

控制流语句的基本文法

➤ $P \rightarrow S$

➤ $S \rightarrow S_1 S_2$

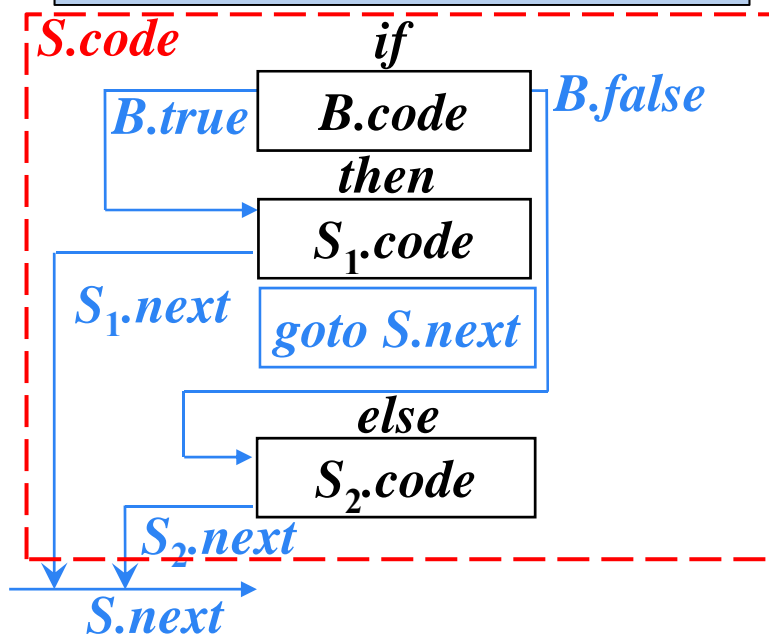
➤ $S \rightarrow \text{id} = E ; \mid L = E ;$

➤ $S \rightarrow \text{if } B \text{ then } S_1$
 $\mid \text{if } B \text{ then } S_1 \text{ else } S_2$
 $\mid \text{while } B \text{ do } S_1$

控制流语句的代码结构

➤ 例

$S \rightarrow \text{if } B \text{ then } S_1 \text{ else } S_2$



布尔表达式 B 被翻译成由
跳转指令构成的跳转代码

➤ 继承属性

- $S.next$: 是一个地址, 该地址中存放了紧跟在 S 代码之后的指令(S 的**后继指令**)的**标号**
- $B.true$: 是一个地址, 该地址中存放了当 B 为**真**时控制流转向的**指令的标号**
- $B.false$: 是一个地址, 该地址中存放了当 B 为**假**时控制流转向的**指令的标号**

用指令的**标号**标识一条三地址指令

控制流语句的SDT

newlabel(): 生成一个用于存放标号的新的临时变量 L , 返回变量地址

➤ $P \rightarrow \{ S.next = newlabel(); \} S \{ label(S.next); \}$

➤ $S \rightarrow \{ S_1.next = newlabel(); \} S_1$
 $\{ label(S_1.next); S_2.next = S.next ; \} S_2$

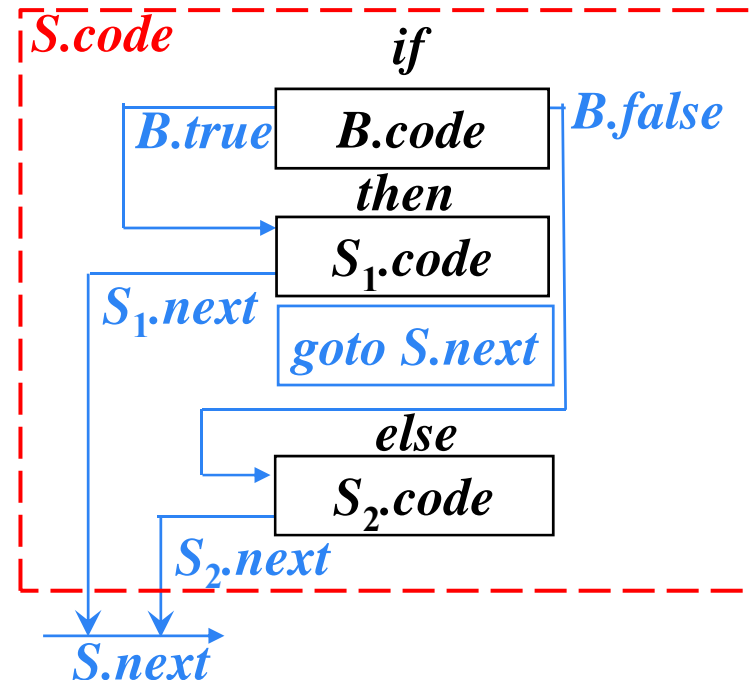
label(L): 将下一条三地址指令的标号赋给 L

➤ $S \rightarrow id = E ; \mid L = E ;$

➤ $S \rightarrow if\ B\ then\ S_1$
 $\mid if\ B\ then\ S_1\ else\ S_2$
 $\mid while\ B\ do\ S_1$

*if-then-else*语句的SDT

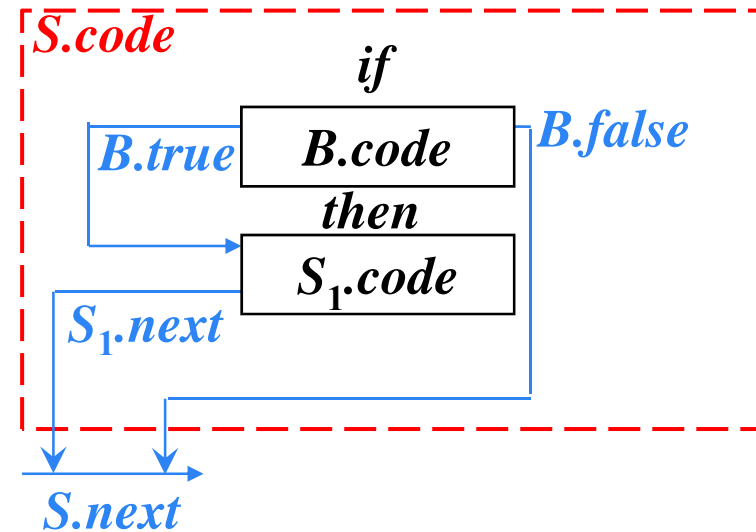
$S \rightarrow \text{if } B \text{ then } S_1 \text{ else } S_2$



$S \rightarrow \text{if } \{ B.true = \text{newlabel}(); B.false = \text{newlabel}(); \} B$
 $\quad \text{then } \{ \text{label}(B.true); S_1.next = S.next; \} S_1 \{ \text{gen}('goto' S.next) \}$
 $\quad \text{else } \{ \text{label}(B.false); S_2.next = S.next; \} S_2$

*if-then*语句的SDT

$S \rightarrow \text{if } B \text{ then } S_1$

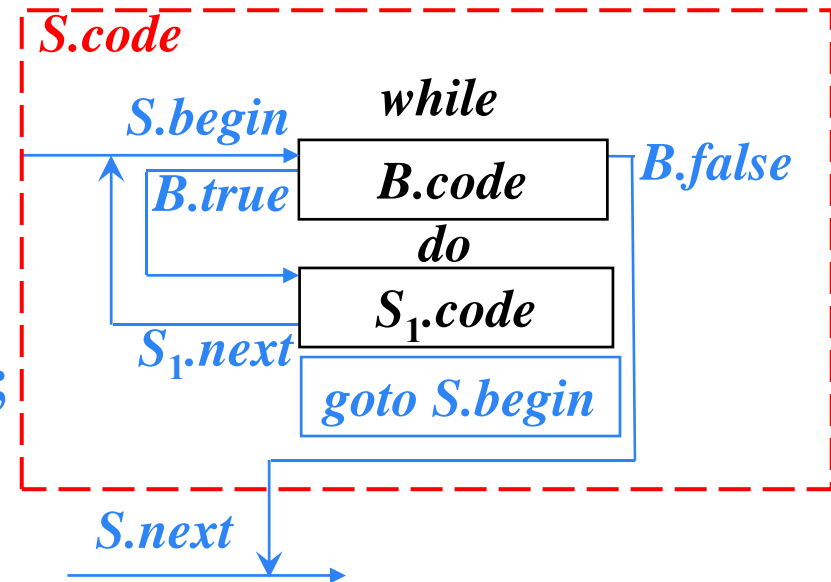


$S \rightarrow \text{if } \{ B.true = \text{newlabel}(); B.false = S.next; \} B$
 $\text{then } \{ \text{label}(B.true); S_1.next = S.next; \} S_1$

*while-do*语句的SDT

$S \rightarrow \text{while } B \text{ do } S_1$

$S \rightarrow \text{while } \{S.begin = \text{newlabel()};$
 $\text{label}(S.begin);$
 $B.true = \text{newlabel()};$
 $B.false = S.next;\} B$
 $\text{do } \{ \text{label}(B.true); S_1.next = S.begin;\} S_1$
 $\{ \text{gen('goto' } S.begin); \}$



布尔表达式的基本文法

$B \rightarrow B \text{ or } B$

| $B \text{ and } B$

| $\text{not } B$

| (B)

| $E \text{ relop } E$

| true

| false

优先级: $\text{not} > \text{and} > \text{or}$

关系表达式

relop (关系运算符) :
<, <=, >, >=, ==, !=

- 
- 在跳转代码中，逻辑运算符&&、|| 和! 被翻译成**跳转指令**。运算符本身**不出现在代码中**，布尔表达式的值是通过代码序列中的位置来表示的

- 例

- 语句

```
if ( x<100 || x>200 && x!=y )  
    x=0;
```

- 三地址代码

```
if x<100 goto L2  
goto L3  
L3 : if x>200 goto L4  
goto L1  
L4 : if x!=y goto L2  
goto L1  
L2 : x=0  
L1 :
```


布尔表达式的SDT

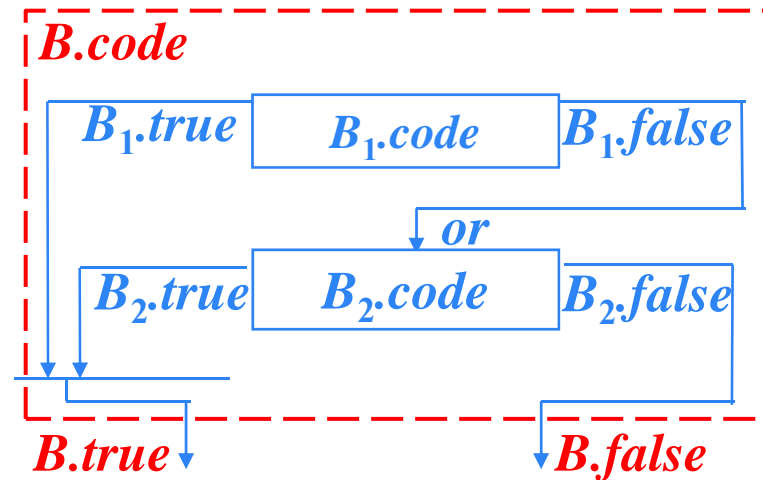
- $B \rightarrow E_1 \text{ relop } E_2 \{ \text{gen}(\text{'if' } E_1.\text{addr relop } E_2.\text{addr 'goto' } B.\text{true});$
 $\text{gen}(\text{'goto' } B.\text{false}); \}$
- $B \rightarrow \text{true} \{ \text{gen}(\text{'goto' } B.\text{true}); \}$
- $B \rightarrow \text{false} \{ \text{gen}(\text{'goto' } B.\text{false}); \}$
- $B \rightarrow (\{ B_1.\text{true} = B.\text{true}; B_1.\text{false} = B.\text{false}; \} B_1)$
- $B \rightarrow \text{not} \{ B_1.\text{true} = B.\text{false}; B_1.\text{false} = B.\text{true}; \} B_1$

$B \rightarrow B_1 \text{ or } B_2$ 的SDT

➤ $B \rightarrow B_1 \text{ or } B_2$

➤ $B \rightarrow \{ B_1.\text{true} = B.\text{true}; B_1.\text{false} = \text{newlabel}(); \} B_1$

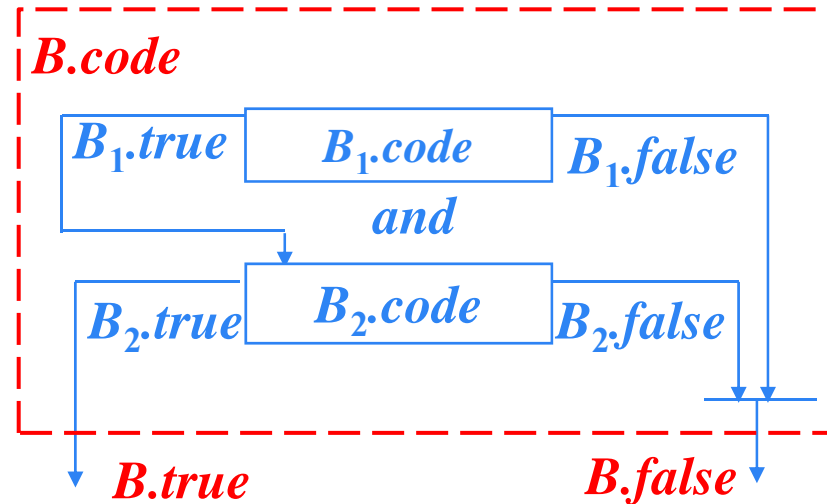
or $\{ \text{label}(B_1.\text{false}); B_2.\text{true} = B.\text{true}; B_2.\text{false} = B.\text{false}; \} B_2$



$B \rightarrow B_1 \text{ and } B_2$ 的SDT

➤ $B \rightarrow B_1 \text{ and } B_2$

➤ $B \rightarrow \{ B_1.\text{true} = \text{newlabel}(); B_1.\text{false} = B.\text{false}; \} B_1$
and $\{ \text{label}(B_1.\text{true}); B_2.\text{true} = B.\text{true}; B_2.\text{false} = B.\text{false}; \} B_2$



控制流语句的 SDT

- $P \rightarrow \{a\}S\{a\}$
- $S \rightarrow \{a\}S_1\{a\}S_2$
- $S \rightarrow \text{id}=E;\{a\} \mid L=E;\{a\}$
- $E \rightarrow E_1+E_2\{a\} \mid -E_1\{a\} \mid (E_1)\{a\} \mid \text{id}\{a\} \mid L\{a\}$
- $L \rightarrow \text{id}[E]\{a\} \mid L_1[E]\{a\}$
- $S \rightarrow \text{if } \{a\}B \text{ then } \{a\}S_1$
 - | $\text{if } \{a\}B \text{ then } \{a\}S_1 \text{ else } \{a\}S_2$
 - | $\text{while } \{a\}B \text{ do } \{a\}S_1\{a\}$
- $B \rightarrow \{a\}B \text{ or } \{a\}B \mid \{a\}B \text{ and } \{a\}B \mid \text{not } \{a\}B \mid (\{a\}B)$
 - | $E \text{ relop } E\{a\} \mid \text{true}\{a\} \mid \text{false}\{a\}$

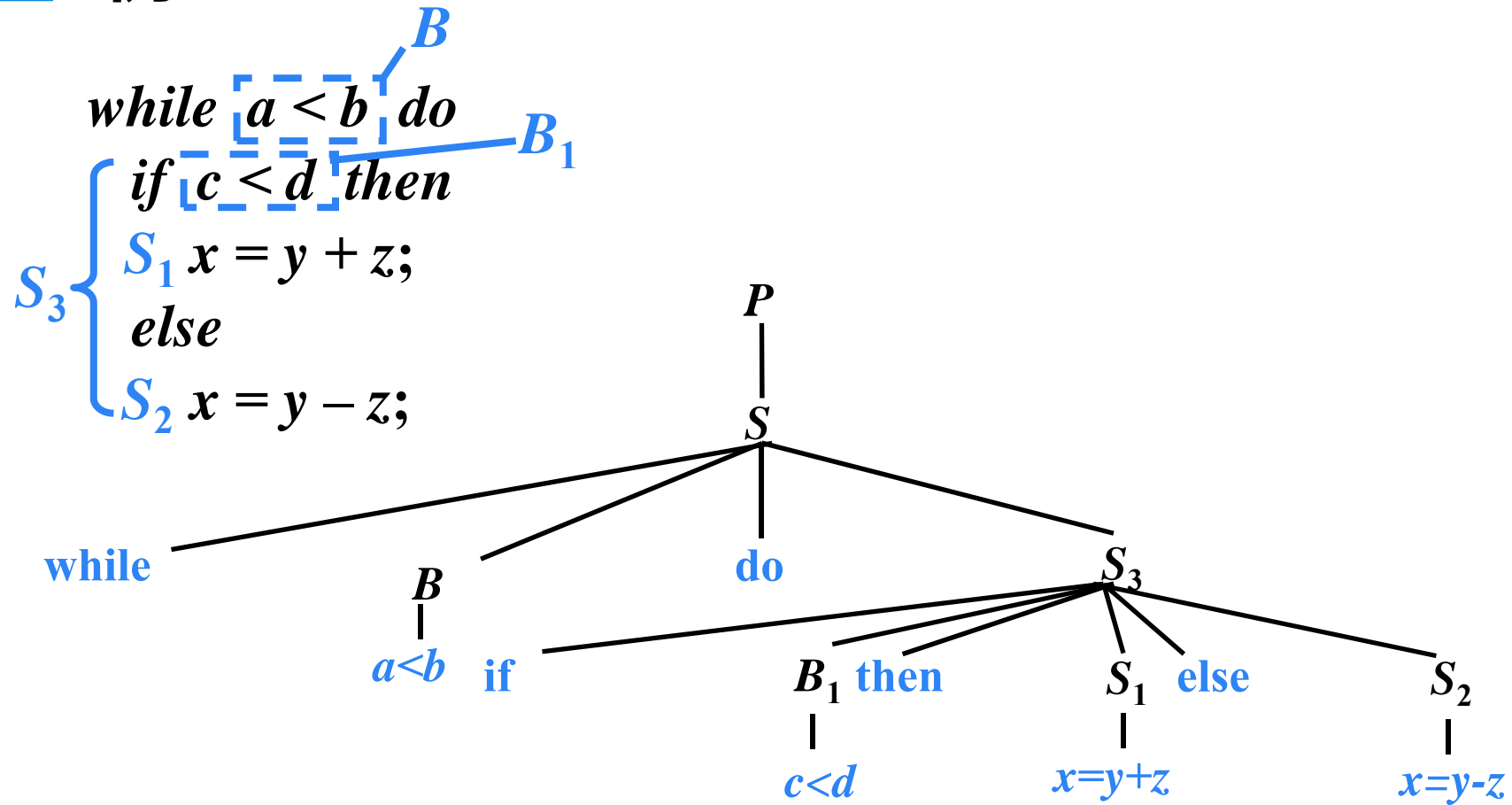
***SDT*的通用实现方法**

- 任何*SDT*都可以通过下面的方法实现
 - 首先建立一棵语法分析树，然后按照从左到右的深度优先顺序来执行这些动作

控制流语句的 SDT

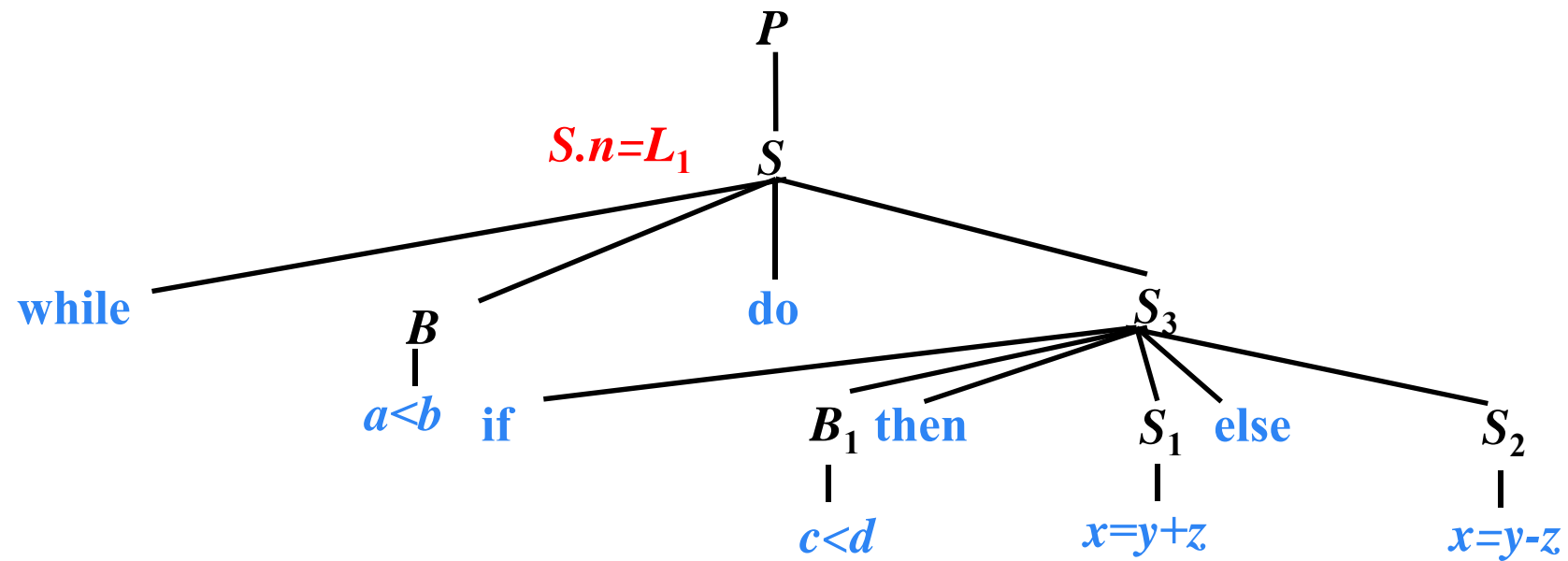
- $P \rightarrow \{a\}S\{a\}$
- $S \rightarrow \{a\}S_1\{a\}S_2$
- $S \rightarrow \text{id}=E;\{a\} \mid L=E;\{a\}$
- $E \rightarrow E_1+E_2\{a\} \mid -E_1\{a\} \mid (E_1)\{a\} \mid \text{id}\{a\} \mid L\{a\}$
- $L \rightarrow \text{id}[E]\{a\} \mid L_1[E]\{a\}$
- $S \rightarrow \text{if } \{a\}B \text{ then } \{a\}S_1$
 - | $\text{if } \{a\}B \text{ then } \{a\}S_1 \text{ else } \{a\}S_2$
 - | $\text{while } \{a\}B \text{ do } \{a\}S_1\{a\}$
- $B \rightarrow \{a\}B \text{ or } \{a\}B \mid \{a\}B \text{ and } \{a\}B \mid \text{not } \{a\}B \mid (\{a\}B)$
 - | $E \text{ relop } E\{a\} \mid \text{true}\{a\} \mid \text{false}\{a\}$

例





$P \rightarrow \{ S.next = newlabel(); \} S \{ label(S.next); \}$

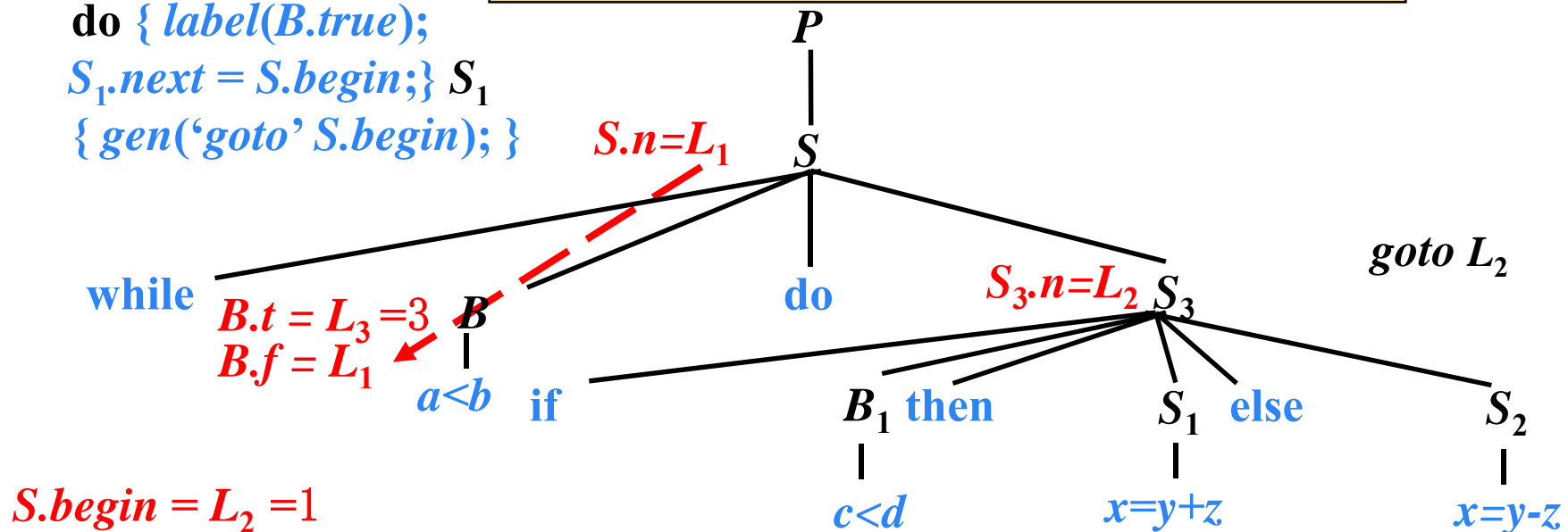


例

1: if $a < b$ goto L_3
2: goto L_1

$S \rightarrow$ while { $S.begin = newlabel()$;
 $label(S.begin)$;
 $B.true = newlabel()$;
 $B.false = S.next$;} B
do { $label(B.true)$;
 $S_1.next = S.begin$;} S_1
{ $gen('goto' S.begin)$;

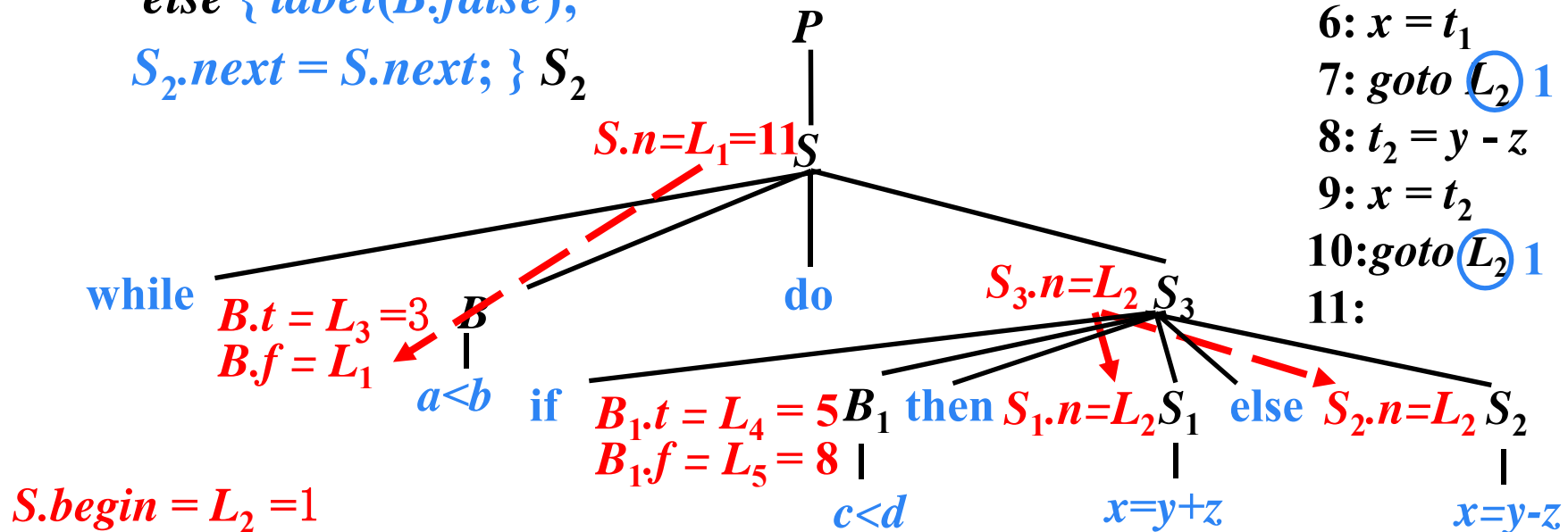
$B \rightarrow E_1 \text{ relop } E_2$
{ $gen('if' E_1.addr \text{ relop } E_2.addr \text{ 'goto' } B.true)$;
 $gen('goto' B.false)$;





$S \rightarrow \text{if } \{ B.true = \text{newlabel}(); B.false = \text{newlabel}(); \} B$
 $\text{then } \{ \text{label}(B.true); S_1.next = S.next; \} S_1$
 $\{ \text{gen}('goto' S.next); \}$
 $\text{else } \{ \text{label}(B.false);$
 $S_2.next = S.next; \} S_2$

1: $\text{if } a < b \text{ goto } \textcircled{L_2}^3$
 2: $\text{goto } \textcircled{L_1}^{11}$
 3: $\text{if } c < d \text{ goto } \textcircled{L_4}^5$
 4: $\text{goto } \textcircled{L_5}^8$
 5: $t_1 = y + z$
 6: $x = t_1$
 7: $\text{goto } \textcircled{L_2}^1$
 8: $t_2 = y - z$
 9: $x = t_2$
 10: $\text{goto } \textcircled{L_2}^1$
 11:



语句 “*while* $a < b$ *do if* $c < d$ *then* $x = y + z$ *else* $x = y - z$ ” 的三地址代码

1: *if* $a < b$ *goto* 3

2: *goto* 11

3: *if* $c < d$ *goto* 5

4: *goto* 8

5: $t_1 = y + z$

6: $x = t_1$

7: *goto* 1

8: $t_2 = y - z$

9: $x = t_2$

10: *goto* 1

11:

1: ($j <$, a , b , 3)

2: (j , -, -, 11)

3: ($j <$, c , d , 5)

4: (j , -, -, 8)

5: (+, y , z , t_1)

6: (=, t_1 , -, x)

7: (j , -, -, 1)

8: (-, y , z , t_2)

9: (=, t_2 , -, x)

10: (j , -, -, 1)

11: