



编译系统
第六章
中间代码生成

哈尔滨工业大学 陈鄞



第13讲（中间代码生成_3）要点

➤ 赋值语句的翻译

➤ 主要任务：生成对表达式求值的三地址码

➤ 难点：数组元素的寻址

➤ 关键：地址计算公式 \longleftrightarrow 文法

第13讲（中间代码生成_3）要点

➤ 控制流语句的翻译

➤ 基础文法

➤ $P \rightarrow S$

➤ $S \rightarrow S_1 S_2$

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

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

$\mid \text{if } B \text{ then } S_1 \text{ else } S_2$

$\mid \text{while } B \text{ do } S_1$

控制流语句的代码结构

➤ 例

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

S.code

if

B.code

then

S₁.code

else

S₂.code

例1: B="a<b"

三地址指令

if a<b goto *S₁.first*

goto *S₂.first*

标号(常量)

问题: 生成跳转指令的时候, *S₁.first*和
*S₂.first*的值还**不知道**

例2: B="a<b || a>200 && b<100"

布尔表达式*B*被翻译成由
跳转指令构成的跳转代码

用指令的**标号**标识一条三地址指令

控制流语句的代码结构

➤ 例

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

S.code

if

B.code

then

S₁.code

else

S₂.code

例1: B="a<b"

三地址指令

if a<b goto *S₁.first*

goto *S₂.first*

标号(常量)

临时指令

if a<b goto *B.true*

goto *B.false*

地址(变量)

布尔表达式*B*被翻译成由
跳转指令构成的跳转代码

用指令的标号标识一条三地址指令

控制流语句的代码结构

➤ 例

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

S.code

if

B.code

then

S₁.code

else

S₂.code

例1: B="a<b"

三地址指令

if a<b goto **S₁.first**

goto **S₂.first**

临时指令

if a<b goto **B.true**

goto **B.false**

if a<b goto (**B.true**)

goto (**B.false**)

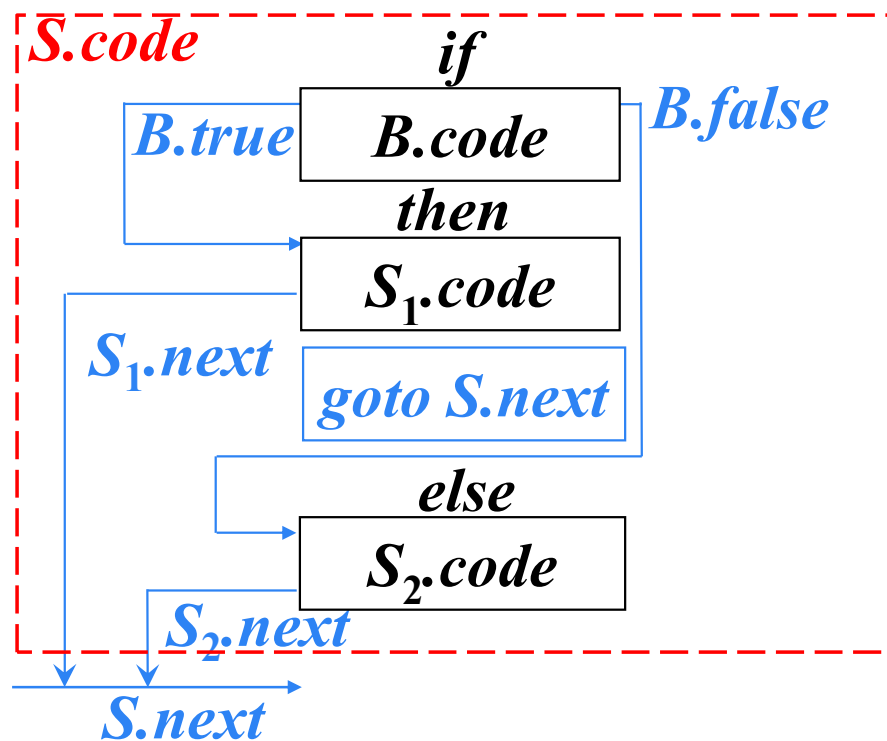
布尔表达式*B*被翻译成由
跳转指令构成的跳转代码

用指令的标号标识一条三地址指令

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

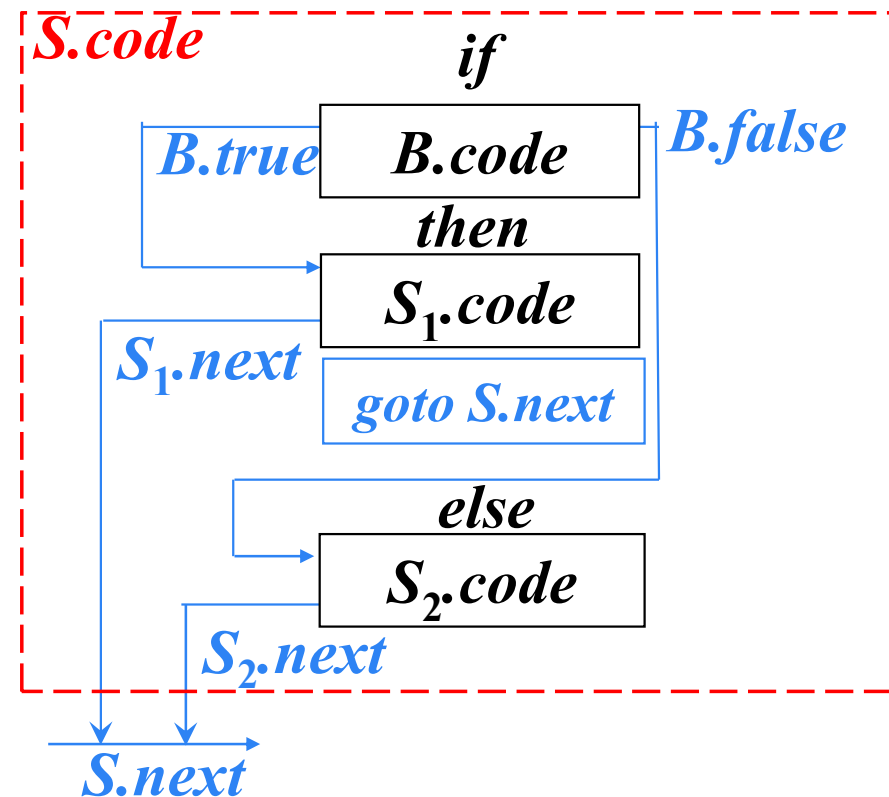
➤ 继承属性

- $B.true$: 是一个地址, 该地址用来存放当 B 为真时控制流转向的指令的标号
- $B.false$: 是一个地址, 该地址用来存放当 B 为假时控制流转向的指令的标号
- $S.next$: 是一个地址, 该地址用来存放紧跟在 S 代码之后执行的指令(S 的后继指令)的标号

用指令的标号标识一条三地址指令

if-then-else语句的SDT

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



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

SDT编写要点

- 分析每一个非终结符之前
 - 先计算继承属性
 - 再观察代码结构图中该非终结符对应的方框顶部是否有导入箭头。如果有，调用label()函数
- 上一个代码框执行完不顺序执行下一个代码框时，生成一条显式跳转指令
- 有自下而上的箭头时，设置begin属性。且定义后直接调用label()函数绑定地址

总结

- 控制流语句翻译的一个关键是确定跳转指令的目标标号
- 存在问题：生成跳转指令时，目标标号还不能确定
- 解决办法：生成一些临时变量用来存放标号，将临时变量的地址作为继承属性传递到标号可以确定的地方。也就是说，当目标标号的值确定下来以后再赋给相应的变量
- 缺点：需要进行两遍处理
 - 第一遍生成临时的指令
 - 第二遍将指令中的临时变量的地址改为具体的标号，从而得到最终的三地址指令



结束

