В.Ф.ТУРЧИН

РЕФАЛ - МАКРОКОД

(Москва)

РЕФАЛ-МАКРОКОД - это система программирования, которую можно определить (если использовать имеющие хождение в программистской среде термины) любым из следующих двух способов:

I. Система макрогенерации, в которой языком описания макрокоманд служит язык РЕФАЛ [I] и в которой определен набор базовых макрокоманд, рекомендуемых пользователю в качестве макрокоманд нижнего уровня.

2. Расширяемый язык программирования, включающий в себя три компонента: базовый язык L_0, являющийся языком машины или ассемблера; язык первого уровня L_1, являющийся в большой степени машинно независимым; метаязык - РЕФАЛ, служащий для определения специализированных языков высокого уровня.

Это двоякое определение системы РЕФАЛ-МАКРОКОД надо понимать не только формально - в том смысле, что введение каждой макрокоманды есть введение (или, по крайней мере, расширение) некоего специализированного языка, но и содержательно - в том смысле, что РЕФАЛ, будучи языком символьных преобразований, дает возможность без труда описывать сложные преобразования текстов-аргументов макрокоманд, тем самым стирая грань между определением макрокоманды и определением языка.

Чтобы продемонстрировать, как выглядят определения макрокоманд на РЕФАЛе, мы возьмем простой пример, который фигурировал на этом семинаре в докладе В.С.Штаркмана. Надо ввести макрокоманду ЗЕТ, которая осуществляет сложение и вычитание произвольного числа чисел, символические адреса которых указываются идентификаторами, и послать результат по указанному

адресу (идентификатору). Аргумент макрокоманды записывается в формате, который лучше всего пояснить примером макровызова:

$$\mathcal{S}'\text{ET}\ \llcorner\!\lrcorner\ \text{X=}\ \text{A} + \text{B1} - \text{B2} + \text{Д}\ \llcorner\!\lrcorner$$

Аргумент ограничивается с двух сторон проблемами; адрес назначения (левая часть оператора присваивания) отделяется знаком равенства; наличие первого идентификатора (А - в нашем примере) обязательно ; остальные компоненты предваряются знаком + или -.

Макрорасширение этого вызова в автокод БЕМШ (для машины БЭСМ-6) должно иметь вид:

СЧ, А ;
СЛ, В1;
ВЫЧ, В2;
СЛ, Д;
ЗП, Х;

Назовем МАКРО рефал-функцию, которая осуществляет макрорасширение одной макрокоманды. Это значит , что конкретизация

$$\mathcal{k}\ \text{'МАКРО'}\,\mathcal{E}_1$$

где $\mathcal{E}$ - любой макровызов, порождает соответствующее макрорасширение. Описание функции с детерминативом 'МАКРО' будет содержать предложения, относящиеся к различным макрокомандам. Наша задача - написать предложения, относящиеся к микрокоманде $\mathcal{S}'$ET.

Как мы построили приведенное выше макрорасширение? Сначала записали команду считывания первого аргумента; затем записали группу команд, осуществляющих " добавление" к содержимому сумматора остальных аргументов; наконец, приписали команду записи. Так как число аргументов произвольно, для осуществления "добавления " нам понадобится вспомогательная рекурсивная функция. Присвоим ей детерминатив 'ДОБ'. Предложение функции 'МАКРО' для

макрокоманды SET запишется следующим образом:

$$k\,'МАКРО'\,S'ET\sqcup e_2 = e_1\,S(+-\sqcup)_2\,e_2 \sim СЧ,\ e_1;$$
$$k\,'ДОБ'\,S_2\,e_2 \perp$$
$$ЗП,\ e_a;$$

Вспомогательную функцию ДОБ опишем предложениями:

$$k\,'ДОБ'+e_1\,3(+-\sqcup)_2\,e_2 \sim СЛ,e_1;\ k\,'ДОБ'\,3_2\,e_2 \perp$$
$$k\,'ДОБ'-e_1\,3(+-\sqcup)_2\,e_2 \sim ВЫЧ,e_1;\ k\,'ДОБ'\,S_2\,e_2 \perp$$
$$k\,'ДОБ'\sqcup \sim$$

Таким образом, полное описание макрокоманды SET осуществляется четырьмя предложениями на РЕФАЛе, которые в наглядной форме показывают и формат обращения к макрокоманде (левые части предложений), и ее смысл (правые части).

Приведенные предложения записаны на " рефал-скорописи", которая используется в процессе разработки алгоритмов, а также как язык публикаций . При вводе в машину, разумеется, приходится отказаться от строчных букв и от индексов. В рефал-системе для машины БЭСМ-6 ввод осуществляется в так называемом " метакоде -Б". Для примера приведем описание в метакоде-Б функции ДОБ:

```
ДОБ '+' E1 S'(+-␣)Z E2='СЛ,'E1';' К/ДОБ/SZE2
'-' E1 S'(+-␣)Z E2='ВЫЧ,'E1';' К/ДОБ/SZE2
'␣'=
```

Итак, транслятор с РЕФАЛа можно с успехом использовать в качестве макрогенератора. Однако РЕФАЛ-МАКРОКОД, включает в себя кроме базового языка L_0 и метаязыка - РЕФАЛа еще язык первого уровня L_1 , который служит связующим звеном между машинным языком L_0 и высокоуровневыми языками. Его назначение - избавить создателей высокоуровневых специализированных

языков от необходимости изучать систему команд машины и сделать высокоуровневую часть нашей системы в значительной степени машинно-независимой. В то же время нельзя забывать и об эффективности использования машины, поэтому необходим некий компромисс. В языке L_1 этот компромисс имеет следующий характер: мы сохраняем особенности размещения информации в оперативной памяти и на внешних носителях, но отвлекаемся от всех особенностей системы команд, таких как адресность, способ индексации и т.п. Мы исходим при этом из того, что правильное размещение информации в соответствии со структурой машины, является совершенно необходимым для получения эффективных программ, и автоматизировать процесс размещения информации - задача весьма трудная. В то же время, при заданном размещении информации автоматизировать процесс выбора подходящих команд для выполнения указанных преобразований - задача гораздо более простая, она-то и решается при переводе с L_1 на L_0.

В настоящее время идет работа по реализации РЕФАЛ-МАКРОКОДа для машины Минск-32, поэтому дальнейшее изложение будет ориентировано на эту машину.

Человек, который программирует на языке L_1 для Минск-32, должен знать, что оперативная память этой машины состоит из 37-разрядных ячеек, каждая из которых может использоваться либо как одно слово, либо как последовательность пяти 7-разрядных символов, расположенных в первых 35 разрядах ячейки. Кроме этого он может, в принципе, не знать о машине ничего. Доступ к различным участкам памяти осуществляется в языке L_1 с помощью адресных выражений, а передача управления - с помощью перехода по меткам.

Адресные выражения образуются из имен (идентификаторов в

смысле АЛГОЛа-60), натуральных чисел, знаков операций +, -, ж, /, ↑ и обращений к макрофункциям. В качестве средства указания структуры выражения служат скобки ( ).

С каждым именем в процессе исполнения программы связывается два значения: первичное значение (или адрес) и вторичное значение (или просто значение). Имена делятся на места и указатели. Место имеет постоянное первичное значение, определяемое в момент компиляции - адрес ячейки. Вторичное значение - содержимое этой ячейки может конечно, меняться в процессе исполнения программы. Место определяется с помощью макрокоманды МЕСТО, например:

.МЕСТО. X, XI

Имена X и XI получают первичные значения и для них резервируется по одной ячейке.

Макрокоманда

I6. МЕСТО. ТЕКСТ

резервирует поле из I6 ячеек под имя ТЕКСТ; первичным значением имени ТЕКСТ становится адрес первой ячейки поля, вторичным значением - содержимое этой ячейки (а не всего поля:вторичное значение - это всегда содержимое одной ячейки).

Последнюю макрокоманду можно записать также в виде

. МЕСТО. (I6) ТЕКСТ

Указатель имеет переменное первичное значение. В соответствии со структурой памяти и команд машины Минск-32 указатели бывают двух типов: указатель ячейки и указатель символа. Первичным значением указателя ячейки является натуральное число в диапазоне $0 \div 2^{16} - 1$, интерпретируемое как адрес ячейки. Первичным значением указателя символа являются пара чисел: адрес ячейки и номер символа в ячейке; вторая компонента лежит в диапазоне $0 \div 4$.

Программист, пишущий на языке L_1 может, в принципе, не знать, каким образом фиксируются в машине текущие первичные значения (адреса) указателей. В действительности есть два способа хранения: в ячейке основной области памяти и в индексной ячейке. В обоих случаях адрес ячейки хранится в поле Δ_2 (разряды 2I-36), адрес символа следующим образом: адрес ячейки - в поле Δ_2, а номер символа - в поле Δ_1 (разряды 5-20). Указатели описываются с помощью макрокоманды УКАЗ; например,

.УКАЗ. Х,У,У2

определяет перечисленные три имени как указатели ячейки. Макрокоманда

С.УКАЗ.SI, S2

вводит два указателя символа. Программист может потребовать, чтобы указатель хранился в памяти или в индексе; для этого надо в первом аргументе макрокоманды дописать букву П или И, соответственно. Например,

ПС.УКАЗ. I, 2

В случае индекса можно также указать номер индексной ячейки, в котором надо хранить адрес.

Если никаких указаний о хранении первичного значения указателя не дано, то размещение производится автоматически в соответствии со следующим алгоритмом.

Среди макрокоманд языка L_1 есть две: *BEGIN* и *END*, с помощью которых вводится блочная структура аналогично тому, как в АЛГОЛе-60. При входе в первый блок те указатели, которые распределяются автоматически, помещаются в свободные индексные ячейки. Если свободные индексные ячейки исчерпаны, то оставшиеся регистры размещаются в памяти. Входя во

внутренний блок, транслятор с L_1 пытается все указатели, описанные в блоке и подлежащие автоматическому размещению, поместить в индексные ячейки. Сначала он использует свободные индексные ячейки. Если они кончаются, то транслятор выбирает указатель, занимающий индексную ячейку и описанный раньше всех остальных, и перемещает его в основную память. При этом записывается команда ПИА, выполняющая необходимую пересылку адреса. Освобожденная индексная ячейка занимается указателем, описанным во внутреннем блоке. Эта процедура повторяется пока не будут исчерпаны описания указателей. При выходе из блока индексные ячейки, занятые под локальные (описанные в нем) указатели, объявляются свободными. Если при входе в блок производилось перемещение нелокальных указателей, то записываются команды ПАИ, восстанавливающие их в своих старых индексных ячейках.

Описанный алгоритм отдает наибольшее предпочтение указателям, описанным в самом внутреннем блоке. Благодаря этому циклы в различных структурных единицах программы можно программировать независимо, используя в качестве переменной цикла локальный в этой структуре указатель. Тогда при вложении структурных единиц одна в другую наиболее внутренние циклы, требующие наиболее быстрой реализации, будут управляться переменными, размещенными в индексных ячейках.

Адресные выражения (АВ) делятся, как указатели, на АВ ячейки и АВ символа. Один из основных синтаксических принципов РЕФАЛ-МАКРОКОДа гласит, что тип адресного выражения определяется не по его виду, а по его вхождению, иначе говоря при описании макрокоманды должно быть указано для каждого аргумента, - АВ, является ли этот аргумент адресным выражением

ячейки или адресным выражением символа. Если конструкция АВ не соответствует требуемому типу, выдается сообщение о синтаксической ошибке.

Знаки +, - и * в АВ ячейки интерпретируются как операции сложения, вычитания и умножения по модулю 2^{16}. Знак / изображает деление нацело с округлением в сторону возрастания. Знаки и / в АВ символе не допускаются, а операции сложения и вычитания определяются естественным способом, описанным в руководстве по машине Минск-32. Все перечисленные операции применяются только к операндам того типа, к которому принадлежит АВ. Поэтому, например, конструкция

X + $I

где имя X описано как указатель ячейки, а $I - как указатель символа, заведомо является синтаксически ошибочной, куда бы она ни входила.

Знак ↑ в АВ обоих типов изображает одноместную операцию перехода от АВ ячейки к ее содержимому, интерпретируемому как АВ ячейки или символа в зависимости от вхождения. По старшинству операции располагаются следующим образом: теснее всего связывает знак одноместной операции ↑, затем идут знаки * и / и, наконец, знаки + и -. О вхождении макрофункций в АВ будет сказано ниже.

Примеры адресных выражений, в которых предполагается, что имена MAC и N описаны как места X1, I и J - как указатели ячейки, а $I - как указатель символа:

X1

MAC + ↑N * (I-1) + J-1

$I + ↑(X1 4)

Здесь первые два выражения могут встречаться только в позиции АВ ячейки, а третье - АВ символа. Четвертое АВ может встретиться в обеих позициях, причем в позиции АВ символа оно будет истолковано как пара чисел: 5 (адрес ячейки) и 3 (номер символа).

Основная структурная единица РЕФАЛ-МАКРОКОДА - это макровыражение. Семантически, макровыражение - это текст, которому соответствует некоторая программа на базовом языке L_0. Эта программа называется переводом макровыражения. Перевод макровыражения $\mathcal{E}$ есть результат конкретизации k 'МАКВ' $\mathcal{E}$.

Таким образом, транслятор с РЕФАЛ-МАКРОКОДа есть не что иное как описание рефал-функции МАКВ (макровыражение). Перевод макровыражения определяется в процессе компиляции. Он не является однозначным, а зависит, вообще говоря, от предшествовавших макрокоманд, и в первую очередь, конечно, от описаний указателей.

Кроме перевода, макровыражение обладает значением, которое определяется как то слово (37-разрядное, то есть типа ячейки), которое вырабатывается на сумматоре машины в результате выполнения перевода макровыражения. Значение макровыражения зависит, разумеется, от состояния памяти машины в процессе выполнения программы. Оно может быть и неопределенным; в частности, не определено значение тех макровыражений, которые имеют пустой перевод.

Макровыражение есть либо макрофункция, либо последовательность макрофункций, разделенных точкой с запятой, либо макровыражение, заключенное целиком в скобки. Макровыражение

может быть также пустым. Вместо термина макрофункция мы будем иногда использовать термин макрокоманда, не делая между ними формального различия. Первый термин удобен тогда, когда надо подчеркнуть наличие значения. Во втором случае подчеркивается эффект выполнения макровыражения; его удобно использовать, когда значение не определено или не существенно. Перевод последовательности макрофункций есть последовательность переводов ее компонент. Таким образом, если последняя макрофункция макровыражения обладает значением, она будет значением всего макровыражения.

Синтаксическим признаком макрофункции является наличие на основном уровне скобочной структуры обозначения функции-конструкции вида $.\mathcal{F}.$, где $\mathcal{F}$ – произвольная последовательность знаков, не содержащая скобок и точек. Для формального описания синтаксиса макровыражений и макрофункций приведем первые предложения транслятора, описывающие функцию 'МАКВ'

k'МАКВ' $(e_1) \sim$ k'МАКВ' $e_1 \perp$

k'МАКВ' $e_f.e_2 \sim$ k'МАКВ1' $(.e_f.)e_2 \perp$

k'МАКВ' $e_1 e_f e_2 \sim$ k'МАКВ1' $(.e_f.(e_1))e_2 \perp$

k'МАКВ' $\sim$

k'МАКВ' $e_1 \sim k$'П' ОШИБКА в МАКРОВЫРАЖЕНИИ: $e_1 \perp$

k'МАКВ1' $(e_1)e_2, e_3 \sim k$'МАКВС' $e_1 e_2 \perp k$'МАКВ' $e_3 \perp$

k'МАКВ1' $(e_1)e_2 \sim k$'МАКВС' $e_1 e_2 \perp$

Первое предложение показывает, что макровыражение можно заключить в скобки целиком, от этого оно не перестает быть макровыражением и не меняется его перевод- а следовательно, и значение. Если выражение не имеет вида ( ), то на основ-

ном уровне скобочной структуры отыскивается первое слева обозначение макрофункции. Если таковое не найдено и выражение не пусто, то оно не является правильным макровыражением и выдается сообщение об ошибке. Таким образом, синтаксическим признаком макровыражения, как и макрофункции, является наличие обозначения макрофункции на основном уровне скобочной структуры (то есть не входящего в скобки).

Первое из них определяет первую макрофункцию, входящую в макровыражение; оно простирается до ближайшей точки с запятой, если таковая имеется, а иначе - до конца макровыражения. Поиск точки с запятой осуществляется функцией'МАКВ 1'.

Функция'МАКФС'- "макрофункция в стандартном формате "- это основная рефал-ФУНКЦИЯ, служащая для определения макрофункций. Макрофункции могут использоваться в двух форматах:

$$\mathcal{F} \cdot \mathcal{E} \qquad (1)$$

и

$$\mathcal{E}_1 \cdot \mathcal{F} \cdot \mathcal{E}_2 \qquad (2)$$

Если макрофункция имеет один аргумент, то естественно записывать ее в формате (I). Тогда,как видно из описания функции'МАКВ', перевод макрофункции сведется к конкретизации

$$k\ \text{'МАКФС'} \cdot \mathcal{F} \cdot \mathcal{E}_1$$

Следовательно, в числе предложений, определяющих функцию МАКФС , должно быть предложение с соответствующей левой частью. Например, макрофункция МЕТКА используется в виде

$$, \text{МЕТКА} \cdot \mathcal{M}$$

где $\mathcal{M}$ - метка. Соответствующее предложение в описании функции МАКФС имеет левую часть

$$k\ \text{'МАКФС'}\ \text{.МЕТКА}.\mathcal{e}_m \sim$$

Если макрофункция имеет два аргумента, то естественно записывать ее в формате (2); тогда, в частности, для макрофункций, выражающих операции, мы получаем привычную алгебраическую запись, например:

XI . + . X2

Здесь .+. - обозначение макрофункции сложения целых чисел, XI и X2 - ее аргументы (адресные выражения). Перевод этой макрофункции сведется к конкретизации

k'МАКФС'.+. (XI) X2 ⊥

В общем случае перевод макрофункции в формате (2) требует конкретизации

k'МАКФС'.$\mathcal{F}$.($\mathcal{E}_1$)$\mathcal{E}_2$ ⊥

таким образом, преобразование к стандартному формату состоит в том,что обозначение функции выносится на первое место, а первый аргумент заключается в скобки, чтобы отделить его от второго аргумента. Именно в этом формате и следует описывать такие макрофункции. Например, левая часть предложения, описывающего макрофункцию . + . , имеет вид

k'МАКФС'. + .(e_1)e_2 ⌣

Если макрофункция зависит от нескольких аргументов,то ее можно записывать как в первом,так и во втором формате.В первом случае $\mathcal{E}$ будет последовательностью термов, представляющих собой аргументы, взятые в скобки (последний аргумент можно в скобки не заключать). Во втором случае будет представлять первый аргумент, а - последовательность остальных аргументов. Заметим, впрочем, что и в случае макрофункций от двух аргументов программист может, если сочтет это удобным, использовать формат (I), заключая первый аргумент в скобки, например:

.+.(XI)X2

При этом не требуется, очевидно, вносить каких-либо изменений в функцию 'МАКФС'.

Теперь мы можем закончить описание адресных выражений. Адресное выражение может быть, в частном случае, макровыражением. Тогда его значением является значение макровыражения, интерпретируемое как адрес ячейки или символа. Макровыражение (в частности, макрофункция) заключенное в скобки, может также входить в АВ в качестве подвыражения. Например, макрофункция ПОЛЕ определена таким образом, что значением макровыражения

5-20 .ПОЛЕ. X

является слово, образованное сдвигом разрядов 5-20 ячейки с адресом X в крайнее правое положение и обнулением остальных разрядов. Это макровыражение может занимать позицию адресного выражения ячейки. Оно может быть также использовано для образования более сложных адресных выражений, например

(5-20 .ПОЛЕ. X)*4N+J

РЕФАЛ-МАКРОКОД, как и всякий расширяемый язык, может быть описан только частично - в своей основе. Расширение РЕФАЛ-МАКРОКОДа производится пополнением набора предложений, описывающих функцию 'МАКФС'. Различные специализированные языки могут быть получены удалением одних макрофункций (предложений функции 'МАКФС') и добавлением других. Основа РЕФАЛ-МАКРОКОДа - язык L_1 содержит около тридцати макрофункций. Рамки настоящего доклада не позволяют останавливаться на каждой из них. Мы отметим лишь особенности макрокоманд перехода, а затем на нескольких примерах дадим общее представление о макрофункциях уровня L_1 .

При описании системы автоматического распределения индексных ячеек (индексных регистров в других машинах) под указатели мы видели, что при входе в блок, способ хранения значения указателя может измениться. При выходе из блока через его END восстанавливается *status quo* Это означает, что выходы из блока по метке должны быть либо запрещены, либо подвергаться серьезному анализу на предмет восстановления способа размещения указателей. В РЕФАЛ-МАКРОКОДе мы идем по первому пути, сильно ограничивая использование операторов перехода в духе идей структуризованного программирования. Итак, выход из блока иначе чем через его конец запрещается. Метка описывается с помощью макрокоманды МЕТКА, например

. МЕТКА .L4

которая не соответствует никакому действию, а лишь помечает следующую за ней макрокоманду. Аргументом макрокоманды безусловного перехода НА может быть только метка, причем она должна быть описана в том же самом блоке, иначе будет выдана распечатка о синтаксической ошибке. Такими же возможностями обладают макрокоманды условного перехода (см.примеры ниже)

Примеры макровыражений

I. .A. X = У + 3

Это макрокоманда присваивания адресов (типа ячейка) Указатель X приобретает первичное значение , равное первичному значению указателя У, увеличенного на 3.

2. .I. X = У

Присваивание (вторичных) значений. В ячейку, на которую указывает X, засылается содержимое ячейки, на которую указывает У.

3. .АС. $S1$= .СЛЕД.$S2$

Присваивание адресов символов. Адрес $S1$ становится равным значению макрофункции СЛЕД от аргумента $S2$.

4. .МЕСТО. N ;

. ↑ . N = (AI .+. BI) .ж. B2

Описание места и присваивание значений. В ячейку N засылается результат вычисления выражения в правой части. Макрокоманды +, - и ж - действия над целыми числами.

5. X,У . $DELA.1$

Приращение адресов. Первичные значения указателей X и У увеличиваются на единицу.

6. .А. X = (.А.У = 0)

Присваивание адресов и значений - это макрофункции, которые своим значением имеют значение правой части. Следовательно, X, как и У принимает значение 0.

7. .БУКВА. $S1$. + НА. L

Условный переход. Проверяется, указывает ли $S1$ на букву. При положительном ответе - переход на метку L .

8. . А . (X.$DELA$ .1)< ↑N . - НА . ЦИКЛ

Сравнение адресов и условный переход. Значение адреса X увеличивается на единицу. Проверяется, меньше ли новое значение, чем содержимое ячейки N . При отрицательном ответе переходим на метку ЦИКЛ.

9. .УСЛ. (.БУКВА. $S1$)$S1.DELAS.1$,
(.ЦИФРА. $S1$)$S1.DELAS.1$,
( ) . НА . L 2

Условное выражение. Пустое условие интерпретируется как тождественно выполненное.

IO. .УСЛ.((.БУКВА. $S1$). ИЛИ. (.ЦИФРА.$S1$)) $S1.DELAS.1$,
( ) .НА. L2

То же, что в предыдущем примере, записанное через логическую связку ИЛИ.

ЛИТЕРАТУРА

I. В.Ф.Турчин. Программирование языка РЕФАЛ. Препринты Институт Прикладной Математики АН СССР, №№ 41,43,44,48, 49 за 1971 г.

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР АКАДЕМИИ НАУК
ГРУЗИНСКОЙ ССР

Т Р У Д Ы
ВСЕСОЮЗНОГО СЕМИНАРА
ПО ВОПРОСАМ МАКРОГЕНЕРАЦИИ

ТБИЛИСИ
1975

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР АКАДЕМИИ НАУК
ГРУЗИНСКОЙ ССР

Т Р У Д Ы

ВСЕСОЮЗНОГО СЕМИНАРА ПО ВОПРОСАМ

МАКРОГЕНЕРАЦИИ

ТБИЛИСИ
1975

Редактор В.М.Курочкин
Техредактор Н.Н.Окуджава

Сдано в набор 24.I.75. Подписано к печати 6.I.75;Формат бумаги 60X90^{I}/I6; Печатных л. I3-5 ; 2у.-издат.л. II;

УЭ 00901 Тираж 600 ; Заказ 251 ;

Цена I руб.I0 коп.

Издательство "Мецниереба",380060,Тбилиси,ул.Кутузова,I9
Типография АН ГССР, 380060, Тбилиси, ул.Кутузова,I9