

ВХОДНОЙ ЯЗЫК МНОГОПУЛЬТОВОЙ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ ДЛЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ

В. Д. Кожухов

Х а р ь к о в

В качестве базового языка вычислительной машины для инженерных расчетов может быть принят любой универсальный алгоритмический язык, пригодный для описания алгоритмов вычислительной математики. Представляется возможным построить входной язык машины на основе алгоритмического адресного языка [1]. Выбор этого языка в качестве базового обусловлен тем, что из разработанных к настоящему времени алгоритмических языков адресный язык наиболее полно удовлетворяет требованиям, предъявляемым к базовому языку [2].

Адресами во входном языке машины являются множество ячеек P_i ($i = 1, 2, \dots$), содержащих исходные числовые данные для решения задач, а также множество рабочих ячеек r_i ($i = 1, 2, \dots$). В языке имеется набор основных операций и указаны допустимые отношения между ними. Из набора основных операций с помощью допустимых отношений строятся дополнительные операции. Часть дополнительных операций можно сделать сменными, что позволит при необходимости вводить во входной язык новые операции.

ОСНОВНЫЕ ОПЕРАЦИИ И ДОПУСТИМЫЕ ОТНОШЕНИЯ

Штрих операция ' $r_i = a$ — операция извлечения содержимого адреса первого ранга.

Операция два штриха '' $r_i = 'r_j = a$ — операция извлечения содержимого адреса второго ранга.

Арифметические операции над адресами. Допускается выполнение арифметических операций $+$; $-$; $\times$; $::$ над адресами, над адресом и целым числом, над адресом и содержимым адреса. Результатом операции является адрес.

Операции над содержимым адресов. Допускается выполнение операций $+$; $-$; $\times$; $::$; $\uparrow$; e^x ; $\ln x$; $\sin x$; $\cos x$; $\arcsin x$; $\arccos x$; $\arctg x$ над содержимым адресов первого и второго ранга. Результатом операции является содержимое адреса.

Операция засылки. Для изменения содержимого адресов применяется операция засылки, символ которой $\Rightarrow$. После выполнения ' $r_1 \Rightarrow r_2$; ' $r_1 = 'r_2$.

Символ $\leftarrow$. Символ $\leftarrow$ ставится, когда хотят присвоить содержимое какому-либо адресу. Вообще говоря, запись $\bar{r}_1 \leftarrow a$ эквивалентна записи $a \Rightarrow r_1$, однако, символ $\leftarrow$ условимся ставить только тогда, когда вычислитель присваивает адреса исходным данным задачи перед занесением алгоритма, а символ $\Rightarrow$ ставится только внутри алгоритма.

Допустимые суперпозиции. Допускаются суперпозиции арифметических операций над адресами, а также суперпозиции операций, введенных как операции над содержимым адресов.

Модуль. Допускается некоторые алгебраические выражения и содержимое адресов заключать под знак модуля.

Скобки. Допускается некоторые алгебраические выражения заключать в скобки.

Строки. Запись алгоритма в языке машины состоит из строк. Строки записываются одна за другой. Каждая строка содержит одно алгоритмическое действие. При выполнении строки алгоритма, представляющей собой алгебраическое выражение (суперпозиция операций, введенных как операции над содержимым адресов или адресами), операции выполняются с соблюдением порядка выполнения операций, принятых в математике. Для того, чтобы алгебраическое выражение можно было записывать строкой, принято операцию деления записывать в виде $a : b$, а выражение a^b — в виде $a \uparrow b$, где $\uparrow$ — символ операции возведения в степень; выражение e принято записывать в виде e^a , где e^a — символ операции. Действие алгоритма выполняется в порядке записи строк, однако этот порядок может быть нарушен с помощью допустимых переходов по меткам.

Точка. Символ $\cdot$ ставится в конце каждой строчки алгоритма.

Метка. Метки обозначаются $M_i (i = 1, 2, \dots)$. Метка ставится перед строкой, а также может составлять отдельную строку. Если метка ставится перед строкой, то после нее ставится троеточие. Если метка составляет отдельную строчку, то троеточие после нее не ставится, и в этом случае она выполняет операцию безусловного перехода на строку, помеченную такой же меткой.

Предикат. Операция условного перехода выполняется с помощью предикатной формулы PLM_iM_j ,

где P — символ операции;

L — высказывание вида, $f_1 = f_2$, где символ $=$ может быть заменен любым из символов $<$; $\leq$; $\geq$; $>$;

f_1 и f_2 — некоторые алгебраические выражения над содержимым адресов или над адресами;

M_i и M_j — метки.

Допускается любую часть высказывания L заключать под знак модуля. Предикатная формула вычисляет значение L и в случае выполнения условия L передает управление строке M_i . Если условие L не выполняется, управление передается строке M_j . Если в случае истинности условия L следует перейти к следующей за предикатом строке алгоритма, метка M может опускаться.

Операция останов. Символ операции останов $!$ ставится в конце алгоритма. В результате выполнения этой операции машина прекратит выполнение алгоритма для данного пункта вычислителя, перепишет результаты вычислений из ОЗУ и БЗУ и примется за решение задачи для другого пульта.

ДОПОЛНИТЕЛЬНЫЕ ОПЕРАЦИИ

Постоянные дополнительные операции

Групповые операции. Групповые операции реализуются с помощью операций циклирования. Первая групповая операция

$$\text{Ц}_1 x_0 y_0 h_x h_y n M_i M_j,$$

где Ц_1 — символ первой групповой операции;

x_0, y_0 — начальные значения аргументов x и y ;

h_x, h_y соответственно шаг по аргументу x и y ;

число n показывает, сколько раз необходимо повторить алгоритм области действия операций циклирования.

Область действия операции начинается со следующей за операцией циклирования строки и заканчивается на метке M_i . Первый раз алгоритм выполняется со значениями переменных, равными соответственно x_0 и y_0 , во второй раз — со значениями $x_0 + h_x$; $y_0 + h_y$ и т. д., всего n раз. После окончания групповой операции происходит переход к строке алгоритма, помеченной меткой M_j .

Допускается модификация первой групповой операции:

$$\Pi_1 x_0 0 h_x 0 n M_i M_j,$$

$$\Pi_1 0 y_0 0 h_y n M_i M_j,$$

т. е. разрешается производить циклирование с изменением только одного аргумента.

Вторая групповая операция:

$$\Pi_2 x_1 x_2 \dots x_n M_k y_1 y_2 \dots y_n M_i M_j,$$

где Π_2 — символ операции;

n — число различных аргументов;

$x_1, x_2, \dots, x_n$ — набор значений аргументов x ;

$y_1, y_2, \dots, y_n$ — набор значений аргументов y ;

M_i, M_j, M_k — метки.

Вторая групповая операция повторяет n раз выполнение алгоритма в пределах области действия групповой операции с различными значениями переменных x и y . Первый раз алгоритм выполняется со значениями x_1, y_1 , второй раз с x_2, y_2 и т. д. до x_n, y_n . Область действия групповой операции начинается со следующей за операцией строки и заканчивается на метке M_i . После окончания групповой операции происходит переход к строке алгоритма, помеченной меткой M_j .

Допускается модификация второй групповой операции:

$$\Pi_2 x_1 x_2 \dots x_n M_k M_i M_j;$$

$$\Pi_2 M_k y_1 y_2 \dots y_n M_i M_j.$$

R -операции введены в язык машины для удобства обработки графической информации [3]:

$$R — \text{конъюнкция } x \wedge_\alpha y = \frac{1}{2} (x + y - \sqrt{x^2 + y^2 - 2\alpha xy});$$

$$R — \text{дизъюнкция } x \vee_\alpha y = \frac{1}{2} (x + y + \sqrt{x^2 + y^2 - 2\alpha xy});$$

$$R — \text{отрицание } \bar{x} = -x,$$

где x и y — значения аргументов;

α — параметр, который изменяется в пределах $-1 < \alpha \leq 1$.

Во входном языке R -операции записываются следующим образом:

R — конъюнкция $\wedge_\alpha xy \alpha$, где $\wedge_\alpha$ — символ операции;

R — дизъюнкция $\vee_\alpha xy \alpha$, где $\vee_\alpha$ — символ операции;

R — отрицание $\bar{x}x$, где $\bar{x}$ — символ операции.

Сменные дополнительные операции

Их в язык машины введено 9. Это соответственно операция 1, операция 2 и т. д., операция 9. Конкретный смысл операции зависит от той подпрограммы, которой в настоящее время присвоен этот номер. Такими операциями могут быть: вычисление определителя n -го порядка, вычисление определенного интеграла методом Симпсона, операция линейного интегрирования функции $f(x)$ по таблице с постоянным шагом и т. п.

Операции и команды внешних устройств [4]

О п е р а ц и я « п е ч а т а т ь ». Печ. $r_1 r_2 \dots r_n M_i$. Печатается содержимое ячеек $r_1 r_2 \dots r_n$ на том пульте вычислителя, задача которого решалась.

О п е р а ц и я « ч е р т е ж ». Черт. $r_1 r_2 \dots r_n M_i$. С помощью УВЫГ выводятся точки, координаты которых хранятся в адресах $r_1 r_2 \dots r_n$.

К о м а н д а « в о з в р а т » применяется при неправильном занесении информации с пульта вычислителя в БЗУ. При подаче с пульта команды «возврат» происходит стирание информации о последнем занесенном символе. Команду «возврат» можно применять несколько раз подряд.

К о м а н д а « с ч е т » разрешает машине начать выполнение алгоритма, занесенного с пульта вычислителя, с которого подана команда «счет».

Язык отладки

О т л а д о ч н ы е м е т к и. При вводе входной информации после заполнения каждой восьмой ячейки памяти на бланке контроля ввода печатается символ «/» (наклонная палочка), а после заполнения некоторого стандартного массива («страницы») на бланке печатается символ «*».

К о м а н д а « у д а л и т ь / $a_1 * a_2 / a_3 * a_4$ ». По этой команде после подачи команды «счет» в массиве памяти, начиная с a_1 -й по счету «палочки» a_2 -й «страницы» и кончая a_3 -й «палочкой» a_4 -й «страницы», будут записаны нули.

К о м а н д а « в с т а в и т ь / $b_1 * b_2$ ». В результате выполнения этой команды заносимая после нее входная информация будет записываться, начиная с ячейки, следующей за $/b_1 * b_2$. В рассматриваемой машине применяется безадресный метод записи программ. Если после ячейки, следующей за $/b_1 * b_2$, нет свободного места для записи вставляемого массива, машина сдвинет информацию, записанную после ячейки, следующей за $/b_1 * b_2$, и в освободившееся место запишет вводимую информацию.

К о м а н д а « п у с к $M_i M_j T_i$ ». Здесь M_i и M_j — метки, T_i — метка времени. По этой команде машина после поступления команды «счет» выполняет алгоритм, начиная с метки M_i и кончая меткой M_j . T_i ограничивает работу алгоритма по времени. Величину T_i можно принять фиксированной, например, в пределах $T_i = 1 \text{ сек}$; $T_2 = 10 \text{ сек}$; $T_3 = 1 \text{ мин}$; $T_4 = 10 \text{ мин}$; $T_5 = \infty$.

ЛИТЕРАТУРА

1. Е. Л. Ю щ е н к о. Адресное программирование. Гостехиздат УССР, 1963.
2. А. Е. Ку л и н к о в и ч, Е. Л. Ю щ е н к о. О базовом алгоритмическом языке. «Кибернетика», 1965, № 2.
3. В. Л. Р в а ч е в. Геометрические приложения алгебры логики. Изд-во «Техника», Киев, 1966.
4. В. П. С е м и к, И. В. С е р г и е н к о. О некоторых принципах построения отладки программ. «Кибернетика», 1966, № 3.