四、中间代码生成 (5. 回填技术)

魏恒峰

hfwei@nju.edu.cn

2023 年 05 月 19 日

为什么需要 “回填技术”?

```
outer:
for (int i = 2; i < 1000; i++) {
    for (int j = 2; j < i; j++) {
        if (i % j == 0)
            continue outer;
    }
    System.out.println (i);
}
```

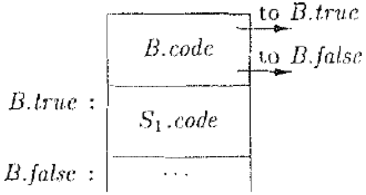
```
0:   iconst_2
1:   istore_1
2:   iload_1
3:   sipush   1000
6:   if_icmpge       44
9:   iconst_2
10:  istore_2
11:  iload_2
12:  iload_1
13:  if_icmpge       31
16:  iload_1
17:  iload_2
18:  irem
19:  ifne     25
22:  goto     38
25:  iinc     2, 1
28:  goto     11
31:  getstatic       #84;
34:  iload_1
35:  invokevirtual   #85;
38:  iinc     1, 1
41:  goto     2
44:  return
```

回填技术：在一趟（one-pass）中生成跳转目标地址（而非目标标签）

$S \rightarrow \textbf{if}\ (\ B\ )\ S_1$

$$\begin{array}{l} B.true = newlabel() \\ B.false = S_1.next = S.next \\ S.code = B.code \mid\mid label(B.true) \mid\mid S_1.code \end{array}$$

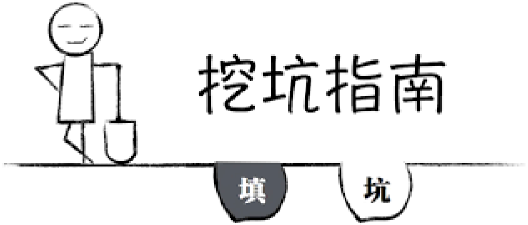
B 可以自行计算 $B.true$ 对应的指令地址



B 计算不出 $B.false$ 对应的指令地址

回填 (Backpatching) 技术

子节点挖坑、祖先节点填坑



子节点暂时不指定跳转指令的目标地址

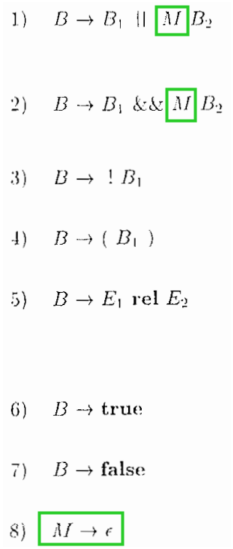
待祖先节点能够确定目标地址时回头填充

父节点通过**综合属性**收集子节点中具有相同目标的跳转指令

为左部非终结符 B 计算综合属性 B.truelist 与 B.falselist

为左部非终结符 S/L 计算综合属性 S/L.nextlist

并为已能确定目标地址的跳转指令进行回填 (考虑每个综合属性)



1) $B \to B_1 \mid\mid \boxed{M} B_2$

2) $B \to B_1 \;\&\&\; \boxed{M} B_2$

3) $B \to \; ! \; B_1$

4) $B \to (\; B_1 \;)$

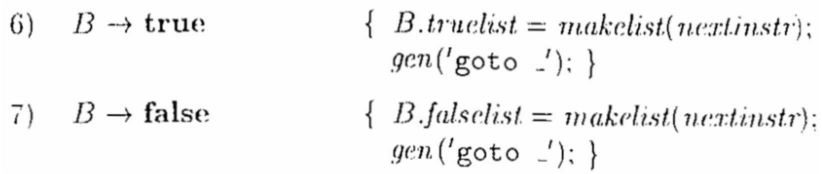
5) $B \to E_1 \; \mathbf{rel} \; E_2$

6) $B \to \mathbf{true}$

7) $B \to \mathbf{false}$

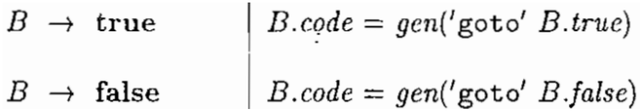
8) $\boxed{M \to \epsilon}$

$B.truelist$ 保存需要跳转到 $B.true$ 标签的指令

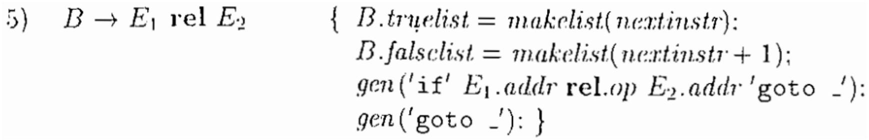


6)	$B \to$ **true**	$\{$ $B.truelist = makelist(nextinstr);$ $gen($'goto _'$);$ $\}$
7)	$B \to$ **false**	$\{$ $B.falselist = makelist(nextinstr);$ $gen($'goto _'$);$ $\}$

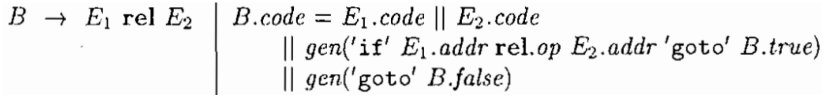
$B.falselist$ 保存需要跳转到 $B.false$ 标签的指令



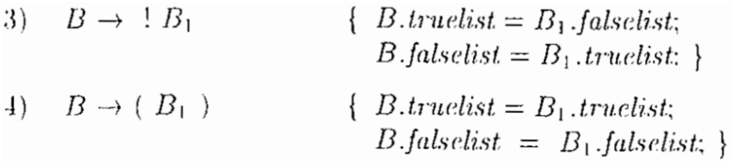
$B \to$ **true**	$B.code = gen($'goto'$\ B.true)$
$B \to$ **false**	$B.code = gen($'goto'$\ B.false)$



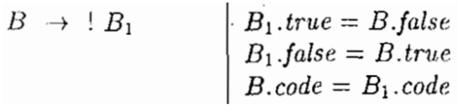
5) $B \to E_1$ **rel** E_2 { $B.truelist = makelist(nextinstr);$
$B.falselist = makelist(nextinstr + 1);$
$gen($'`if`' $E_1.addr$ **rel**$.op$ $E_2.addr$ '`goto _`'$);$
$gen($'`goto _`'$);$ }



$B \to E_1$ **rel** E_2	$B.code = E_1.code \mid\mid E_2.code$ $\mid\mid gen($'`if`' $E_1.addr$ **rel**$.op$ $E_2.addr$ '`goto`' $B.true)$ $\mid\mid gen($'`goto`' $B.false)$



3)	$B \to \; ! \; B_1$	$\{ \; B.truelist = B_1.falselist; \; B.falselist = B_1.truelist; \; \}$
4)	$B \to (\; B_1 \;)$	$\{ \; B.truelist = B_1.truelist; \; B.falselist = B_1.falselist; \; \}$



$B \to \; ! \; B_1$	$B_1.true = B.false$ $B_1.false = B.true$ $B.code = B_1.code$

$$2)\quad B \rightarrow B_1 \;\&\&\; M\; B_2 \qquad \{\; \boxed{backpatch(B_1.truelist, M.instr);} \\ \boxed{\begin{array}{l} B.truelist = B_2.truelist; \\ B.falselist = merge(B_1.falselist, B_2.falselist); \end{array}} \;\}$$

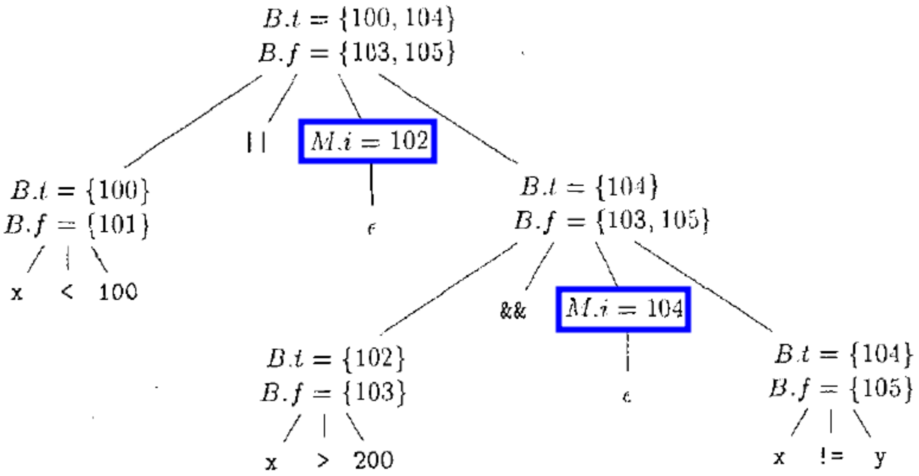
$$8)\quad M \rightarrow \epsilon \qquad \{\; M.instr = nextinstr;\; \}$$

$$B \;\rightarrow\; B_1 \;\&\&\; B_2 \quad \left| \begin{array}{l} \boxed{B_1.true = newlabel()} \\ \boxed{\begin{array}{l} B_1.false = B.false \\ B_2.true = B.true \\ B_2.false = B.false \end{array}} \\ B.code = B_1.code \;||\; \boxed{label(B_1.true)} \;||\; B_2.code \end{array} \right.$$

1) $B \to B_1 \mid\mid M\ B_2$ $\{\ \boxed{backpatch(B_1.falselist, M.instr);}$
$B.truelist = merge(B_1.truelist, B_2.truelist);$
$B.falselist = B_2.falselist;\ \}$

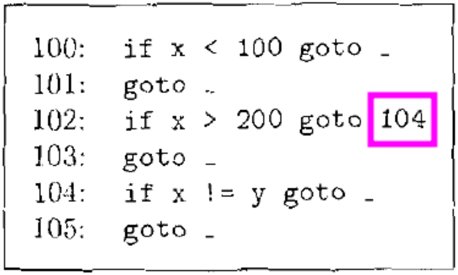
8) $M \to \epsilon$ $\{\ M.instr = nextinstr;\ \}$

$$
\begin{array}{l|l}
B \to B_1 \mid\mid B_2 & B_1.true = B.true \\
& \boxed{B_1.false = newlabel()} \\
& B_2.true = B.true \\
& B_2.false = B.false \\
& B.code = B_1.code \mid\mid \boxed{label(B_1.false)} \mid\mid B_2.code
\end{array}
$$



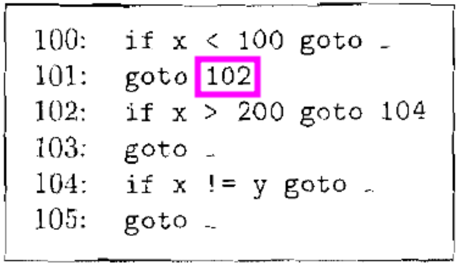
$B.t = \{100, 104\}$
$B.f = \{103, 105\}$
||
$M.i = 102$
ϵ
$B.t = \{100\}$
$B.f = \{101\}$
x < 100
$B.t = \{104\}$
$B.f = \{103, 105\}$
&&
$M.i = 104$
ϵ
$B.t = \{102\}$
$B.f = \{103\}$
x > 200
$B.t = \{104\}$
$B.f = \{105\}$
x != y

`x < 100 || x > 200 && x != y`



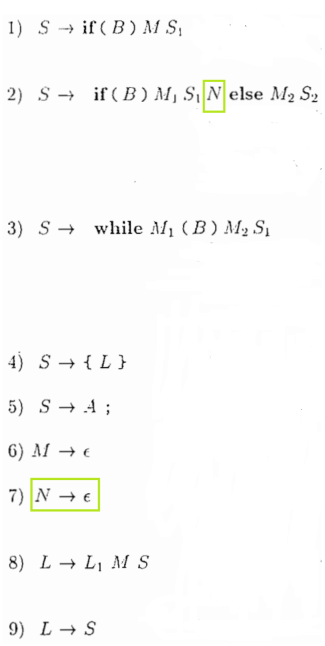
```
100:   if x < 100 goto _
101:   goto _
102:   if x > 200 goto 104
103:   goto _
104:   if x != y goto _
105:   goto _
```

a) 将 104 回填到指令 102 中之后



```
100:   if x < 100 goto _
101:   goto 102
102:   if x > 200 goto 104
103:   goto _
104:   if x != y goto _
105:   goto _
```

b) 将 102 回填到指令 101 中之后



1) $S \to \textbf{if}\ (\ B\ )\ M\ S_1$

2) $S \to \textbf{if}\ (\ B\ )\ M_1\ S_1\ \boxed{N}\ \textbf{else}\ M_2\ S_2$

3) $S \to \textbf{while}\ M_1\ (\ B\ )\ M_2\ S_1$

4) $S \to \{\ L\ \}$

5) $S \to A\ ;$

6) $M \to \epsilon$

7) $\boxed{N \to \epsilon}$

8) $L \to L_1\ M\ S$

9) $L \to S$

$$
\begin{aligned}
1)\quad S \to \textbf{if}\,(\,B\,)\;M\;S_1\quad \{\ & \boxed{backpatch}(B.truelist,\ M.instr);\\
& \boxed{S.nextlist} = merge(B.falselist,\ S_1.nextlist);\ \}
\end{aligned}
$$

$$
6)\quad M \to \epsilon \qquad\qquad \{\ M.instr = nextinstr;\ \}
$$

$$
S \to \textbf{if}\,(\,B\,)\;S_1 \qquad
\left|\begin{array}{l}
\boxed{B.true} = newlabel()\\
B.false = S_1.next = \boxed{S.next}\\
S.code = B.code \mathbin{||} \boxed{label(B.true)} \mathbin{||} S_1.code
\end{array}\right.
$$

$$
\begin{aligned}
S \rightarrow\ & \textbf{if}\ (\ B\ )\ M_1\ S_1\ N\ \textbf{else}\ M_2\ S_2 \\
& \{\ \boxed{backpatch}(B.truelist,\ M_1.instr); \\
& \ \ \boxed{backpatch}(B.falselist,\ M_2.instr); \\
& \ \ temp\ =\ merge(S_1.nextlist,\ N.nextlist); \\
& \ \ \boxed{S.nextlist}\ =\ merge(temp,\ S_2.nextlist);\ \}
\end{aligned}
$$

$$
6)\ M \rightarrow \epsilon \qquad \{\ M.instr\ =\ nextinstr;\ \}
$$

$$
\begin{aligned}
7)\ N \rightarrow \epsilon \qquad & \{\ N.nextlist\ =\ makelist(nextinstr); \\
& \ \ \boxed{gen(\texttt{'goto \_'});}\ \}
\end{aligned}
$$

$$
S \rightarrow \textbf{if}\ (\ B\ )\ S_1\ \textbf{else}\ S_2 \quad \left|\
\begin{aligned}
B.true &= newlabel() \\
B.false &= newlabel() \\
S_1.next &= S_2.next = \boxed{S.next} \\
S.code &= B.code \\
& \ \ \|\ label(\boxed{B.true})\ \|\ S_1.code \\
& \ \ \|\ gen(\texttt{'goto'}\ \boxed{S.next}) \\
& \ \ \|\ label(\boxed{B.false})\ \|\ S_2.code
\end{aligned}\right.
$$

3) $S \to \textbf{while } M_1 \; (\; B \;) \; M_2 \; S_1$

$$
\begin{aligned}
\{ \; & \boxed{\begin{aligned}&\mathit{backpatch}\\&\mathit{backpatch}\end{aligned}}\begin{aligned}&(S_1.\mathit{nextlist}, \; M_1.\mathit{instr});\\&(B.\mathit{truelist}, \; M_2.\mathit{instr});\end{aligned}\\
& \boxed{S.\mathit{nextlist}} = B.\mathit{falselist};\\
& \mathit{gen}(\boxed{\text{'goto'} \; M_1.\mathit{instr});} \; \}
\end{aligned}
$$

6) $M \to \epsilon \qquad \{ \; M.\mathit{instr} = \mathit{nextinstr}; \; \}$

$S \to \textbf{while} \; (\; B \;) \; S_1$

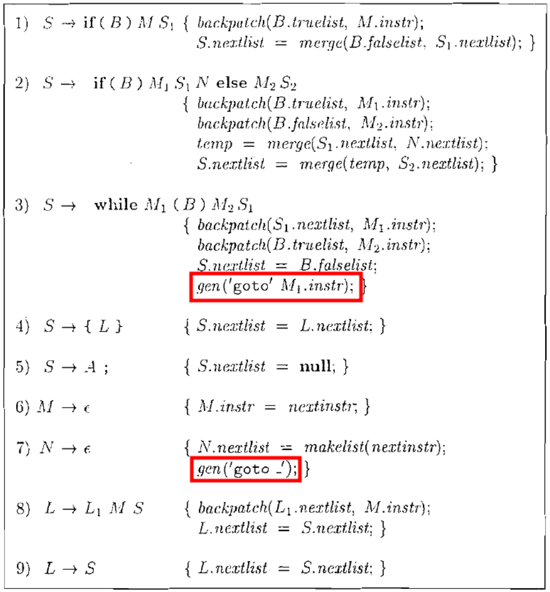
$$
\begin{aligned}
\mathit{begin} &= \mathit{newlabel}()\\
B.\mathit{true} &= \mathit{newlabel}()\\
B.\mathit{false} &= \boxed{S.\mathit{next}}\\
S_1.\mathit{next} &= \mathit{begin}\\
S.\mathit{code} &= \mathit{label}(\boxed{\mathit{begin}}) \; || \; B.\mathit{code}\\
&\quad || \; \mathit{label}(\boxed{B.\mathit{true}}) \; || \; S_1.\mathit{code}\\
&\quad || \; \mathit{gen}(\boxed{\text{'goto'} \; \mathit{begin}})
\end{aligned}
$$

$$8)\quad L \to L_1\ M\ S \qquad \{\ \boxed{backpatch}(L_1.nextlist,\ M.instr);\ \boxed{L.nextlist} = S.nextlist;\ \}$$

$$9)\quad L \to S \qquad \{\ \boxed{L.nextlist} = S.nextlist;\ \}$$

4) $S \rightarrow \{\ L\ \}$ $\qquad \{\ \boxed{S.nextlist} = L.nextlist;\ \}$

5) $S \rightarrow A\ ;$ $\qquad \{\ \boxed{S.nextlist} = \textbf{null};\ \}$



1) $S \rightarrow \textbf{if}\ (\ B\ )\ M\ S_1$ $\{\ backpatch(B.truelist,\ M.instr);$
$S.nextlist\ =\ merge(B.falselist,\ S_1.nextlist);\ \}$

2) $S \rightarrow \textbf{if}\ (\ B\ )\ M_1\ S_1\ N\ \textbf{else}\ M_2\ S_2$
$\{\ backpatch(B.truelist,\ M_1.instr);$
$backpatch(B.falselist,\ M_2.instr);$
$temp\ =\ merge(S_1.nextlist,\ N.nextlist);$
$S.nextlist\ =\ merge(temp,\ S_2.nextlist);\ \}$

3) $S \rightarrow \textbf{while}\ M_1\ (\ B\ )\ M_2\ S_1$
$\{\ backpatch(S_1.nextlist,\ M_1.instr);$
$backpatch(B.truelist,\ M_2.instr);$
$S.nextlist\ =\ B.falselist;$
$\boxed{gen(\texttt{'goto'}\ M_1.instr);}\ \}$

4) $S \rightarrow \{\ L\ \}$ $\{\ S.nextlist\ =\ L.nextlist;\ \}$

5) $S \rightarrow A\ ;$ $\{\ S.nextlist\ =\ \textbf{null};\ \}$

6) $M \rightarrow \epsilon$ $\{\ M.instr\ =\ nextinstr;\ \}$

7) $N \rightarrow \epsilon$ $\{\ N.nextlist\ =\ makelist(nextinstr);$
$\boxed{gen(\texttt{'goto \_'});}\ \}$

8) $L \rightarrow L_1\ M\ S$ $\{\ backpatch(L_1.nextlist,\ M.instr);$
$L.nextlist\ =\ S.nextlist;\ \}$

9) $L \rightarrow S$ $\{\ L.nextlist\ =\ S.nextlist;\ \}$

只有 (3) 与 (7) 生成了新的代码, 控制流语句的主要目的是“控制” 流。

```
procedure AreYouOK(score)
    if score $\geq$ 60 then
        while true do
            print "WanSui"
    else
        print "Sad"
```

2) $S \rightarrow$ **if** (B) $M_1\ S_1\ N$ **else** $M_2\ S_2$
{ $backpatch(B.truelist,\ M_1.instr);$
$backpatch(B.falselist,\ M_2.instr);$
$temp\ =\ merge(S_1.nextlist,\ N.nextlist);$
$S.nextlist\ =\ merge(temp,\ S_2.nextlist);$ }

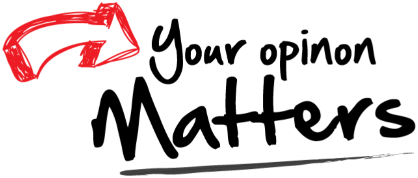
3) $S \rightarrow$ **while** M_1 (B) $M_2\ S_1$
{ $backpatch(S_1.nextlist,\ M_1.instr);$
$backpatch(B.truelist,\ M_2.instr);$
$S.nextlist\ =\ B.falselist;$
$gen($'`goto`' $M_1.instr);$ }

6) $M \rightarrow \epsilon$ { $M.instr\ =\ nextinstr;$ }

7) $N \rightarrow \epsilon$ { $N.nextlist\ =\ makelist(nextinstr);$
$gen($'`goto _`'$);$ }

6) $B \rightarrow$ **true** { $B.truelist = makelist(nextinstr);$
$gen($'`goto _`'$);$ }

Thank You!



Office 926

hfwei@nju.edu.cn