

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

ДУБНА



Ц 8408
3-141

22/14-75
11 - 9109

В.А.Загинайко, М.Ю.Попов

3604 / 2-75

МАКРОГЕНЕРАТОР С ОПТИМИЗАЦИЕЙ
ПОИСКА МАКРОПРЕДЕЛЕНИЙ
В ПРОЦЕССЕ ТРАНСЛЯЦИИ

1975

11 - 9109

В.А.Загинайко, М.Ю.Попов

**МАКРОГЕНЕРАТОР С ОПТИМИЗАЦИЕЙ
ПОИСКА МАКРООПРЕДЕЛЕНИЙ
В ПРОЦЕССЕ ТРАНСЛЯЦИИ**

Объединенный институт
ядерных исследований
БИБЛИОТЕКА

I. Основные определения и постановка задачи.

Целью настоящей работы является описание алгоритма работы макрогенератора, реализованного авторами на ЭВМ БЭСМ-6.

За основу был взят макрогенератор, разработанный одним из авторов на ЭВМ М-20^{/1/}. Входной язык и язык макроопределений рассматриваемого в данной работе транслятора совпадает с языком, описанным в работе ^{/1/}, а также языком "МАКРОС" системы математического обеспечения ЭВМ БЭСМ-4 ^{/2/}.

Вначале необходимо напомнить основные положения работы ^{/1/}. Текст программы пользователя, написанной на языке "МАКРОС", представляет собой набор операторов, отделяемых друг от друга символом ";" (точка с запятой). Оператор может быть помеченным или непомеченным. Метка представляет собой идентификатор, отделенный от остальной части оператора двоеточием (как в языке АЛГОЛ-60). Операторы рабочей программы будем называть макровывозами. В процессе трансляции макровывозов преобразуется в последовательность операторов автокода, называемую в дальнейшем макрорасширением данного макровывоза.

Для организации процесса получения автокодного текста макрогенератору (транслятору с языка "МАКРОС") необходимо задать информацию о способе трансляции конструкций языка "МАКРОС", т.е. макроопределения этих конструкций. Макроопределения в данном случае имеют вид таблицы соответствий между операторами языка "МАКРОС" в

некоторой абстрактной форме и результатом трансляции этих операторов в виде последовательности символов автокодного текста.

Непомеченная часть операторов языка "МАКРОС" представляет собой последовательность величин и разделителей. Величина есть последовательность букв и цифр в количестве не более шести (фактически величинами могут быть идентификаторы или целые числа без знака). Разделителем может быть либо знак (+ - * / † , ; ()) либо основной символ, т.е. последовательность любых символов кроме точки, расположенных между двумя точками. Например, в выражении

```
.IF. a+1 .GO TO.S;
```

величинами являются A I S . Разделителями являются .IF. GO TO. (основные символы), а также знаки + и ;

Сущность процесса макрогенерации состоит в следующем. На первом этапе трансляции операторов языка "МАКРОС" (этап выделения понятия) производится выделение величин, входящих в состав макровызова (фактических параметров) в специальный буфер, после чего в тексте макровызова они заменяются комбинацией X# (формальными параметрами). # в данном случае представляет собой цифру - относительный адрес текста фактического параметра в буфере фактических параметров. В результате такой замены получается абстрактная форма оператора программы пользователя.

На втором этапе трансляции производится поиск макроопределения в таблице соответствий, левая часть которого совпадает (как текстовая величина) с абстрактной формой оператора пользователя, полученной на первом этапе.

На третьем этапе (макрогенерация) с помощью правой части найденного на втором этапе макроопределения и таблицы фактических параметров производится выдача макрорасширения, т.е. результата трансляции оператора в виде автокодного текста.

Процесс макрогенерации выполняется следующим образом: происходит посимвольная выдача текста правой части макроопределения на выводной текстовый буфер транслируемой программы. Если при выводе встречается комбинация X#, то вместо нее выдается текст соответствующего фактического параметра. Например, если имеется макровывоз

```
.IF. A = B .GO TO. S;
```

и в таблице соответствий имеется макроопределение

```
.IF. X1 = X2 .GO TO. X3; ,XTA,X1; ,AEX,X2; ,UZA,X3;
```

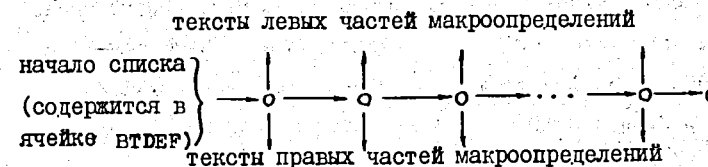
то макрорасширением будет являться следующая последовательность операторов в автокоде "МАКРОС":

```
,XTA,A; ,AEX,B; ,UZA,S;
```

2. Оптимизация поиска макроопределения на втором этапе трансляции

Из рассмотрения процесса трансляции видно, что с ростом числа задаваемых транслятору макроопределений замедляется скорость работы второго этапа трансляции (поиск макроопределения в таблице соответствий). При линейном поиске время возрастает пропорционально количеству макроопределений в таблице.

Следует заметить, что макроопределения могут задаваться транслятору пользователем в том же виде (в смысле набивки), что и программа в языке "МАКРОС". При вводе макроопределений препроцессор транслятора преобразует их в таблицу поиска - список, имеющий следующий вид:



Это сокращает процесс поиска нужного макроопределения. Для более существенного уменьшения времени поиска макроопределения в рассматриваемом варианте транслятора был использован метод автоматической генерации программы поиска макроопределения. Ниже дается описание этого метода.

Получаемая в результате программа поиска (в представлении на языке "МАКРОС") имеет следующий вид:

```

1. { .CALL.TMDEF,ETMDEF;.IF. FMDEF = INF1 .ELSE. LAB1;
    CINF1→BTDEF;
    .PO to .LABEND;
    LAB1: bss,0;
    :
    :
    :
K { .IF. FMDEF = INFK .ELSE. LARK;
    CINFK→BTDEF;
    .GO TO. LABEND;
    LARK:RSS,0;
    :
    :
    :
LABEND: .CALL.SEARC., ESEARC;

```

Подпрограмма (блок) TMDEF читает первую строчку текста макроопределения, полученного на первом этапе, и записывает ее в ячейку FMDEF. Фрагменты программы с номерами 1, 2, ..., K формируются и накапливаются в процессе поиска макроопределений в таблице с новыми начальными фрагментами. Величины CINF1, ..., CINFK, связанные с фрагментами INF1, ..., INFK, характеризуют ссылки на начала фрагментов таблицы поиска, в которых расположены макроопределения, начинающиеся с фрагмента INFK. Подпрограмма SEARC осуществляет просмотр таблицы поиска, начиная с макроопределения, ссылка на которое имеется в ячейке BTDEF. В процессе поиска макроопределе-

ний производится их упорядочение таким образом, чтобы макроопределения, начинающиеся с одного фрагмента, располагались одно за другим (кроме тех макроопределений, которые до данного момента не были использованы в задаче на языке "МАКРОС").

Процесс переупорядочения макроопределений и формирования программы поиска выглядит следующим образом. Накапливается таблица начальных фрагментов макроопределений, уже использованных в процессе трансляции. С каждым начальным фрагментом связана ссылка на начало участка упорядоченных макроопределений в таблице поиска, начинающихся с данного фрагмента. Пусть производится поиск нового макроопределения. В этом случае управление передается на сформированную программу поиска. Если начальный фрагмент первого же найденного макроопределения не совпадает с фрагментом оператора пользователя, выделенным на первом этапе трансляции, в программу поиска вставляется группа команд обработки нового макроопределения. В противном случае несовпадающее макроопределение с тем же самым начальным фрагментом записывается в хвост группы макроопределений с этим начальным фрагментом.

3. Инструкция по пользованию транслятором.

Для организации трансляции перфокарты в задаче пользователя должны располагаться в следующей последовательности:

```

*NAME ...
*ASSIGN TIME....
перфокарты транслятора
*EXECUTE
*MAP
таблица макроопределений
*MAC
программа на языке "МАКРОС"

```

*END

*ASSEMBLER

*READ DRUM

программы, следующие за программой на языке "МАКРОС"

*EXECUTE

информация, обрабатываемая задачей

*END FILE

диспетчерский конец

Перфокарты макроопределений и операторов на языке "МАКРОС" пробиваются в коде УПП или СДС аналогично тому, как готовятся карты на языках автокод или ФОРТРАН. Карта на языке "МАКРОС" должна иметь в конце символ \$ ("доллар") в коде СДС или \diamond ("ромб") в коде УПП. Этим обеспечивается совместимость информации по пробивке с БЭСМ-4, где символам \$ и \diamond соответствует символ ";" ("точка с запятой"). Текстовая карта, у которой отсутствуют упомянутые символы, рассматривается как карта на автокоде "МАДЛЕН". В программе на языке "МАКРОС" допускается наличие в произвольном месте автокодных карт.

Левая и правая части макроопределений пробиваются на отдельных картах, следующих одна за другой. Каждая карта должна заканчиваться символом \$ (\diamond).

В результате трансляции получается программа на автокоде, причем операторы на языке "МАКРОС" выдаются в качестве комментариев к автокодной программе.

ЛИТЕРАТУРА

1. В.А.Загинайко. Инвариантное программирование на машины М-20, Минск-22 и БЭСМ-6. Препринт ОИЯИ, РИ-3993, 1968 г.
2. В.А.Загинайко. Система математического обеспечения БЭСМ-4. Сообщение ОИЯИ, ИИ-6005, Дубна, 1971 г.

Рукопись поступила в издательский отдел
5 августа 1975 года.