *]$; };

для $j :=$ очередной 2 индекс $A1 * \gamma[*]$, $*$ цикл

{ $n\tau := D1[k] + j$; $CO := CC[n\tau]$; для $\tau :=$ очередной $v[*]$

цикл { $\mu :=$ вес $Q[\tau, *]$;

$f :=$ очередной 2 справка $A1 * \mu[*]$;

$CO := CO - Y * \mu[*] \circ f[*]$; } ; $CO < 0?$ да $\alpha 0$; };

$\omega 3$: ; СТОП; КОНЕЦ α

Вычисление λ по цепи ν :

$\alpha 0$: начало $g[*] := 0$; $q := k$;
 $СС[*] := BO[*], *] \circ A1 * \nu[*], *]$;
для $\tau :=$ очередной $\nu[2: *]$ цикл $\{ \nu :=$ вес $Q[\tau, *]$;
 $j := n\tau - D1[\tau]$; включить $\Lambda * q[m1: *]$ $:= СС[*]$;
 $AO := t * \tau[*]$; $AO[*] := 0$;
для $i, a :=$ очередной индекс значение $g[*]$ цикл
 $\{ T(q, i)$; $g[i] := a + t * q[*] \circ СС[*]$; $np := \nu[i] - D1[\tau]$;
 $AO[*] := AO[*] - A1 * \nu[np, *] * g[i]$;};
 $BO[*], *] := B * \tau[*], *]$; $AO[*] := AO[*] + A1 * \nu[j, *]$;
 $СС[*] := BO[*], *] \circ AO[*]$; $q := \tau$;};

• Довычисление g и определение $j 0$:

$\varepsilon := 0$; для $i, a :=$ очередной индекс значение $g[*]$ цикл
 $\{ T(p, i)$; $g[i] := a + t * p[*] \circ СС[*]$;
 $a \leq 0$? да $\omega 4$; $a := x[i] / a$; $\varepsilon \leq a$? да $\omega 4$;
 $\varepsilon := a$; $i 0 := i$; $\omega 4$: ; $j 0 := \nu[i 0]$;
 $ko, lo :=$ коорд $D[i 0]$;

Выбор рабочей леммы:

$\beta := u[*] := 0$; $i 1 := m 1$;
для $j, a :=$ очередной 2 признак значение $\Lambda * ko[*], lo$ цикл
 $\{ u[*], j] := a$; $a = 0$? да $\omega 5$; $i 1 \leq j$? да $\omega 5$;
 $i 1 := j$; $\omega 5$:}; $i 1 = m 1$? да $\alpha 1$; на $\alpha 2$; конец $\alpha 0$

Лемма I:

$\alpha 1$: начало

Исправление Λ :

для $\tau :=$ очередной индекс $Q[k, ko)$ цикл
 $\{$ исключить $\Lambda * \tau[i 0: *]$; признак $2 \Lambda * \tau[m 1: *]$ $:= i 0$;};

Исправление B :

$B :=$ справка $\Lambda * ko[m 1: *]$; $B(ko, lo)$;

Доисправление Λ :

для $\tau :=$ очередной индекс $Q[ko, *)$ цикл
 $\{$ исключить $\Lambda * \tau[m 1: *]$;};

Исправление x :
 $x[*] := x[*] - \varepsilon \times g[*]$; $x[i0] := \varepsilon$;

Изменение базиса :
 $k \mathcal{J}[i0] := k$; $\mathcal{J}[i0] := n \tau$; $C[\mathcal{J}, i0] := C C[n \tau]$;
на a ;
конец $\alpha 1$

Лемма 2 :

$\alpha 2$: начало $a := -u[*i1]$; $u[*i1] := 1 - a$;
 $u[*] := u[*] / a$;

Исправление B :

$B :=$ справка $\Lambda * k o \mathcal{J}[i1, *]$; $B \mathcal{J}(k o, \ell o)$; $B \mathcal{J} 1(i1)$;

Исправление Λ :

$Q[k o] \neq Q[k, *]$? да $\omega 8$; $\Lambda \mathcal{J} 1$;
для $\tau :=$ очередной индекс $Q(k o, k 1)$ цикл $\{\Lambda \mathcal{J} 2(\tau)\}$;
 $\omega 8$: $Q[k 1] \in Q[k, *]$? да $\omega 9$; $\Lambda \mathcal{J} 3$;
 $\omega 9$: $k j 0 := k \mathcal{J}[i0]$; $k j 1 := k \mathcal{J}[i1]$;
 $Q[k k, k o] := Q[k j 0, k o) \cap Q[k j 1, k o)$;
для $\tau :=$ очередной индекс $Q[k k, k o)$ цикл
 $\{$ признак $2 \Lambda * \tau [i0, *]$ $:= 0$; признак $2 \Lambda * \tau [i1, *]$ $:= i0$;
признак $2 \Lambda * \tau [o, *]$ $:= i1$; $\}$;
для $\tau :=$ очередной индекс $Q[k j 0, k k)$ цикл
 $\{$ признак $2 \Lambda * \tau [i0, *]$ $:= i1$; $\}$;
для $\tau :=$ очередной индекс $Q[k j 1, k k)$ цикл
 $\{$ признак $2 \Lambda * \tau [i1, *]$ $:= i0$; $\}$;

Изменение базиса :

$\mathcal{J}[i0] := j1$; $\mathcal{J}[i1] := j0$; $n n := k \mathcal{J}[i0]$;
 $k \mathcal{J}[i0] := k \mathcal{J}[i1]$; $k \mathcal{J}[i1] := n n$;
 $C[n o, i0] := C C[j1]$; $C[n o, i1] := C C[j0]$;
 $k o := k 1$; $i0 := i1$; $\ell o := \ell 1$; $j0 := j1$; на B
конец $\alpha 2$
конец АЛГОРИТМА

Л и т е р а т у р а

1. Канторович Л.В., Перспективы работы в области автоматизации программирования на базе крупноблочной системы. Труды МИАН СССР, 96 (1968), 5-15.

2. Нахбазян К.В. и Лебединский М.М., Функциональный алгоритмический язык. Там же, 16-47.
3. Iverson K.E., A Programming Language. John Wiley and Sons, inc. N. York, London, 1962.
4. Универсальный язык программирования *PL/I*. "Мир", Москва, 1968 (перев. с англ.).
5. Катков В.Л., Морозов В.П., Поттосин И.В., Пар А.Ф., Семенова Л.Я., Хоперсков А.Е. ЭПСИЛОН-система автоматизации программирования задач символьной обработки. Новосибирск, "Наука", 1972.
6. Звягина Р.А., Об общем методе решения задач линейного программирования блочной структуры. Оптимизация, I (1971), 22-40.

Поступила в ред.-изд. отд.
30.У. 1972 г.

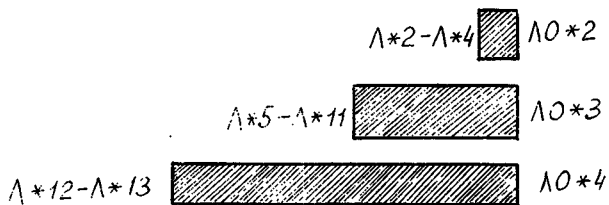
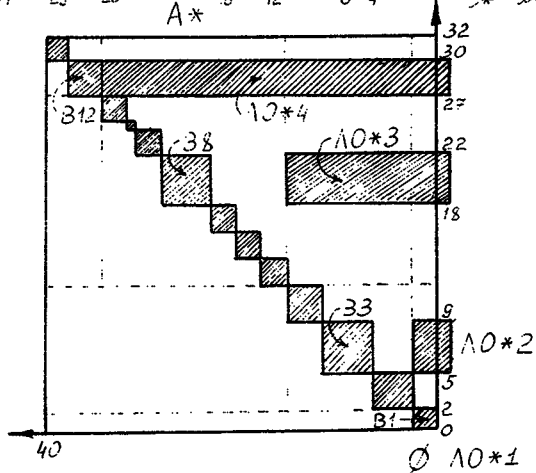
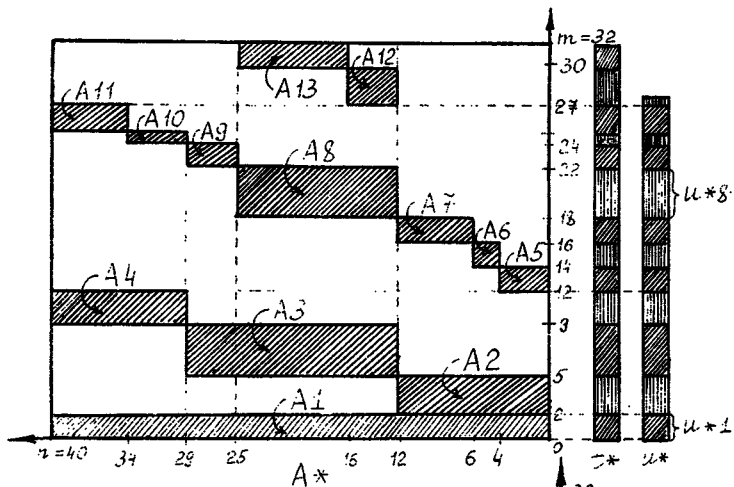


Рис. 2