



编译原理

第七章 语义分析和中间代码产生

第七章 语义分析和中间代码产生

- 中间语言
- 赋值语句的翻译
- 布尔表达式的翻译
- 控制语句的翻译
- 过程调用的处理

第七章 语义分析和中间代码产生

- 中间语言
- 赋值语句的翻译
- 布尔表达式的翻译
- 控制语句的翻译
- 过程调用的处理

7.4 布尔表达式的翻译

- 布尔表达式的两个基本作用

- 用于逻辑演算，计算逻辑值
- 用于控制语句的条件式

- 产生布尔表达式的文法

$$E \rightarrow E \text{ or } E \mid E \text{ and } E \mid \text{not } E \mid (E) \mid i \text{ rop } i \mid i$$

计算布尔表达式的两种方法

- 如同计算算术表达式一样，一步步算

$1 \text{ or } (\text{not } 0 \text{ and } 0) \text{ or } 0$
 $= 1 \text{ or } (1 \text{ and } 0) \text{ or } 0$
 $= 1 \text{ or } 0 \text{ or } 0$
 $= 1 \text{ or } 0$
 $= 1$

- 采用某种优化措施

- ☐ 把 $A \text{ or } B$ 解释成
B

if A then true else

- ☐ 把 $A \text{ and } B$ 解释成

if A then B else false

- ☐ 把 $\text{not } A$ 解释成

if A then false else true

两种不同的翻译方法

- 第一种翻译法

A or B and C=D 翻译成

(1) (=, C, D, T_1)

(2) (and, B, T_1 , T_2)

(3) (or, A, T_2 , T_3)

- 第二种翻译法适合于作为条件表达式的布尔表达式使用

7.4.1 数值表示法

- $a \text{ or } b \text{ and not } c$ 翻译成

$T_1 := \text{not } c$

$T_2 := b \text{ and } T_1$

$T_3 := a \text{ or } T_2$

- $a < b$ 的关系表达式可等价地写成

$\text{if } a < b \text{ then } 1 \text{ else } 0$, 翻译成

100: if $a < b$ goto 103

101: $T := 0$

102: goto 104

103: $T := 1$

104:

关于布尔表达式的数值表示法的翻译模式

- 过程 `emit` 将三地址代码送到输出文件中
- `nextstat` 给出输出序列中下一条三地址语句的地址索引
- 每产生一条三地址语句后，过程 `emit` 便把 `nextstat` 加 1

关于布尔表达式的数值表示法的翻译模式

$E \rightarrow E_1 \text{ or } E_2$ $\{E.\text{place} := \text{newtemp};$
 $\text{emit}(E.\text{place} := E_1.\text{place} \text{ 'or' } E_2.\text{place})\}$

$E \rightarrow E_1 \text{ and } E_2$ $\{E.\text{place} := \text{newtemp};$
 $\text{emit}(E.\text{place} := E_1.\text{place} \text{ 'and' } E_2.\text{place})\}$

$E \rightarrow \text{not } E_1$ $\{E.\text{place} := \text{newtemp};$
 $\text{emit}(E.\text{place} := \text{'not' } E_1.\text{place})\}$

$E \rightarrow (E_1)$ $\{E.\text{place} := E_1.\text{place}\}$

关于布尔表达式的数值表示法的翻译模式

a<b 翻译成
100: if a<b goto 103
101: T:=0
102: goto 104
103: T:=1
104:

$E \rightarrow id_1 \text{ relop } id_2$ { $E.place := newtemp$;
 $emit('if' \ id_1.place \ relop. \ op \ id_2. \ place$
 $'goto' \ nextstat+3);$
 $emit(E.place \ ':=' \ '0');$
 $emit('goto' \ nextstat+2);$
 $emit(E.place \ ':=' \ '1')$ }

$E \rightarrow id$ { $E.place := id.place$ }

布尔表达式 $a < b$ or $c < d$ and $e < f$ 的翻译结果

100: if $a < b$ goto 103

101: $T_1 := 0$

102: goto 104

103: $T_1 := 1$

104: if $c < d$ goto 107

105: $T_2 := 0$

106: goto 108

107: $T_2 := 1$

108: if $e < f$ goto 111

109: $T_3 := 0$

110: goto 112

111: $T_3 := 1$

112: $T_4 := T_2$ and T_3

113: $T_5 := T_1$ or T_4

$E \rightarrow id_1 \text{ relop } id_2$

```
{ E.place:=newtemp;
  emit('if' id1.place relop. op id2. place
      'goto' nextstat+3);
  emit(E.place ':=' '0');
  emit('goto' nextstat+2);
  emit(E.place ':=' '1') }
```

$E \rightarrow id$

```
{ E.place:=id.place }
```

$E \rightarrow E_1 \text{ or } E_2$

```
{ E.place:=newtemp;
  emit(E.place ':=' E1.place 'or' E2.place)}
```

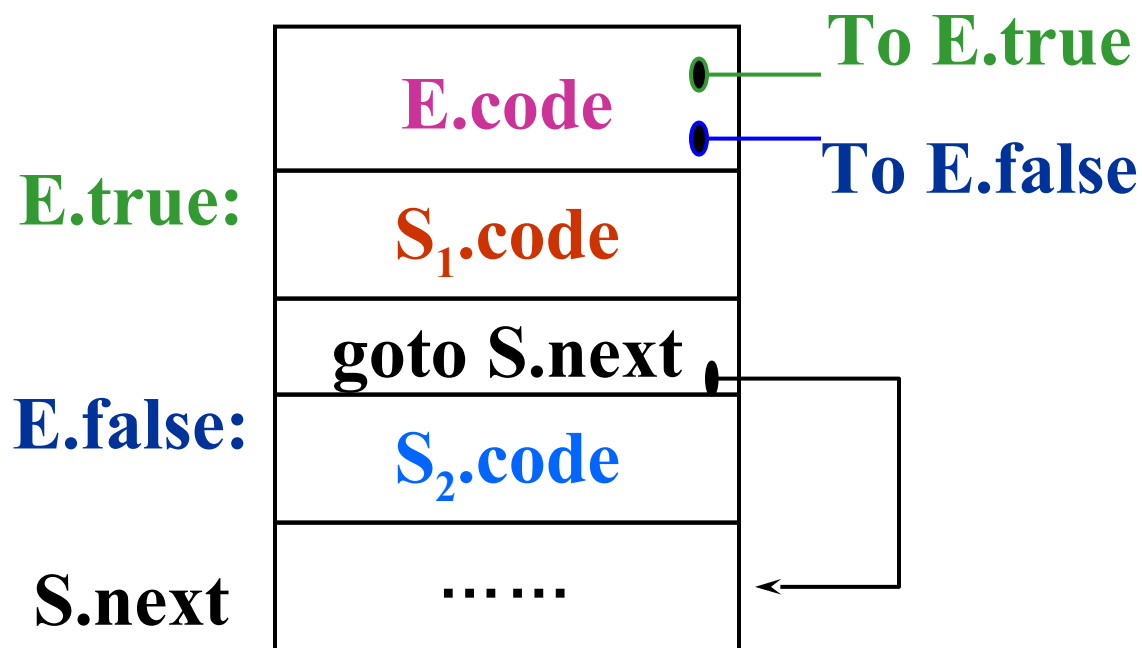
$E \rightarrow E_1 \text{ and } E_2$

```
{ E.place:=newtemp;
  emit(E.place ':=' E1.place 'and' E2.place) }
```

7.4.2 作为条件控制的布尔式翻译

- 条件语句 if E then S_1 else S_2

赋予 E 两种出口：一真一假



- 例：把语句：if $a > c$ or $b < d$ then S_1 else S_2
翻译成如下的一串三地址代码

	if $a > c$ goto L2	“真”出口
	goto L1	
L1:	if $b < d$ goto L2	“真”出口
	goto L3	“假”出口
L2:	(关于 S_1 的三地址代码序列)	
	goto Lnext	
L3:	(关于 S_2 的三地址代码序列)	

Lnext:

产生布尔表达式三地址代码的语义规则

- 每次调用函数 `newlabel` 后都返回一个新的符号标号
- 对于一个布尔表达式 E ，引用两个标号
 - $E.true$ 是 E 为‘真’时控制流转向的标号
 - $E.false$ 是 E 为‘假’时控制流转向的标号

产生布尔表达式三地址代码的语义规则

产生式

$E \rightarrow E_1 \text{ or } E_2$

语义规则

$E_1.\text{true} := E.\text{true};$

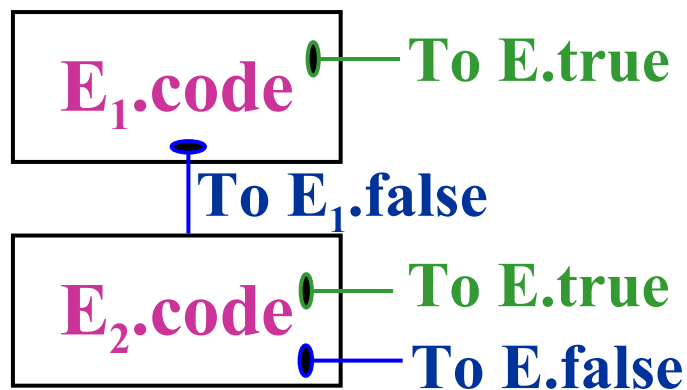
$E_1.\text{false} := \text{newlabel};$

$E_2.\text{true} := E.\text{true};$

$E_2.\text{false} := E.\text{false};$

$E.\text{code} := E_1.\text{code} \parallel$

$\text{gen}(E_1.\text{false} ':') \parallel E_2.\text{code}$



产生布尔表达式三地址代码的语义规则

产生式

$E \rightarrow E_1 \text{ and } E_2$

语义规则

$E_1.\text{true} := \text{newlabel};$

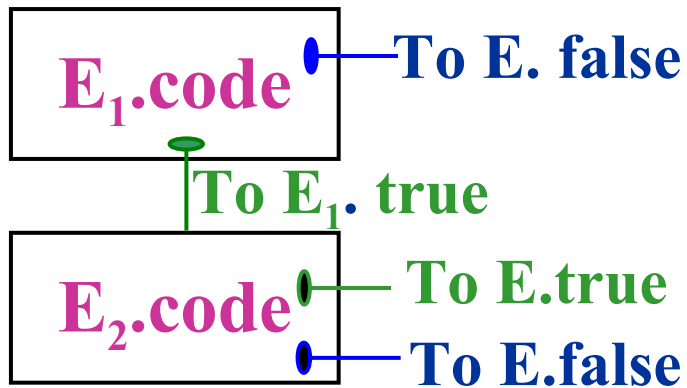
$E_1.\text{false} := E.\text{false};$

$E_2.\text{true} := E.\text{true};$

$E_2.\text{false} := E.\text{false};$

$E.\text{code} := E_1.\text{code} \parallel$

$\text{gen}(E_1.\text{true} ':') \parallel E_2.\text{code}$



产生布尔表达式三地址代码的语义规则

产生式

$E \rightarrow \text{not } E_1$

$E \rightarrow (E_1)$

语义规则

$E_1.\text{true} := E.\text{false};$
 $E_1.\text{false} := E.\text{true};$
 $E.\text{code} := E_1.\text{code}$

$E_1.\text{true} := E.\text{true};$
 $E_1.\text{false} := E.\text{false};$
 $E.\text{code} := E_1.\text{code}$

产生布尔表达式三地址代码的语义规则

产生式

语义规则

$E \rightarrow id_1 \text{ relop } id_2$	$E.code := \text{gen}(\text{'if ' } id_1.place \text{ relop.op } id_2.place \text{ 'goto' } E.true) \parallel \text{gen}(\text{'goto' } E.false)$
--	---

$E \rightarrow \text{true}$	$E.code := \text{gen}(\text{'goto' } E.true)$
-----------------------------	---

$E \rightarrow \text{false}$	$E.code := \text{gen}(\text{'goto' } E.false)$
------------------------------	--

考虑如下表达式:

$a < b$ or $c < d$ and $e < f$

产生式

$E \rightarrow id_1 \text{ relop } id_2$

语义规则

$E.code := \text{gen}(\text{'if ' } id_1.place$
 $\text{relop.op } id_2.place \text{'goto' } E.true) \parallel$
 $\text{gen('goto' } E.false)$

$E \rightarrow E_1 \text{ or } E_2$

$E_1.true := E.true;$
 $E_1.false := \text{newlabel};$
 $E_2.true := E.true;$
 $E_2.false := E.false;$
 $E.code := E_1.code \parallel \text{gen}(E_1.false \text{' : '}) \parallel E_2.code$

$E \rightarrow E_1 \text{ and } E_2$

$E_1.true := \text{newlabel};$
 $E_1.false := E.false;$
 $E_2.true := E.true;$
 $E_2.false := E.false;$

语义规则

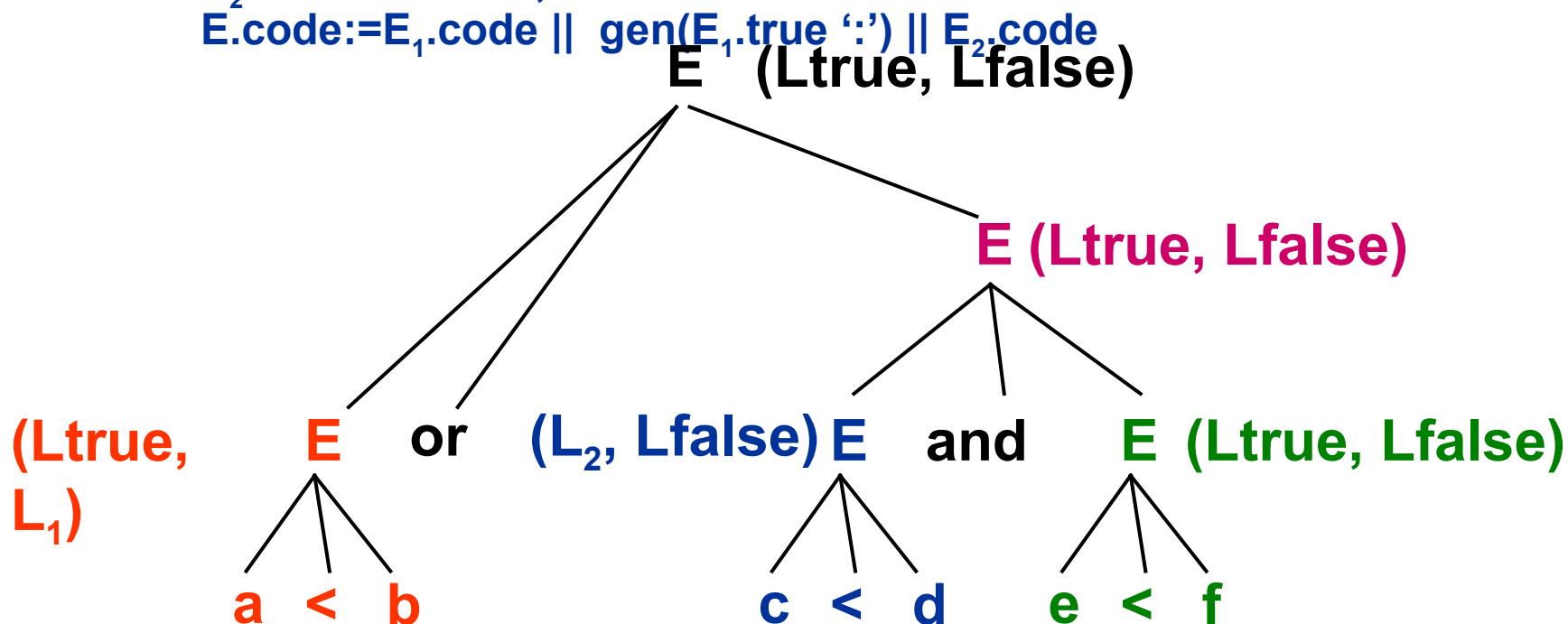
产生式

$E \rightarrow E_1 \text{ or } E_2$

$E_1.\text{true} := E.\text{true};$
 $E_1.\text{false} := \text{newlabel};$
 $E_2.\text{true} := E.\text{true};$
 $E_2.\text{false} := E.\text{false};$
 $E.\text{code} := E_1.\text{code} \parallel \text{gen}(E_1.\text{false} ':') \parallel E_2.\text{code}$

$E \rightarrow E_1 \text{ and } E_2$

$E_1.\text{true} := \text{newlabel};$
 $E_1.\text{false} := E.\text{false};$
 $E_2.\text{true} := E.\text{true};$
 $E_2.\text{false} := E.\text{false};$
 $E.\text{code} := E_1.\text{code} \parallel \text{gen}(E_1.\text{true} ':') \parallel E_2.\text{code}$



产生式

语义规则

$E \rightarrow id_1 \text{ relop } id_2$

$E.code := \text{gen}(\text{'if ' } id_1.place \text{ relop.op } id_2.place \text{ 'goto' } E.true) \parallel \text{gen}(\text{'goto' } E.false)$

$E \rightarrow E_1 \text{ or } E_2$

$E_1.true := E.true;$
 $E_1.false := \text{newlabel};$
 $E_2.true := E.true;$
 $E_2.false := E.false;$
 $E.code := E_1.code \parallel \text{gen}(E_1.false ':') \parallel E_2.code$

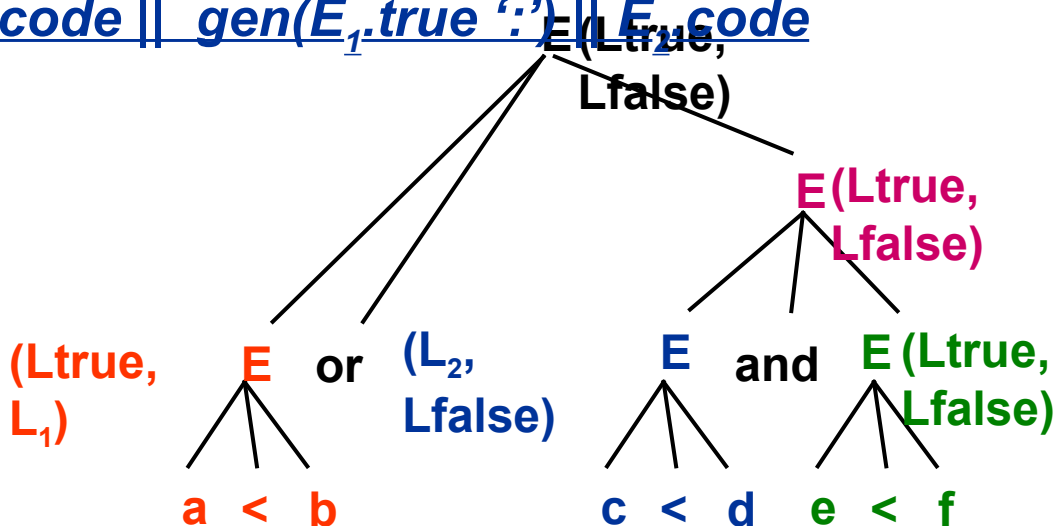
$E \rightarrow E_1 \text{ and } E_2$

$E_1.true := \text{newlabel};$
 $E_1.false := E.false;$
 $E_2.true := E.true;$
 $E_2.false := E.false;$
 $E.code := E_1.code \parallel \text{gen}(E_1.true ':') \parallel E_2.code$

if a < b goto Ltrue
 goto L₁

L₁: if c < d goto L₂
 goto Lfalse

L₂: if e < f goto Ltrue
 goto Lfalse



小结

- 布尔表达式的翻译
 - 数值表示法
 - 作为条件控制的布尔式翻译