

第八章 语法制导翻译 和 中间代码生成

8.5 布尔表达式的翻译

广东工业大学计算机学院

本课内容

■ 8.5 布尔表达式的翻译

- 程序设计语言中的布尔表达式有两个作用：

- (1) 计算逻辑值。

- (2) 用于改变控制流语句中的条件表达式。

- 例如在**if E then**、**if E then-else**，或是**while E do**语句中，都会用到布尔表达式**E**。

布尔表达式文法

- 为简单起见，我们只考虑如下文法生成的布尔表达式。

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

- **$| E \text{ or } E$**

- **$| \text{ not } E$**

- **$| \text{id rop id}$**

- **$| \text{id}$**

- **$| \text{true}$**

- **$| \text{false}$**

- **rop**是关系符，如 \leq ， $<$ ， $=$ ， \geq ， \neq 等等。

- 按通常习惯，约定布尔算符的**优先顺序**为**not > and > or**。

- 并且**and**和**or**服从**左结合**。

布尔表达式值的计值1

- 布尔表达式值的计算有两种方法：
- **1.** 计算出各部分的真假值，最后计算出整个表达式的值。
- 例如用数值**1**表示**true**，用**0**表示**false**。布尔表达式**1 or (not 0 and 0) or 0**的计算过程是：
 - **1 or (not 0 and 0) or 0**
 - **= 1 or (1 and 0) or 0**
 - **= 1 or 0 or 0**
 - **= 1 or 0**
 - **= 1**

布尔表达式值的计值2

- 2. 采取某种优化措施，只计算部分表达式。
- 例如要计算 **A or B**，若计算出 **A** 的值为 **1**，那么 **B** 的值就无需再计算。
- 对于不包含布尔函数调用的表达式，上述两种方法没有什么差别。
- 但如果一个布尔式中含有布尔函数调用，并且这种函数调用引起副作用（如有对全局量的赋值）时，这两种方法未必等价。
- 例如存在表达式：**A or test(B) /*B是全局变量*/**
- 采用哪种方法取决于程序设计语言的语义。

布尔表达式的翻译举例

- 如果按第一种办法计算布尔表达式，则
a or b and not c翻译成的四元式序列为：
- (1) **t1 := not c**
- (2) **t2 := b and t1**
- (3) **t3 := a or t2**

布尔表达式的翻译举例

- 对于像 $a < b$ 这样的关系表达式，可看成等价的条件语句 **if $a < b$ then 1 else 0**
- 翻译成的四元式序列为：
 - (1) **if $a < b$ goto (4)**
 - (2) **$t := 0$**
 - (3) **goto (5)**
 - (4) **$t := 1$**
 - (5) ...
- 其中临时变量 t 存放布尔表达式 $a < b$ 的值，(5)为后续的四元式序号。

布尔表达式的翻译

■ 下面是常见布尔表达式的翻译：

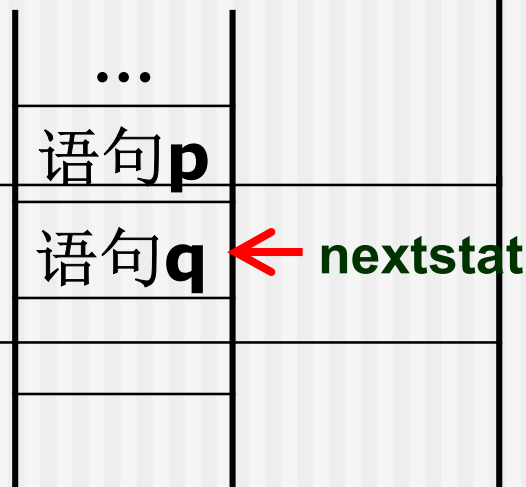
| 赋值语句 | 翻译(四元式语义描述) |
|--|--|
| (1) $E \rightarrow E1 \text{ or } E2$ | { E.place := newtemp; emit(E.place ':=' E1.place 'or' E2.place) } |
| (2) $E \rightarrow E1 \text{ and } E2$ | { E.place := newtemp; emit(E.place ':=' E1.place 'and' E2.place) } |
| (3) $E \rightarrow \text{not } E1$ | { E.place := newtemp; emit(E.place ':=' 'not' E1.place) } |
| (4) $E \rightarrow (E1)$ | { E.place := E1.place } |

布尔表达式的翻译(续)

与栈中的**top**指针类似

- **nextstat** : 该过程给出在输出序列中下一四元式的序号, **emit**过程每被调用一次, **nextstat**增加**1**。

| 赋值语句 | 翻译(四元式语义描述) |
|--|---|
| (5) $E \rightarrow id1 \text{ rop } id2$ | <pre> { E.place := newtemp; emit('if' id1.place 'rop' id2.place 'goto' nextstat + 3); emit(E.place ':=' '0'); emit('goto' nextstat + 2); emit(E.place ':=' '1') }</pre> |
| (6) $E \rightarrow true$ | <pre> { E.place := newtemp; emit(E.place ':=' '1') }</pre> |
| (7) $E \rightarrow false$ | <pre> { E.place := newtemp; emit(E.place ':=' '0') }</pre> |



控制语句中布尔表达式的翻译

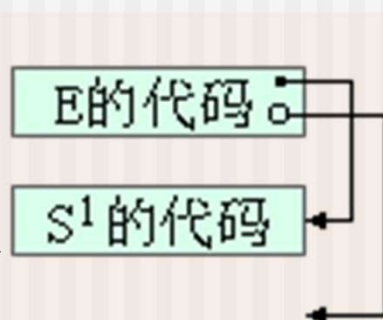
- 现在讨论在**if – then**、**if – then - else**和**while - do**等语句中的**布尔表达式E**的翻译。这些语句的语法为：

- **S** → **if E then S1**
| **if E then S1 else S2**
| **while E do S1**

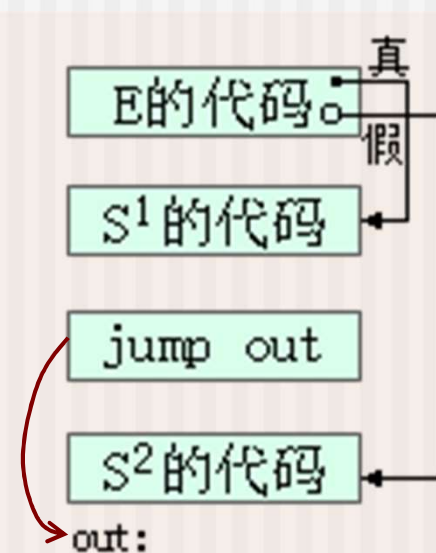
- 分别使用“•”和“○”表示**E**为真和假时控制流向的转移。

- “•”叫真出口；
“○”叫假出口

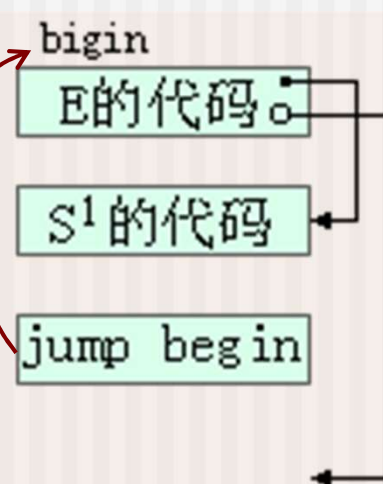
广东工业大学计算机学院



(a) if E then S¹
代码结构



(b) if E then S¹else S²
代码结构



(c) while E do S¹
代码结构

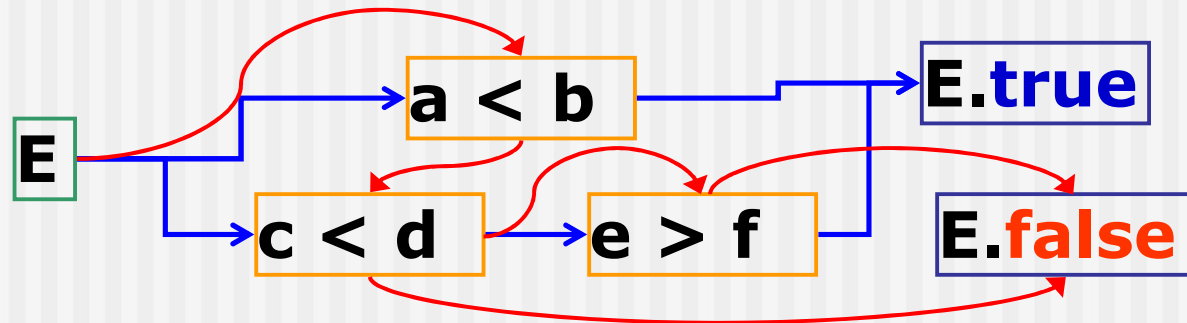
翻译布尔表达式的基本思路

- 例如将布尔表达式 **$E = a \text{ rop } b$** 翻译成四元式代码:
- **if a rop b goto E.true**
- 和
goto E.false
- **E.true** 和 **E.false** 分别表示 **E** 的“真”和“假”出口转移目标地址，但此时未能立即确定具体的值。
- 例如: **$S \rightarrow \text{if } E \text{ then } S1 \text{ else } S2$**
- 要翻译到 **S1** 的第一条四元式，才能确定 **E.true**
- 要翻译到 **S2** 的第一条四元式，才能确定 **E.false**

布尔表达式翻译举例

- 例如布尔表达式 $a < b \text{ or } c < d \text{ and } e > f$ 翻译为如下四元式序列:

- (1) **if** $a < b$ **goto** E.true
- (2) **goto** (3)
- (3) **if** $c < d$ **goto** (5)
- (4) **goto** E.False
- (5) **if** $e > f$ **goto** E. true
- (6) **goto** E.False



控制语句中布尔表达式的翻译举例

■ **if** $a < b$ **or** $c < d$ **and** $e > f$ **then** **S1** **else** **S2** 四元式序列:

■ (1) **if** $a < b$ **goto** (7)

(2) **goto** (3)

■ (3) **if** $c < d$ **goto** (5)

(4) **goto** (p+1)

■ (5) **if** $e > f$ **goto** (7)

(6) **goto** (p+1)

■ (7) (关于**S1**的四元式)

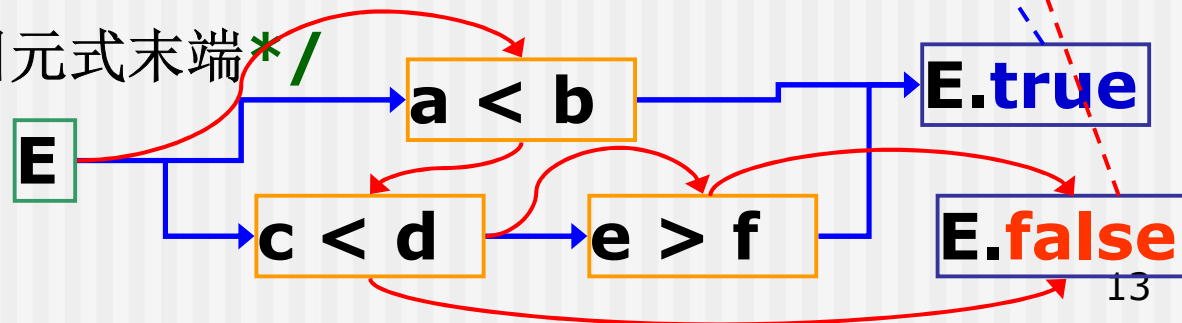
...
(p) **goto** (q)

/* 跳过关于**S2**的四元式 */

■ (p+1) /* 关于**S2**的四元式 */

...
(q-1) ... /* **S2**的四元式末端 */

■ (q) ...



四元式中地址的回填

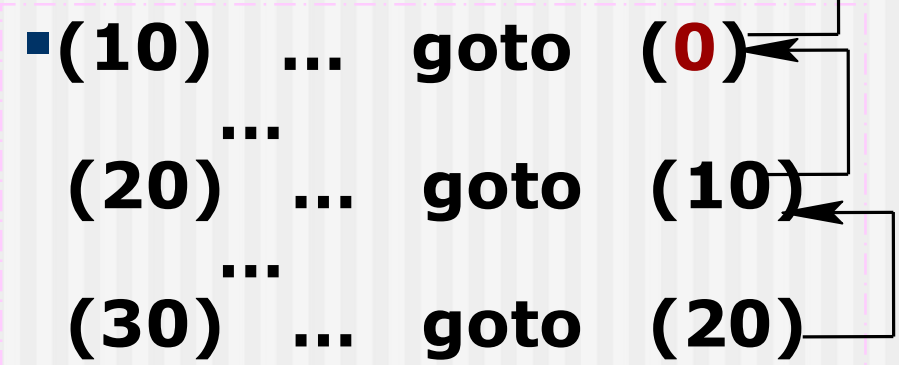
- 所以，上述四元式（**1**）和（**5**），（**4**）和（**6**）的转移地址**并不能**在产生这些四元式的同时得知。
- 它们是在整个布尔表达式的四元式序列生成之后，才**回填**的地址。
- 为了记录需回填地址的四元式，常采用一种“**拉链**”的办法：
- 把需回填**E. true**的四元式拉成一条链，把需回填**E. false**的四元式拉成一条链，分别称做“真”链和“假”链。

四元式中地址的回填举例

- 例有四元式序列：

- (10) ... goto E.true
...
(20) ... goto E.true
...
(30) ... goto E.true

- (10) ... goto (0)
...
(20) ... goto (10)
...
(30) ... goto (20)

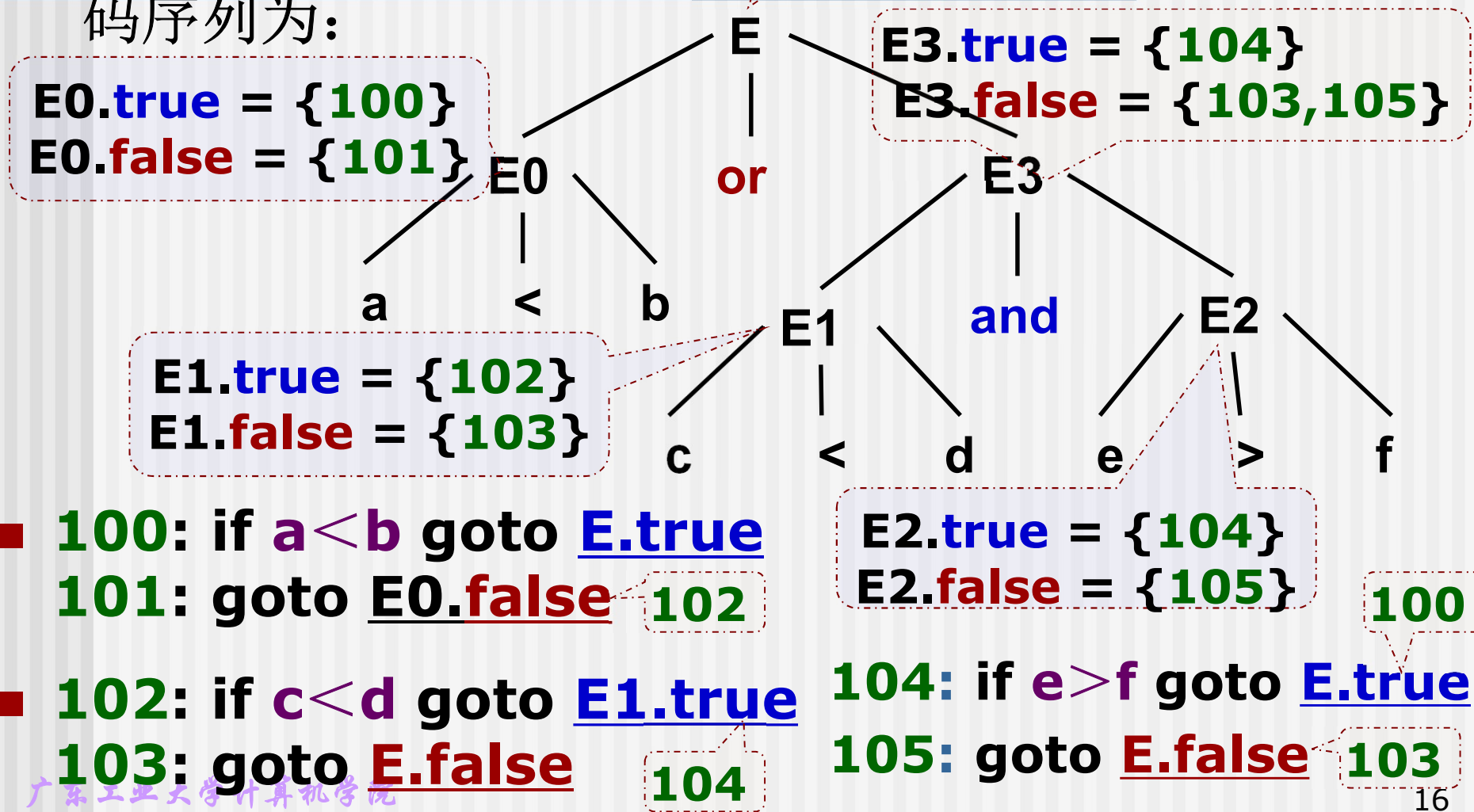


- 地址 (30) 称作链首，0为链尾标志，即地址(10)为链尾。

“拉链” 举例

E.true = {100, 104}
E.false = {103, 105}

- 对于表达式 **a < b or c < d and e > f**, 翻译得的代码序列为:



“拉链” 举例讨论

假链的链首

真链的链首

- 对于 $E = a < b \text{ or } c < d \text{ and } e > f$, 翻译得

- 100: if $a < b$ goto $E.true$

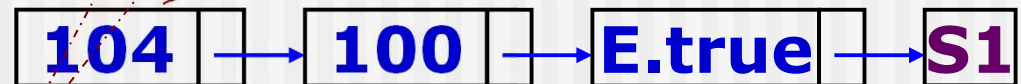
101: goto 102

- 102: if $c < d$ goto 104

103: goto $E.false$

- 104: if $e > f$ goto $E.true$

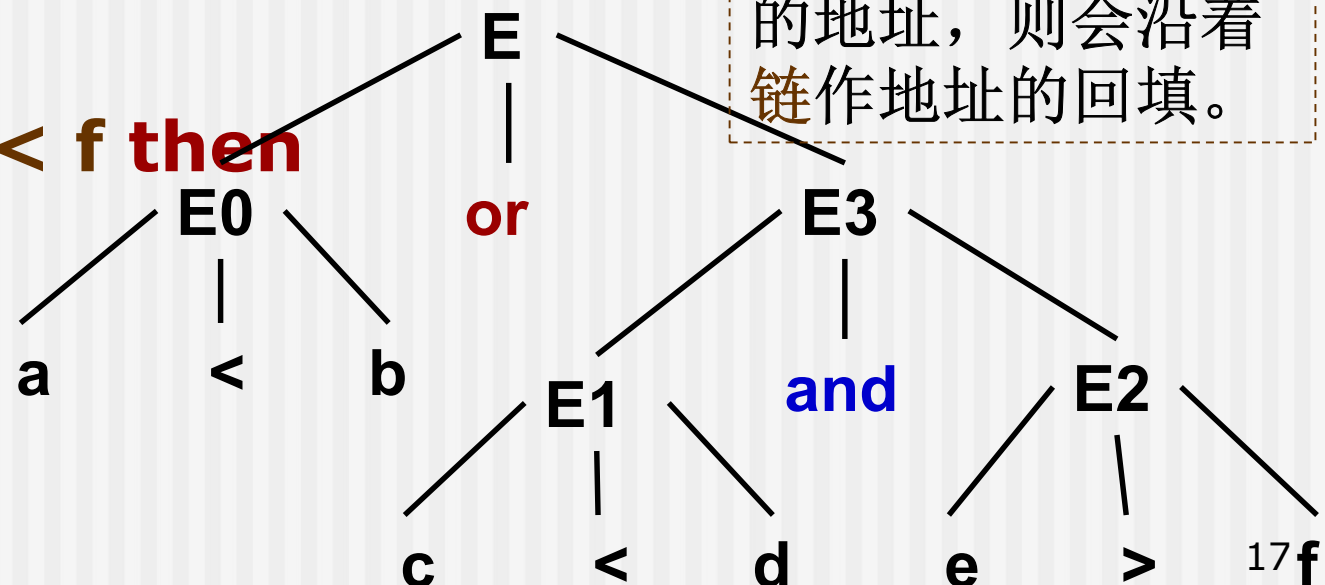
105: goto $E.false$



- if $E \text{ and } e < f \text{ then}$

■ S1

■ else S2



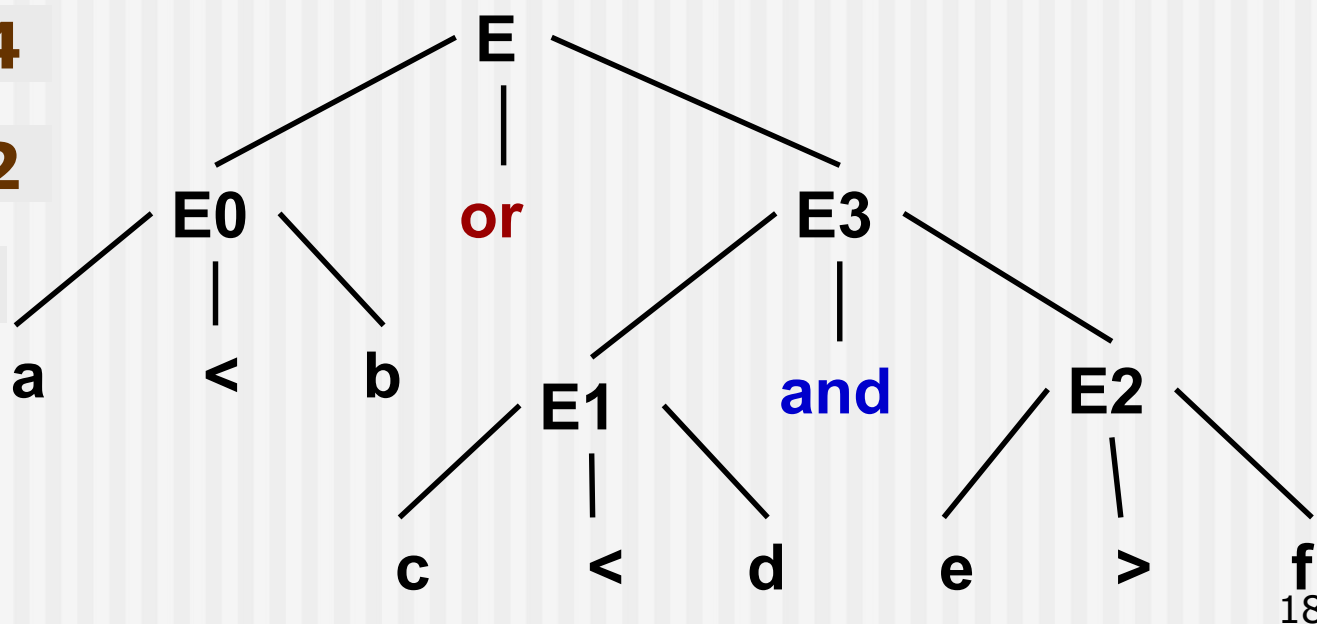
一旦确定 S1 和 S2 的地址, 则会沿着链作地址的回填。

布尔表达式的一

- 一些常用的属性:
- 语义值 **codebegin**: 与非终结符 **E** 相连, 表示表达式 **E** 的第 **1** 个四元式语句序号。
- **E0.cb = 100**
- **E2.cb = 104**
- **E3.cb = 102**
- **E.cb = 100**

■ 对于 **E = a < b or c < d and e > f**, 翻译得:

```
100: if a < b goto E.true
101: goto 102
102: if c < d goto 104
103: goto E.false
104: if e > f goto 100
105: goto 103
```



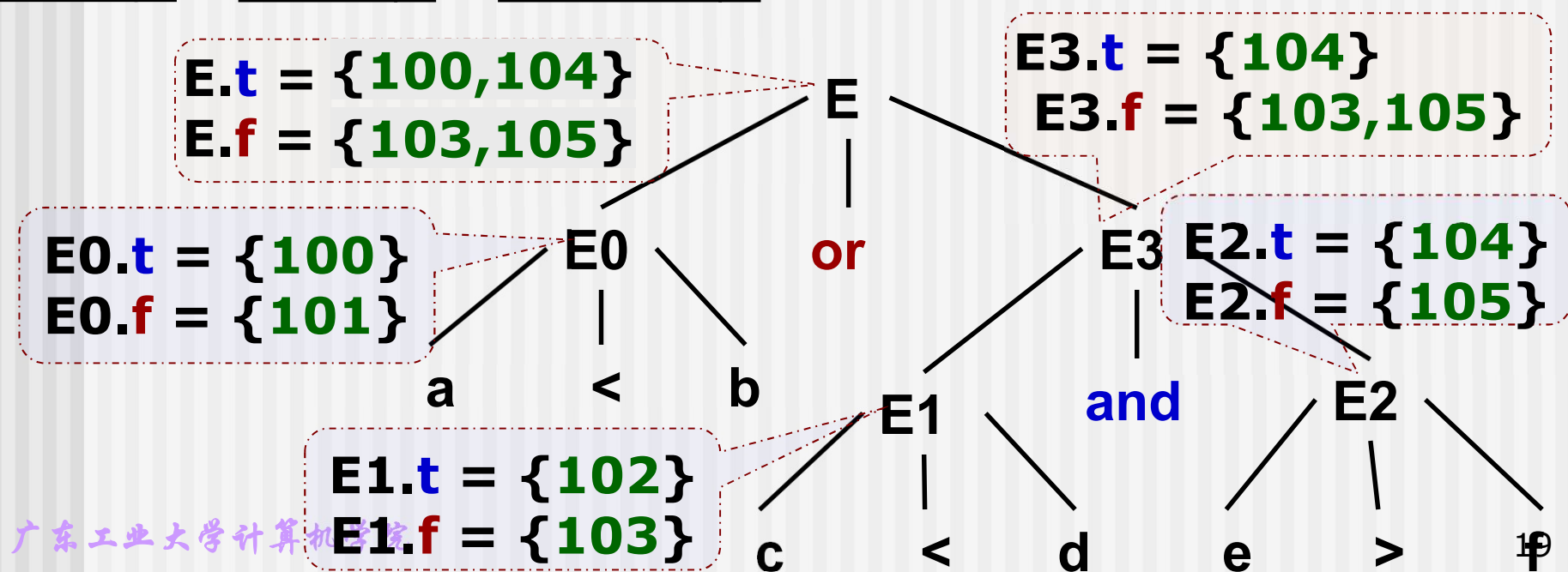
1)

- 一些常用的过程:
- **merge(p1, p2)**: 把**p1**和**p2**为链首的两条链合并为**1**条。得到的新链首是**p2**。



■ 对于 $E = a < b \text{ or } c < d \text{ and } e > f$, 翻译得:

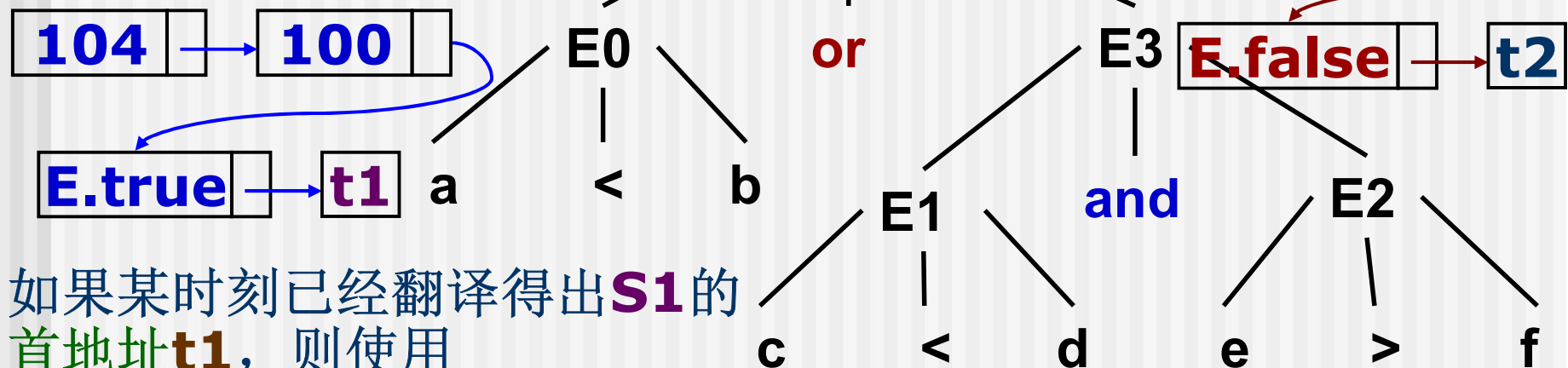
■ **100**: if $a < b$ goto **E.true**
■ **101**: goto **102**
■ **102**: if $c < d$ goto **104**
■ **103**: goto **E.false**
■ **104**: if $e > f$ goto **100**
■ **105**: goto **103**



2)

- 一些常用的过程:
- **3. backpatch(p, t):** 把p所链接的每个四元式的第4元都填上t。

- 例如:
- **if E and e < f then S1**
- **else S2**



- 如果某时刻已经翻译得出S1的首地址t1, 则使用backpatch过程回填“真”

■ 对于 **E = a < b or c < d and e > f**, 翻译得:

- **100: if a < b goto t1**
- **101: goto 102**
- **102: if c < d goto 104**
- **103: goto t2**
- **104: if e > f goto t1**
- **105: goto t2**

常见布尔表达式

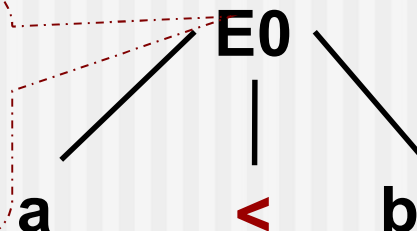
nextstat : 给出在输出序列中下一四元式的序号。**emit**过程每被调用一次, **nextstat**增加**1**。
假设**nextstat**的初值为 **100**

| 布尔表达式 | 翻译(四元式语义描述) |
|--|--|
| (5) $E \rightarrow id1 \text{ rop } id2$ | <pre> { E.true = nextstat; E.codebegin := nextstat; E.false = nextstat + 1; emit('if' id1.place 'rop' id2 'goto' --); emit('goto' --); }</pre> |

- 例如: 规约 $a < b$ 到 **E0** 时, 先计算属性, 然后产生**2**个四元式:
- **100**: if $a < b$ goto _____
- **101**: goto _____

此时 **nextstat** = **102**

E0.t = **100**
E0.cb = **100**
E0.f = **101**



常见布尔表达式的翻译(续1)

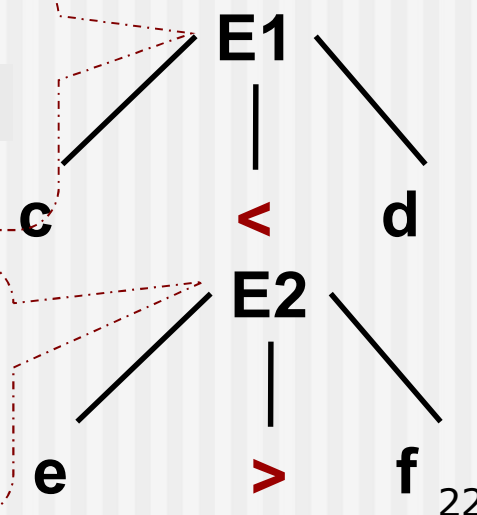
| 布尔表达式 | 翻译(四元式语义描述) |
|---------------------------------------|--|
| (5) $E \rightarrow id1$ $rop\ id2$ | <pre> { E.true = nextstat; nextstat = 104 E.codebegin := nextstat; E.false = nextstat + 1; emit('if' id1.place 'rop' id2 'goto' --); emit('goto' --); }</pre> |

■ 对于 $E = a < b \text{ or } c < d \text{ and } e > f$, 继续翻译:

- 100: if $a < b$ goto _____
- 101: goto _____
- 102: if $c < d$ goto _____
- 103: goto _____
- 104: if $e > f$ goto _____
- 105: goto _____

$E1.t = 102$
 $E1.cb = 102$
 $E1.f = 103$

$E2.t = 104$
 $E2.cb = 104$
 $E2.f = 105$



常见布尔表达式的翻译(续2)

| 布尔表达式 | 翻译(四元式语义描述) |
|---|--|
| (1) $E3 \rightarrow E1$ $\text{and } E2$ | <pre> { backpatch(E1.true, E2.codebegin); E3.codebegin := E1.codebegin; E3.true := E2.true; E3.false := merge(E1.false, E2.false); }</pre> |

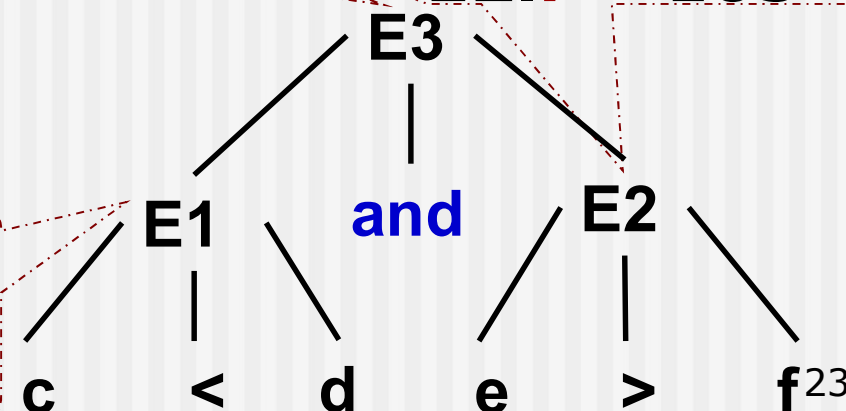
- 对于 $E = a < b \text{ or } c < d \text{ and } e > f$ 的翻译, 现在规约 $E3 \rightarrow E1 \text{ and } E2$

- 100: if $a < b$ goto _____
- 101: goto _____
- 102: if $c < d$ goto 104
- 103: goto $E3.false$
- 104: if $e > f$ goto $E3.true$
- 105: goto _____

$E3.t = 104$
 $E3.cb = 102$
 $E3.f = 105$

$E2.t = 104$
 $E2.cb = 104$
 $E2.f = 105$

$E1.t = 102$
 $E1.cb = 102$
 $E1.f = 103$



常见布尔表达式的翻译(续3)

| 布尔表达式 | 翻译(四元式语义描述) |
|---------------------------------------|---|
| (1) $E \rightarrow E0 \text{ or } E3$ | <pre> { backpatch(E0.false, E3.codebegin); E.codebegin := E0.codebegin; E.true := merge(E0.true, E3.true); E.false := E3.false; }</pre> |

- 对于 $E = a < b \text{ or } c < d \text{ and } e > f$ 的翻译, 现在规约 $E \rightarrow E0 \text{ or } E3$

- 100: if $a < b$ goto $E.true$

101: goto 102

- 102: if $c < d$ goto 104

103: goto $E.false$

- 104: if $e > f$ goto $E3.true$

105: goto 103

$E.t = 104$

$E.cb = 100$

$E.f = 105$

$E3.t = 104$

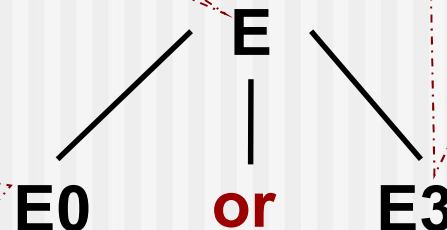
$E3.cb = 102$

$E3.f = 105$

$E0.t = 100$

$E0.cb = 100$

$E0.f = 101$



常见布尔表达式的翻译(续4)

| 布尔表达式 | 翻译(四元式语义描述) |
|------------------------------------|---|
| (3) $E \rightarrow \text{not } E1$ | $\{ E.\text{true} = E1.\text{false};$ $E.\text{codebegin} := E1.\text{codebegin};$ $E.\text{false} := E2.\text{true}; \}$ |
| (4) $E \rightarrow (E1)$ | $\{ E.\text{true} = E1.\text{true};$ $E.\text{codebegin} := E1.\text{codebegin};$ $E.\text{false} := E2.\text{flase}; \}$ |

$E.t = 101$
 $E.cb = 100$
 $E.f = 100$

not

E

E1

$E1.t = 100$
 $E1.cb = 100$
 $E1.f = 101$

常见布尔表达式的翻译(续5)

| 布尔表达式 | 翻译(四元式语义描述) |
|----------------------------------|---|
| (6) $E \rightarrow \text{true}$ | <pre>{ E.true = nextstat; E.codebegin := nextstat; emit('goto' --); }</pre> |
| (7) $E \rightarrow \text{false}$ | <pre>{ E.false = nextstat; E.codebegin := nextstat; emit('goto' --); }</pre> |

■ **true**和**false**都是终结符。对于 $E \rightarrow \text{true}$:

■ **p: goto --**

假设**nextstat**的初值为 **P**

广东工业大学计算机学院

$E.t = p$
 $E.cb = p$
 $E.f = \text{nil}$

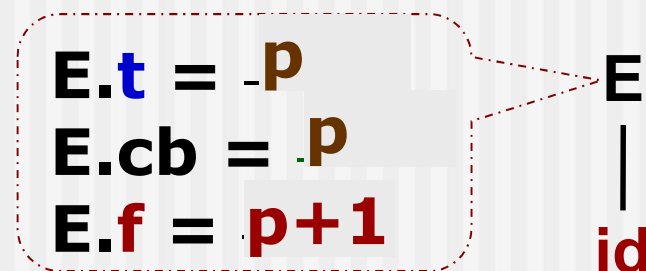
E
|
true
e

常见布尔表达式的翻译(续5)

| 布尔表达式 | 翻译(四元式语义描述) |
|------------------------|---|
| (8) $E \rightarrow id$ | <pre> { E.true := nextstat; E.codebegin := nextstat; E.false := nextstat + 1; emit('if' id.place 'goto' --); emit('goto' --) } </pre> |

- 在这样的程序段翻译中，会用到 $E \rightarrow id$ 作规约：
 if **p** **then** **S1** */*p是一个终结符*/*

- **p:** **if** **id** **goto** _____
 p+1: **goto** _____



假设**nextstat**的初值为 **p**