Chapter7

Semantic Analysis
&
Intermediate Code Generation

Example : A + B * (C - D) + E / (C - D) ^N

X := a*b + c/d − (e+f)/g

(1) c* a b)
(2) c/ c d)
(3) c+ (1) (2)
(4) c+ e f)
(5) c/ (4) g)
(6) c− (3) (5)

逆波兰式：
ab* cd/+ef+g/−:=

(1)	(−	C	D)
(2)	(*	B	(1))
(3)	(+	A	(2))
(4)	(−	C	D)
(5)	(^	(4)	N)
(6)	(/	E	(5))
(7)	(+	(3)	(6))

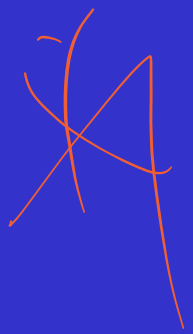
Quad Code

- Two forms
  - z:= x op y
  - (op, x, y, z)
- Other quad codes
  - z := x op y
  - z := op x
  - z := x
  - if x relop y goto –

Example:

a: =b*c+b*d

- Form1
  - （1） （*， b， c， t_1）
  - （2） （*， b， d， t_2）
  - （3） （+， t_1， t_2， t_3）
  - （4） （:=， t_3， —， a）
- Form2
  - （1） t_1:=b*c
  - （2） t_2:=b*d
  - （3） t_3:=t_1+t_2
  - （4） a:=t_3



X:= a*b + c/d - (e+f)/g

(1) (*, a, b, t_1)
(2) (/, c, d, t_2)
(3) (+, t_1, t_2, t_3)
(4) (+, e, f, t_4)
(5) (/, t_4, g, t_5)
(6) (-, t_3, t_5, t_6)
(7) (:=, t_6, -, X)

Assignment statement translation

- Assignment statement
  - a:=b*(c+d)
- Quad code
  - t1:=c+d
  - t2:=b*t1
  - a:=t2

1.Numerical representation

1 represent true，0 represent false

Boolean expression：

a or b and not c

Quad code：

t1:=not c
t2:=b and t1
t3:=a or t2

布尔表达式（优先级）
非 与 或

Relational expression：

a<b

Equivalent to

if a<b then t:=1 else t:=0

Quad code:

100： if a＜b goto 102 真出口

101： goto105 假出口

102： t:=1

103： goto 105

104： t:=0

105：

2. Boolean expression in control statement

E.true represents true exit, **E.false** represents false exit.

- **Boolean expression： a<b**

  Quad code:

  100 if a<b goto **E.true**

  101 goto **E.false**

- **How to translate a<b or c>d ?**

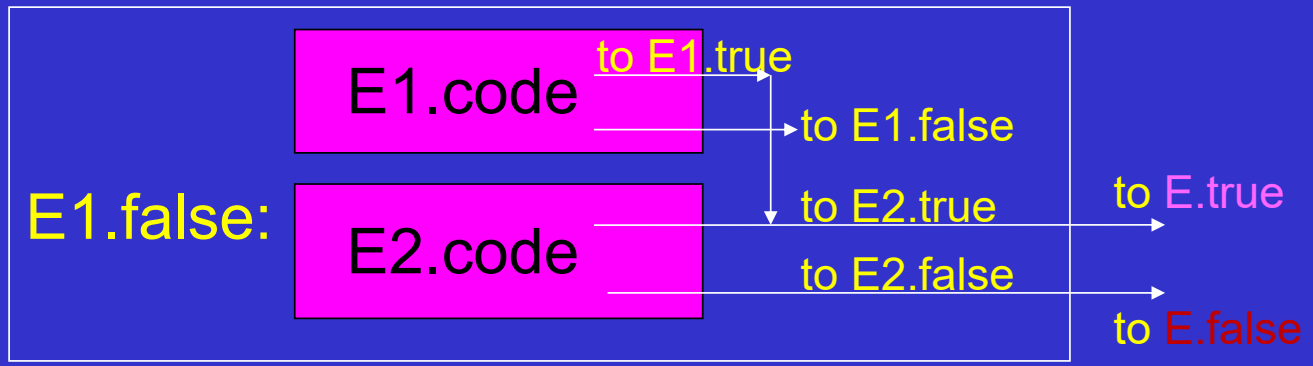
X := a or b and not C
逆波兰：
X a b C not and or :=

四元式：
(1) (not, —, C, t_1)
(2) (and, b, t_1, t_2)
(3) (or, a, t_2, t_3)
(4) (:=, t_3, —, X)

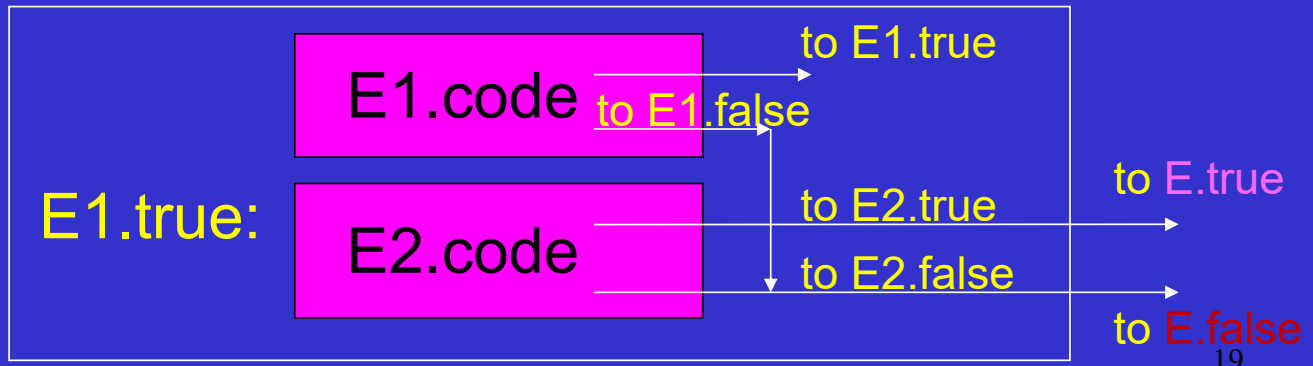
或：t_1 := not C
t_2 := b and t_1
t_3 := a or t_2
X := t_3

~~not X a b and or :=~~

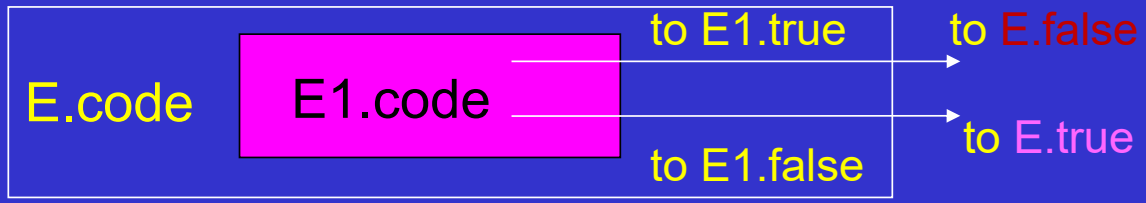
（1）E→E1 or E2 （a<b or c>d）



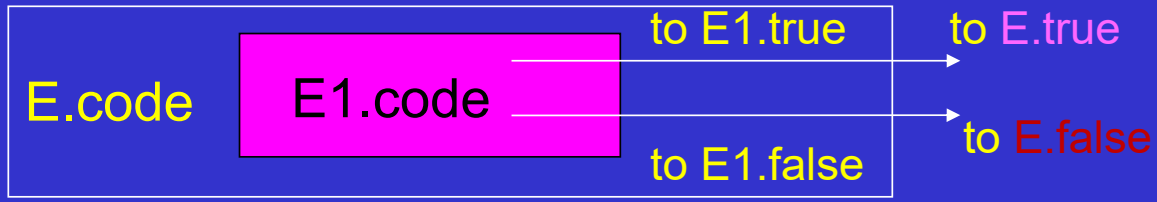
（2） E→ E1 and E2 （a<b and c>d）



（3） E→ not E1 （not a<b）



（4） E→(E1) （ （a<b））

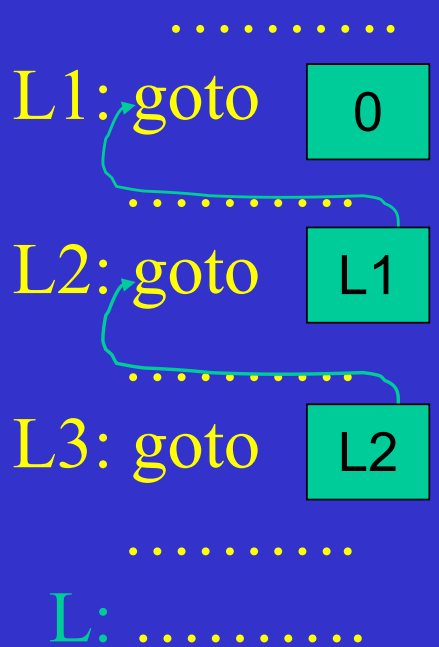


（5） E→ id1 relop id2 （a<b）

（6） E→ true

（7） E→ false

Merge:



a<b / E1 and c>d / E2 or not e<f / E3

100: if a<b goto E2.B (102)
101: goto E3.B (104)
102: if c>d goto E.True
103: goto 104
104: if e<f goto E.False
105: goto E.True

Backfill:

When the turning point is determined, backfill along the chain.

a<b or c<d and e>f

E1 or E2 and E3

Before Merge：

100 : if a＜b goto E.true
101 : goto E2.begin
102 : if c＜d goto E3.begin
103 : goto E.false
104 : if e>f goto E.true
105 : goto E.false

After Merge：

100 : if a＜b goto 0
101 : goto l02
102 : if c＜d goto l04
103 : goto 0
104 : if e>f goto 100
105 : goto 103

Head of chain E.True is 104

Head of chain E.false is 105

How to translate?

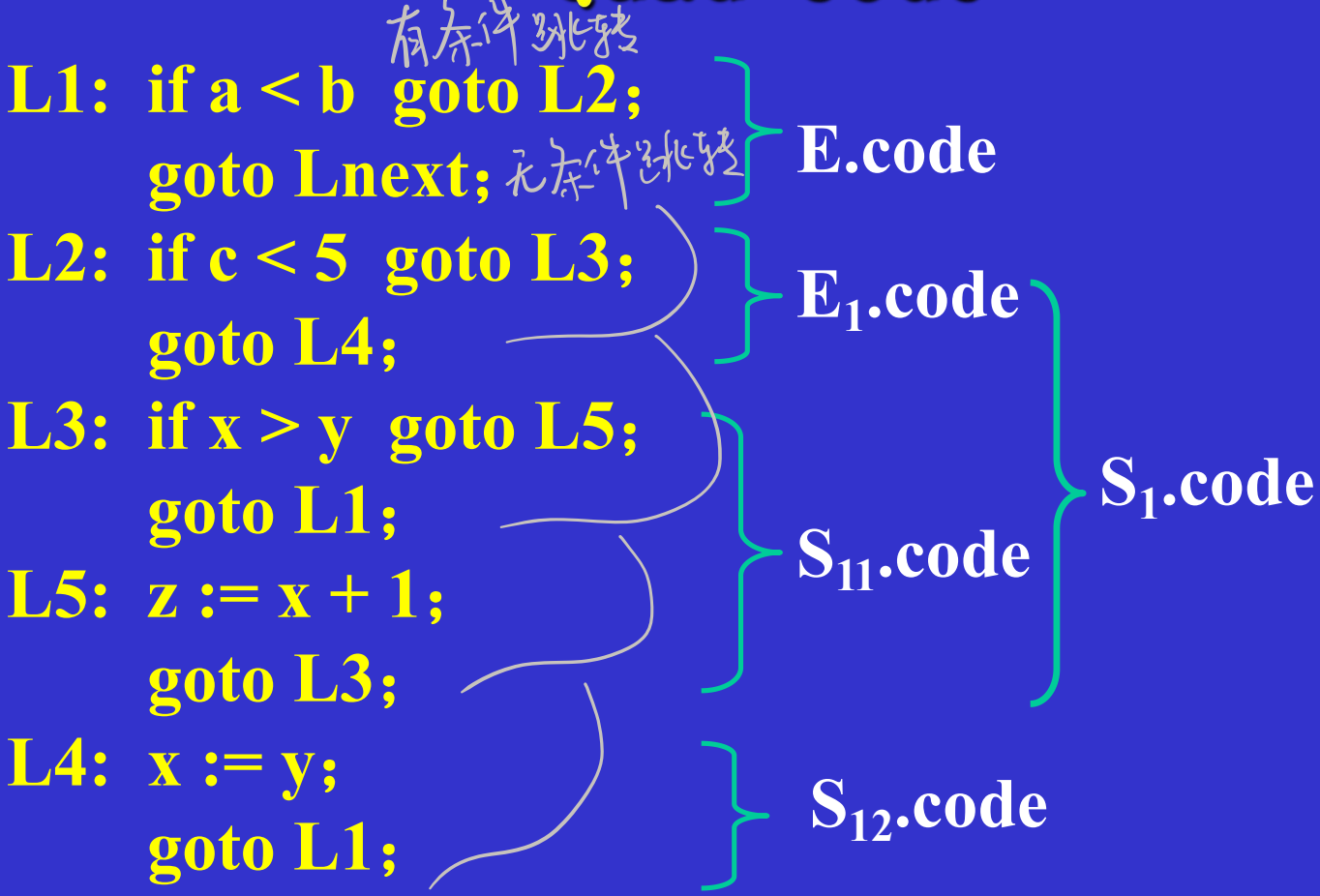
```
while a < b do
    if c < 5 then
        while x > y do z := x + 1;
    else
        x := y
```

if a<b then 四元式:
  x := a+b 100 (j<, a, b,)
else 101 (
  x := a-b

100: if a<b goto 102
101: goto E.F
102: t1 := a+b
103: x := t1
104: goto 107

105: t2 := a-b
106: x := t2
107:

Quad code


有条件跳转
无条件跳转
L1: if a < b goto L2;
goto Lnext;
L2: if c < 5 goto L3;
goto L4;
L3: if x > y goto L5;
goto L1;
L5: z := x + 1;
goto L3;
L4: x := y;
goto L1;
Lnext:
E.code
E1.code
S11.code
S1.code
S12.code

Array translation

- Array A[I]
- Quad code

  t1:=VARPART;

  t2:=CONSPART;

  t2[t1]

  - t1 is relative address

X:=a[I] :

T1:=VARPART;

T2:=CONSPART;

T3:=T2[T1];

X:=T3;

P:=P+A[I]*B[I]

1. T1:=4*I
2. T2:=addr(A)
3. T3:=T2[T1]
4. T4:=4*I
5. T5:=addr(B)
6. T6:=T5[T4]
7. T7:=T3*T6
8. P:=P+T7

- Each element occupies W bytes
- addr(A) and addr(B)
  - base address of array
- low
  - lower bound，
- address of A[I]

  addr(A)+(I-low)*W

  =addr(A)+(I-0)*4

  =addr(A)+4*I
- CONSTPART
  - addr(A)
- VARPART
  - 4*I