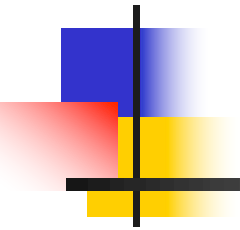


第八章 语法制导翻译及中间代码生成

8.1语法制导翻译概述

在语法分析过程中，随着分析的步步进展，根据每个产生式所对应的**语义子程序**（语义动作）进行翻译（产生中间代码）的办法。

标记说明

- 描述语义动作时，需要赋予每个文法符号X（终结符或非终结符）以种种不同方面的**值**，如X.TYPE(类型)，X.VAL(值)等。
- 一个产生式中同一符号多次出现，用**上角标**来区分

  $E \rightarrow E+E$ 表示为 $E \rightarrow E^{(1)}+ E^{(2)}$
- 每个产生式的语义动作，写在该产生式之后的

例 文法及其语义动作

产生式	语义规则
$L \rightarrow E$	$\{print(E.val)\}$
$E \rightarrow E^{(1)} + E^{(2)}$	$\{E.val = E^{(1)}.val + E^{(2)}.val\}$
$E \rightarrow E^{(1)} * E^{(2)}$	$\{E.val = E^{(1)}.val * E^{(2)}.val\}$
$E \rightarrow (E^{(1)})$	$\{E.val = E^{(1)}.val\}$
$E \rightarrow n$	$\{E.val = lexval\}$

上述的语义动作等于给出了计算由+,*组成的证书算术式的过程。其相应的程序段如下

产生式	语义规则
$L \rightarrow E$	*print* VAL[TOP]
$E \rightarrow E^{(1)} + E^{(2)}$	VAL[TOP] = VAL[TOP]+VAL[TOP+2]
$E \rightarrow E^{(1)} * E^{(2)}$	VAL[TOP] = VAL[TOP]*VAL[TOP+2]
$E \rightarrow (E^{(1)})$	VAL[TOP] = VAL[TOP+1]
$E \rightarrow \mathbf{n}$	VAL[TOP] = *lexval*

若把语义动作改为产生中间代码的动作，就能分析的进展逐步生成中间代码

中间代码的必要性

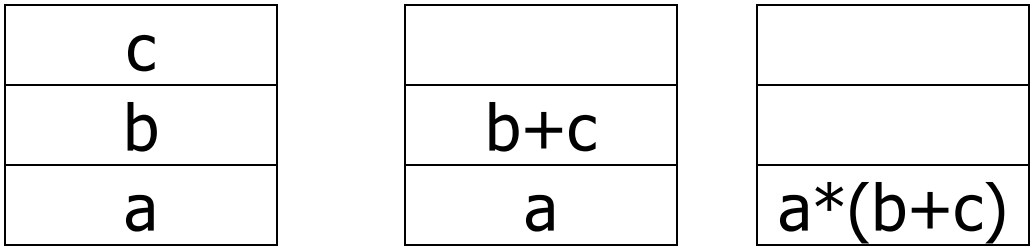
大部分的编译器都不直接产生目标代码，虽然这是可以实现的，但是产生的代码不是最优的。因为这涉及到**寄存器的分配问题**，在语义分析阶段，很难有效地分配它们。

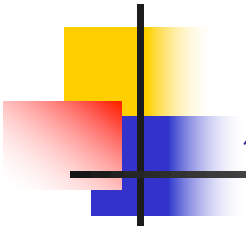
8.2逆波兰表示法：后缀式

- 1、把运算量写在前面，算符写在后面，不用括号的表示法。
- a*(b+c)=>abc+*
- (a+b)*(c+d)=>ab+cd+*

计值方法：

- 适合栈来计算：从左到右扫描后缀式，碰到运算量就推进栈；碰到K目运算就作用于栈顶的K项，并将运算结果代替这K项。





后缀式的推广

- 推广到比表达式更大的范围
- 例：if e then x else y
- 可表示成exy#, #代表三目运算符if-then-else.

8.3 三元式（op,arg1,arg2）

- A+B*C:
- OP ARG1 ARG2
- (1) * B C
- (2) + A (1)

直接三元式

间接三元式

- 用一张间接码表辅以三元式表，表示中间代码。
- 例：X:=(A+B)*C
- Y:=D^(A+B)
- 间接码表　　三元式表

间接码表		op	arg1	arg2
（1）				
（2）	(1)	+	A	B
（3）	(2)	*	(1)	C
(1)	(3)	:=	x	(2)
（4）	(4)	^	D	(1)
（5）	(5)	:=	Y	(4)

间接三元式

- 优点：
- 便于优化时调整运算顺序，只需重新安排码表，无需改动三元式表。
- 相同的三元式无需重复填进三元式表中，节省空间。

8.4四元式

- （OP，ARG1，ARG2，RESULT）
- A:=-B*（C+D）

	OP	ARG1	ARG2	RESULT
（1）	@取反	B	-	T1
（2）	+	C	D	T2
（3）	*	T1	T2	T3
（4）	:=	T3	-	A

- 优点：便于优化。

8.5 简单算术表达式与赋值语句到四元式的翻译

一、例文法：

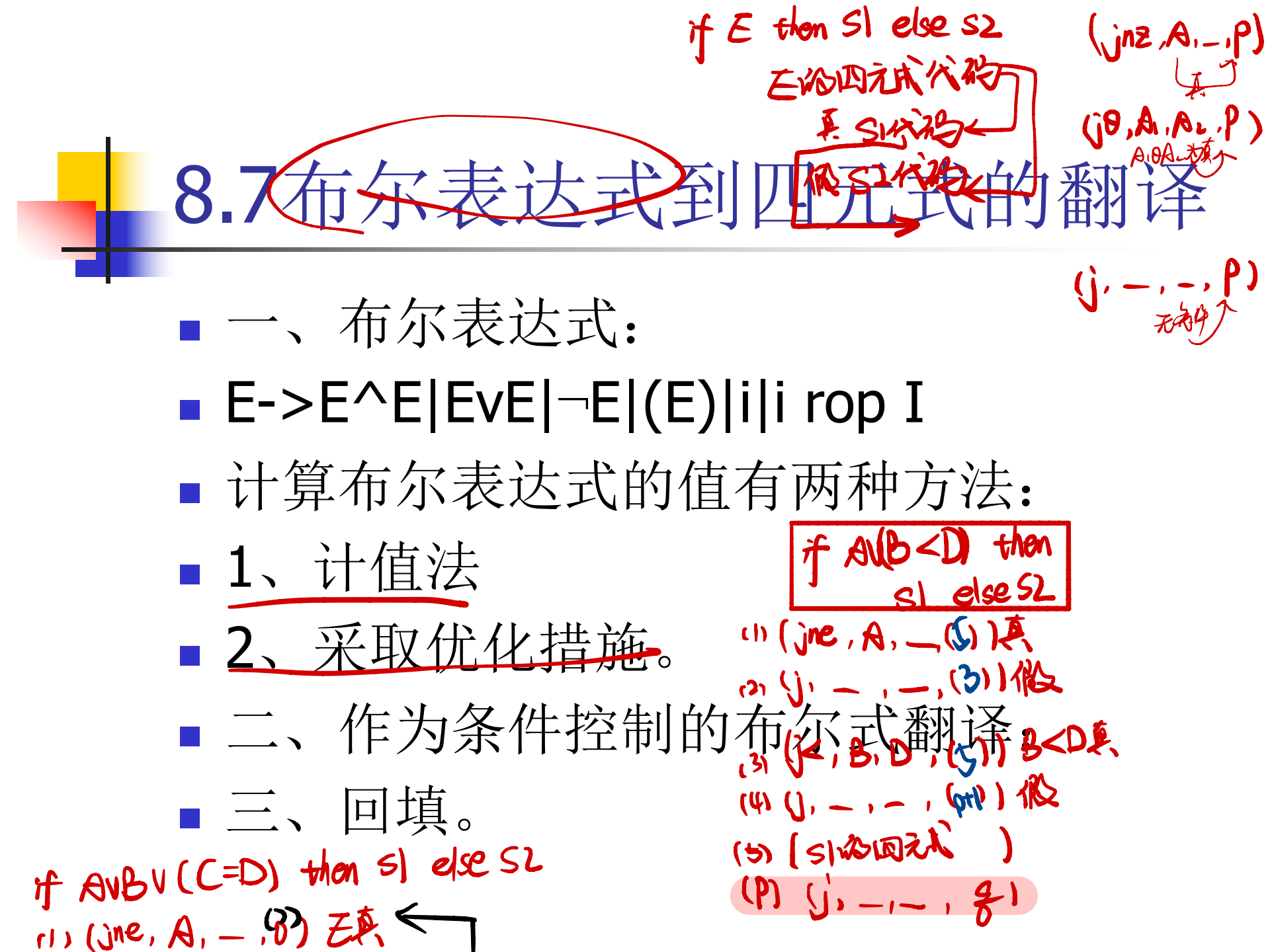
- A->i:=E
- E->E+E|E*E|-E|(E)|i
- NEWTEMP:函数过程。每次调用时，都回送一个代表新临时变量名的整数码作为函数值，T1，T2，。。。
- ENTRY（i）:函数过程，查符号表以确定i在表中的位置。
- E.PLACE:与非终结符E相联系的语义变量，表示存放E值的变量名在符号表的入口或整数码。
- GEN（OP，ARG1，ARG2，RESULT）：一个语义过程，把一个四元式填入四元式表。

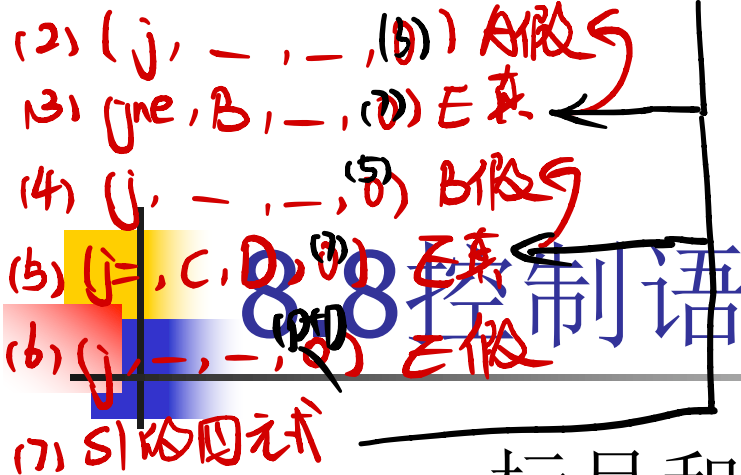
8.6 类型转换(暂时不介绍)

- E.MODE：表示非终结符E的类型信息。
- 例：X:=X+I*J
- (X^i,I,J,T1)
- (itr,T1,_,T2)
- ($+^r$,Y,T2,T3)
- (:=,T3,_,X)

8.7布尔表达式到四元式的翻译

- 一、布尔表达式：
- E->E^E|EvE|¬E|(E)|i|i rop I
- 计算布尔表达式的值有两种方法：
- 1、计值法
- 2、采取优化措施。
- 二、作为条件控制的布尔式翻译：
- 三、回填。





8.8控制语句的翻译

- 一、标号和转移语句（goto L）

语句形式：L：S;

- 1、若goto L是向后转移的语句，则L已经定义了，可立即产生出四元式（j,_,_,_p）
- 2、若goto L是向前转移语句，则L未定义。产生一个不完全的四元式（j,_,_,_），它的转移目标待L定义时再回填。

3533 15:54 5.12 Wed

8.9条件语句

- 文法：
- S->if E then S
- | if E then S else S
- | while E do S
- | begin L end
- | A
- L->L,S
- |S

8.10循环语句

- S->for i:=E^1 step E^2 until E^3 do S^1
- 四、分叉语句：
- 语法：case E of
- c1:S_1;
- c2:S_2;
- ……
- cn-1:S_{n-1};
- otherwise:S_n
- end