

3. Лабораторная работа № 3.

Перевод ОПЗ исходного выражения в текст на выходном языке. Генерация машинного кода

Оглавление

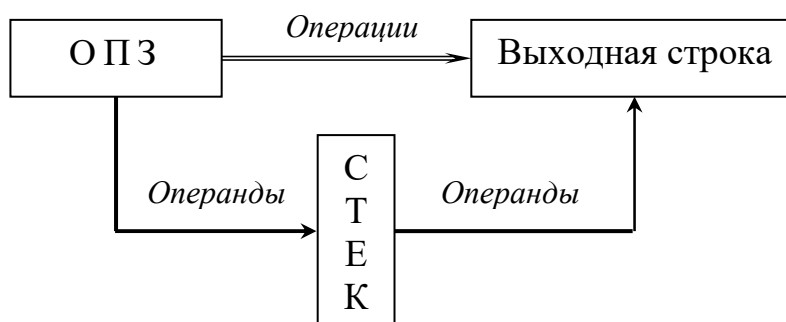
3. Лабораторная работа № 3. Перевод ОПЗ исходного выражения в

текст на выходном языке. Генерация машинного кода	1
3.1. Базовые понятия.....	2
3.2. Правила генерации машинного кода	2
3.3. Комплексный пример перевода ОПЗ исходной программы в машинный код	6
3.4. Задание к лабораторной работе	10

3.1. Базовые понятия

Перевод внутреннего представления программы (ОПЗ) в текст на выходном (машинном) языке представляет собой следующий этап трансляции исходной программы в машинные коды. Для реализации этой процедуры также используется автомат с магазинной памятью (МП-автомат).

Эту процедуру можно схематично представить следующим образом:



Токены внутреннего представления последовательно считываются и обрабатываются автоматом. Токены, представляющие операнды (идентификаторы и константы всех типов), помещаются в стек. Чтение токенов операций из ОПЗ инициирует семантические процедуры, которые извлекают из стека количество операндов, необходимое для выполнения операции, и генерируют соответствующие заготовки машинного кода.

При обработке переменных и констант из таблиц идентификаторов и констант извлекается соответствующая им информация, которая используется для образования правильных адресных частей соответствующих машинных команд.

3.2. Правила генерации машинного кода

Рассмотрим генерацию машинного кода для PL-подобного языка, описанного и разобранного и преобразованного во внутреннее представление в лабораторных работах №№ 1 и 2.

Пусть в качестве машинного языка выступает язык программирования Бейсик. Его особенностью является обязательная нумерация строк и отсутст-

вие символьных меток. Поэтому в операторах перехода в качестве меток используют номера строк, на которые нужно передать управление.

Для построения МП-автомата, переводящего внутренне представление в машинные коды и его семантических процедур, введем ряд внутренних переменных:

P - счетчик вспомогательных переменных;

STR - счетчик строк (этот счетчик вводится ввиду специфики выбранного выходного языка - Бейсика, где обязательна нумерация строк; при трансляции в другие выходные (машинные) языки этот счетчик не нужен).

Кроме того, организуем таблицу меток, которая реализует отображение символьных меток исходного языка в номера строк машинного языка:

Метка	Номер строки
-------	--------------

Эта таблица потребуется в дальнейшем для замены символьных меток на номера строк, что также является особенностью Бейсика как выходного языка.

Так как в языке Бейсик отсутствуют заголовки процедур и операторы описания типов переменных, то при обработке соответствующих элементов из внутреннего представления на выход будем помещать комментарий (REM), обозначающий место этих операторов в исходной программы.

Рассмотрим работу МП-автомата.

1. Если элемент входной строки - идентификатор или константа, то он заносится в стек (в исходном виде, т.е. не условное обозначение, а имя из таблицы идентификаторов или константа из таблицы констант); вспомогательные переменные и константы переносятся без изменения.
2. Если элемент входной строки – число (счетчик операндов), то оно заносится в стек без изменения.

3. При поступлении на вход автомата операции и оператора с определенной арностью (количеством операндов), таких как сложение или присваивание, из стека извлекается необходимое количество операндов и запускается соответствующая семантическая процедура.
4. При обработке операций с непостоянной арностью (например, АЭМ или Ф) работа семантической процедуры начинается с определения арности операции, которая соответствует счетчику операндов, расположенному перед операцией.
5. После выполнения каждой семантической процедуры в выходную строку заносится символ <ВК>, счетчик строк STR наращивается на единицу и заносится в начало новой строки.

Семантические процедуры МП-автомата для операторов и операций входного языка приведены в табл. 3.1.

Таблица 3.1

Лексема	Действия
НП	Извлечь из стека два элемента, занести в выходную строку текст <i>"REM Начало процедуры arg2, arg1"</i>
КП	Занести в выходную строку текст <i>"REM Конец процедуры"</i>
DFD BFD DFT CHAR	Извлечь из стека <i>arg1</i> - число переменных <i>k</i> ; извлечь из стека <i>k</i> аргументов; занести в выходную строку текст <i>"REM Вещественные переменные arg1, arg2,...,argk"</i>
КО	Извлечь из стека два аргумента
УПЛ	Извлечь из стека два аргумента, занести в выходную строку текст <i>" IF NOT(arg2) THEN GOTO arg1"</i>
БП	Извлечь из стека один аргумент, занести в выходную строку текст <i>"GOTO arg1"</i>
:	Извлечь из стека один аргумент, занести в таблицу меток <i>arg1</i> и значение счетчика STR
+ * > <	Извлечь из стека два аргумента, нарастить счетчик вспомогательных переменных Р, занести в выходную строку текст <i>"Rp = arg2 <операция> arg1"</i> , занести в стек Рр.
:=	Извлечь из стека 2 аргумента и занести в выходную строку текст <i>" arg2 = arg1 "</i>

В стеке *arg1* – это элемент, находящийся в вершине стека. Увеличение номера аргумента показывает его удаление от вершины стека и обратно порядку занесения элементов в стек.

При трансляции на языки высокого уровня для арифметических выражений в целях уменьшения количества операторов присваивания и временных переменных возможен вариант формирования строки "(arg2 <операция> arg1)" и занесение ее в стек как единого аргумента для последующих опера-

ций и операторов. Недостатком такого подхода является избыточность круглых скобок в выражениях машинного кода.

В языке Бейсик символьных меток не существует, поэтому необходим дополнительный просмотр сгенерированного текста с целью замены в нем символьных меток на номера строк в соответствии с построенной таблицей меток.

3.3. Комплексный пример перевода ОПЗ исходной программы в машинный код

Рассмотрим комплексный пример генерации машинного кода на языке Бейсик по ОПЗ исходной программы, которая была построена в примере, рассмотренном в лабораторной работе № 2.

MAIN 1 1 НП A1 A2 2 DFD 1 1 КО A1 378 := A2 .73 := CALC2 2 НП
SUM MULT 2 BFD 2 2 КО A1 A2 + 3.2 > M1 УПЛ Р БП M1: SUM A2 A1 + :=
P : MULT A1 A2 SUM + * := КП КП

Последовательность действий МП-автомата изобразим в виде таблицы (табл. 3.2), в строке которой будем записывать состояние стека, счетчика переменных Р, счетчика строк STR после обработки элемента ОПЗ, а также сформированный фрагмент машинного кода.

Таблица 3.2

Шаг	Элемент ОПЗ	Стек	Р	STR	Машинный код
0	-	-	1	1	
1	MAIN	MAIN	1	1	
2	1	1 MAIN	1	1	
3	1	1 1 MAIN	1	1	

Таблица 3.2

Шаг	Эле- мент ОПЗ	Стек	P	ST R	Машинный код
4	НП	-	1	2	1 REM Начало процедуры MAIN
5	A1	A1	1	2	
6	A2	A2 A1	1	2	
7	2	2 A2 A1	1	2	
8	DFD	-	1	3	2 REM Вещественные переменные A1, A2
9	1	1	1	3	
10	1	1 1	1	3	
11	KO	-	1	3	
12	A1	A1	1	3	
13	378	378 A1	1	3	
14	:=	-	1	4	3 A1=378
15	A2	A2	1	4	
16	.73	.73 A2	1	4	
17	:=	-	1	5	4 A2=.73
18	CALC	CALC	1	5	
19	2	2 CALC	1	5	
20	2	2 2 CALC	1	5	
21	НП	-	1	6	5 REM Начало процедуры CALC
22	SUM	SUM	1	6	
23	MULT	MULT SUM	1	6	

Таблица 3.2

Шаг	Элемент ОПЗ	Стек	P	ST R	Машинный код
24	2	2 MULT SUM	1	6	
25	BFD	-	1	7	6 REM Вещественные переменные SUM, MULT
26	2	2	1	7	
27	2	2 2	1	7	
28	KO	-	1	7	
29	A1	A1	1	7	
30	A2	A2 A1	1	7	
31	+	R1	2	8	7 R1=A1+A2
32	3.2	3.2 R1	2	8	
33	>	R2	3	9	8 R2=R1>3.2
34	M1	M1 R2	3	9	
35	УПЛ	-	1	10	9 IF NOT(R2) GOTO M1
36	P	P	1	10	
37	БП	-	1	11	10 GOTO P
38	M1	M1	1	11	
39	:	-	1	11	<i>Занести в таблицу меток пару M1, 11</i>
40	SUM	SUM	1	11	
41	A2	A2 SUM	1	11	
42	A1	A1 A2 SUM	1	11	
43	+	R1 SUM	2	12	11 R1=A2+A1

Таблица 3.2

Шаг	Элемент ОПЗ	Стек	P	ST R	Машинный код
44	:=	-	1	13	12 SUM=R1
45	P	P	1	13	
46	:	-	1	13	<i>Занести в таблицу меток пару P, 13</i>
47	MULT	MULT	1	13	
48	A1	A1 MULT	1	13	
49	A2	A2 A1 MULT	1	13	
50	SUM	SUM A2 A1 MULT	1	13	
51	+	R1 A1 MULT	2	14	13 R1=A2+SUM
52	*	R1 MULT	2	15	14 R1=A1*R1
53	:=	-	1	16	15 MULT=R1
54	КП	-	1	17	16 REM Конец процедуры
55	КП	-	1	18	17 REM Конец процедуры

После замены символьных меток на числовые получим следующий машинный код:

```

1 REM Начало процедуры MAIN
2 REM Вещественные переменные A1, A2
3 A1=378
4 A2=.73
5 REM Начало процедуры CALC
6 REM Вещественные переменные SUM, MULT
7 R1=A1+A2

```

```
8 R2=R1>3.2
9 IF NOT(R2) GOTO 11
10 GOTO 13
11 R1=A2+A1
12 SUM=R1
13 R1=A2+SUM
14 R1=A1*R1
15 MULT=R1
16 REM Конец процедуры
17 REM Конец процедуры
```

Полученный текст на машинном языке соответствует всем требованиям языка Бейсик.

3.4. Задание к лабораторной работе

Написать на языке высокого уровня программу преобразования внутреннего представления (ОПЗ) входной программы в текст на выходном языке (машинном или высокого уровня) в соответствии с вариантом задания.

Программа получает на входе файл, содержащий ОПЗ исходной программы, а также таблицы идентификаторов и констант. Выходом программы является текст программы на выходном языке.

Отчет по работе должен содержать:

1. Описание выходного языка;
2. Словесное описание всех особенностей работы МП-автомата для заданных языков;
3. Таблицу – описание семантических процедур МП-автомата;
4. Описание программной реализации (структуры данных и алгоритмы);
5. Листинг программы;
6. Пример входного файла (в исходном виде и в виде ОПЗ) и выходного файла (текста программы на выходном языке).