

第六章（中间代码生成）作业

6-4

P263: 6.6.1 (中文版厚书)

P246: 6.6.1 (中文版薄书)

练习 6.6.1：在图 6-36 的语法制导定义中添加处理下列控制流构造的规则：

1) 一个 repeat 语句，**repeat S while B**。

注：原题第 1) 小题。

产生式	语义规则
$P \rightarrow S$	$S.next = newlabel()$ $P.code = S.code \parallel label(S.next)$
$S \rightarrow \text{assign}$	$S.code = \text{assign}.code$
$S \rightarrow \text{if} (B) S_1$	$B.true = newlabel()$ $B.false = S_1.next = S.next$ $S.code = B.code \parallel label(B.true) \parallel S_1.code$
$S \rightarrow \text{if} (B) S_1 \text{ else } S_2$	$B.true = newlabel()$ $B.false = newlabel()$ $S_1.next = S_2.next = S.next$ $S.code = B.code$ $\quad \parallel label(B.true) \parallel S_1.code$ $\quad \parallel \text{gen('goto' } S.next)$ $\quad \parallel label(B.false) \parallel S_2.code$
$S \rightarrow \text{while} (B) S_1$	$begin = newlabel()$ $B.true = newlabel()$ $B.false = S.next$ $S_1.next = begin$ $S.code = label(begin) \parallel B.code$ $\quad \parallel label(B.true) \parallel S_1.code$ $\quad \parallel \text{gen('goto' } begin)$
$S \rightarrow S_1 S_2$	$S_1.next = newlabel()$ $S_2.next = S.next$ $S.code = S_1.code \parallel label(S_1.next) \parallel S_2.code$

图 6-36 控制流语句的语法制导定义

P263: 6.6.1 (中文版厚书)

P246: 6.6.1 (中文版薄书)

练习 6.6.1: 在图 6-36 的语法制导定义中添加处理下列控制流构造的规则:

! 2) 一个 for 循环语句, $\text{for } (S_1; B; S_2) S_3$ 。

注: 原题第 2) 小题。

产生式	语义规则
$P \rightarrow S$	$S.\text{next} = \text{newlabel}()$ $P.\text{code} = S.\text{code} \parallel \text{label}(S.\text{next})$
$S \rightarrow \text{assign}$	$S.\text{code} = \text{assign}.\text{code}$
$S \rightarrow \text{if } (B) S_1$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = S_1.\text{next} = S.\text{next}$ $S.\text{code} = B.\text{code} \parallel \text{label}(B.\text{true}) \parallel S_1.\text{code}$
$S \rightarrow \text{if } (B) S_1 \text{ else } S_2$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = \text{newlabel}()$ $S_1.\text{next} = S_2.\text{next} = S.\text{next}$ $S.\text{code} = B.\text{code}$ $\quad \parallel \text{label}(B.\text{true}) \parallel S_1.\text{code}$ $\quad \parallel \text{gen('goto' } S.\text{next})$ $\quad \parallel \text{label}(B.\text{false}) \parallel S_2.\text{code}$
$S \rightarrow \text{while } (B) S_1$	$\text{begin} = \text{newlabel}()$ $B.\text{true} = \text{newlabel}()$ $B.\text{false} = S.\text{next}$ $S_1.\text{next} = \text{begin}$ $S.\text{code} = \text{label(begin)} \parallel B.\text{code}$ $\quad \parallel \text{label}(B.\text{true}) \parallel S_1.\text{code}$ $\quad \parallel \text{gen('goto' begin)}$
$S \rightarrow S_1 S_2$	$S_1.\text{next} = \text{newlabel}()$ $S_2.\text{next} = S.\text{next}$ $S.\text{code} = S_1.\text{code} \parallel \text{label}(S_1.\text{next}) \parallel S_2.\text{code}$

图 6-36 控制流语句的语法制导定义

P268: 6.7.1 (中文版厚书)

P251: 6.7.1 (中文版薄书)

练习 6.7.1: 使用图 6-43 中的翻译方案翻译下列表达式。给出每个子表达式的 *truelist* 和 *falselist*。你可以假设第一条被生成的指令的地址是 100。

1) $a == b \ \&\& \ (c == d \ || \ e == f)$

注: 原题第 1) 小题。

- | | |
|---|--|
| 1) $B \rightarrow B_1 \ \ M \ B_2$ | { <i>backpatch</i> ($B_1.falselist, M.instr$);
$B.trueclist = merge(B_1.trueclist, B_2.trueclist);$
$B.falselist = B_2.falselist;$ } |
| 2) $B \rightarrow B_1 \ \&\& \ M \ B_2$ | { <i>backpatch</i> ($B_1.trueclist, M.instr$);
$B.trueclist = B_2.trueclist;$
$B.falselist = merge(B_1.falselist, B_2.falselist);$ } |
| 3) $B \rightarrow ! B_1$ | { $B.trueclist = B_1.falselist;$
$B.falselist = B_1.trueclist;$ } |
| 4) $B \rightarrow (B_1)$ | { $B.trueclist = B_1.trueclist;$
$B.falselist = B_1.falselist;$ } |
| 5) $B \rightarrow E_1 \ rel \ E_2$ | { $B.trueclist = makelist(nextinstr);$
$B.falselist = makelist(nextinstr + 1);$
$gen('if' \ E_1.addr \ rel.op \ E_2.addr \ 'goto \ '_{});$
$gen('goto \ '_{});$ } |
| 6) $B \rightarrow \text{true}$ | { $B.trueclist = makelist(nextinstr);$
$gen('goto \ '_{});$ } |
| 7) $B \rightarrow \text{false}$ | { $B.falselist = makelist(nextinstr);$
$gen('goto \ '_{});$ } |
| 8) $M \rightarrow \epsilon$ | { $M.instr = nextinstr;$ } |

图 6-43 布尔表达式的翻译方案