

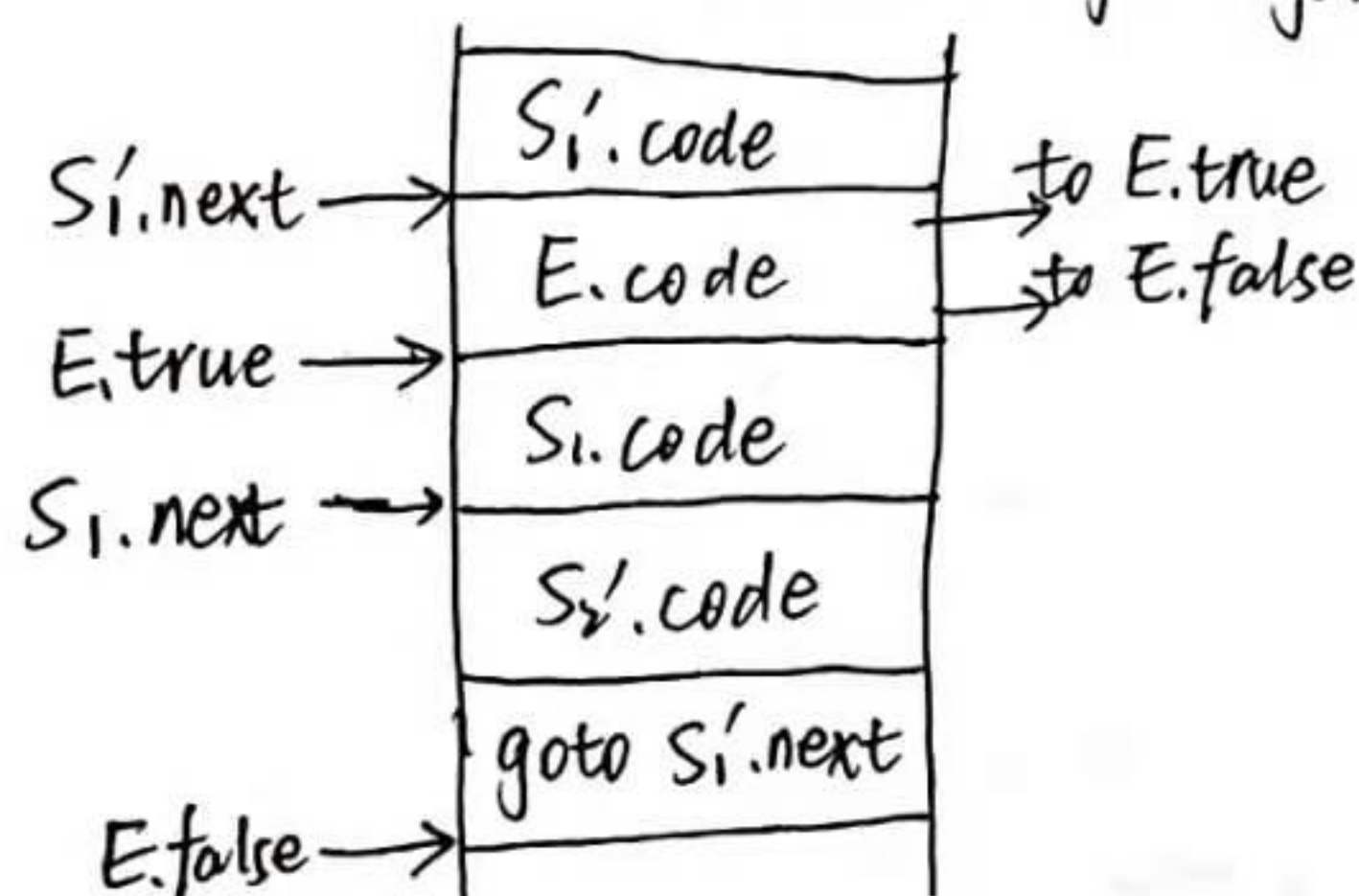
数学作业纸

班级: 计84

姓名: 刘泓源 编号: 2018011446 科目:

第 页

4.

$$\begin{aligned}
 S \rightarrow & \text{for} (\{ S_1'.\text{next} := \text{newlabel} \} S_1'; \\
 & \{ E.\text{true} := \text{newlabel}; E.\text{false} := S.\text{next} \} E; \\
 & \{ S_2'.\text{next} := S_1'.\text{next} \} S_2'; \\
 & \{ S_1.\text{next} := \text{newlabel} \} S_1; \\
 & \{ S.\text{code} := S_1'.\text{code} \parallel \text{gen}(S_1'.\text{next}::') \parallel E.\text{code} \parallel \\
 & \text{gen}(E.\text{true}::') \parallel S_1.\text{code} \parallel \text{gen}(S_1.\text{next}::') \parallel \\
 & S_2'.\text{code} \parallel \text{gen}(\text{'goto' } S_1'.\text{next}) \}
 \end{aligned}$$


5. 解:

$$\begin{aligned}
 S \rightarrow & \text{for} (S_1'; M_1 E; M_2 S_2' N_1) M_3 S_1 N_2 \\
 & \{ \text{backpatch}(E.\text{truelist}, M_3.\text{goto stm}); \\
 & \text{backpatch}(S_1.\text{nextlist}, M_2.\text{goto stm}); \\
 & \text{backpatch}(S_1'.\text{nextlist}, M_1.\text{goto stm}); \\
 & \text{backpatch}(S_2'.\text{nextlist}, M_1.\text{goto stm}); \\
 & \text{backpatch}(N_1.\text{nextlist}, M_1.\text{goto stm}); \\
 & \text{backpatch}(N_2.\text{nextlist}, M_2.\text{goto stm}); \\
 & S_1.\text{nextlist} := E.\text{falselist}; \}
 \end{aligned}$$

10. 解:

(a) $E \rightarrow E_1 \uparrow E_2$

```

{ E2.label = E1.label ;
  E2.case = not E1.case ;
  E1.case = false ;
  if E1.case {
    E1.label = E1.label
    E.code = E1.code || E2.code
  } else {
    E1.label = newlabel ;
    E.code = E1.code || E2.code || gen(E1.label ':')
  }
}

```

(若 E_1 .case 为 1, 则 E_1 .label 为 false 时直接跳到 E_1 .label 处)

(若 E_1 .case 为 0, 则新生成一个 label 当 E_1 为 false 时跳到 newlabel, 不用计算 E_2)

(b)

 $S \rightarrow \text{repeat } S_1 \text{ until } E$

```

{ S1.next = newlabel ;
  E.case = false ;
  E.label = S1.next ;
  S1.code = gen(S1.next ':') || S1.code || E.code || gen(S1.next ':')
}

```

(注意 until 和 while 意义相反)

A1.

```

E → { E1.false = E.false ; E1.true = newlabel } E1 ?
{ E2.false = E.false ; E2.true = newlabel } E2 : E1.true
{ E3.true = E.false ; E3.false = E1.true } E3 E2.true
{ E.code = E1.code || gen(E1.true ':') || E2.code ||
  gen(E2.true ':') || E3.code }

```

E1.code	→
E2.code	→
E3.code	→

A2:

```

E → E1 ? M1 E2 : M2 E3 { backpatch(E1.true list, M1.goto stm) ;
  backpatch(E2.true list, M2.goto stm) ;
  E.true list = E3.true list ;
  E.false list = merge(merge(E1.false list, E2.false list), E3.true list) ;
}

```


A3. (1)

$A \rightarrow A_1 + A_2 \{ A.instr = A_1.instr \parallel A_2.instr \parallel Plus \}$

$A \rightarrow A_1 - A_2 \{ A.instr = A_2.instr \parallel A_1.instr \parallel Minus \}$

(2)

$E \rightarrow E_1 \text{ if } B \{ E.instr = E_1.instr \parallel B.instr \parallel Cond \}$

$A \rightarrow id \{ A.instr = Load \ id.val \}$

$B \rightarrow A_1 > A_2 \{ B.instr = A_2.instr \parallel A_1.instr \parallel Minus \parallel Cmp \parallel A_2.instr \parallel A_1.instr \parallel Minus \parallel Cond \}$

$B \rightarrow B_1 \& B_2 \{ B.instr = B_2.instr \parallel B_1.instr \parallel Cond \}$

$B \rightarrow ! B_1 \{ B.instr = Push \ 1 \parallel Push \ 1 \parallel B_1.instr \parallel Minus \parallel Cond \}$

$B \rightarrow true \{ B.instr = Push \ 1 \}$

$B \rightarrow false \{ B.instr = Push \ 0 \}$

$S \rightarrow id = E \{ S.instr = E.instr \parallel Store \ id.val \}$

$S \rightarrow S_1 ; S_2 \{ S.instr = S_1.instr \parallel S_2.instr \}$

$(A_1 - A_2 \geq 0) \text{ if } (A_1 - A_2 \neq 0) \text{ else } 0$

A4. (1)

$L \rightarrow id \{ L.types = [lookup_type(id.entry)] \}$

$L \rightarrow id, L_1 \{ L.types = [lookup_type(id.entry)] + L_1.types \}$

$R \rightarrow E \{ R.types = [E.type] \}$

$R \rightarrow E, R_1 \{ R.types = [E.type] + R_1.types \}$

$S \rightarrow L := R \{ S.type = \text{if } L.types = R.types \text{ then ok else type_error} \}$

(2)

$L \rightarrow id \{ L.places = [id.place] \}$

$L \rightarrow id, L_1 \{ L.places = [id.place] + L_1.places \}$

$R \rightarrow E \{ R.places = [E.place]; R.codes = [E.code] \}$

$R \rightarrow E, R_1 \{ R.places = [E.place] + R_1.places; R.codes = [E.code] + R_1.codes \}$

$S \rightarrow L := R \{ S.code = ""; \text{ for } i=0 \text{ to } \text{len}(R.codes)-1 \text{ do } S.code = S.code \parallel R.codes[i] \text{ end;}$

$\text{for } i=0 \text{ to } \text{len}(R.places)-1 \text{ do}$

$S.code = S.code \parallel \text{gen}(L.places[i], R.places[i]); \text{ end;}$

(3)

 $S \rightarrow \dots \{ \dots$ for $i = \text{len}(R.\text{places})$ to $\text{len}(L.\text{places}) - 1$ do $S.\text{code} = S.\text{code} \parallel \text{gen}(L.\text{places}[i] := _)$ if $L.\text{type} = \text{bnd}$ thenelse $S.\text{ebool} := [L.\text{places}[i]] + S.\text{ebool}$ end $\{ S.\text{eint} := [L.\text{places}[i]] + S.\text{eint}$ $S \rightarrow \text{begin } S_1 \text{ end default } E_1, E_2 \{ S.\text{code} := E_1.\text{code} \parallel E_2.\text{code} \parallel$ $S_1.\text{code} ;$ $\text{backpatch}(S_1, \text{ebool}, E_1, \text{place});$ $\text{backpatch}(S_1, \text{eint}, E_2, \text{place}); \}$