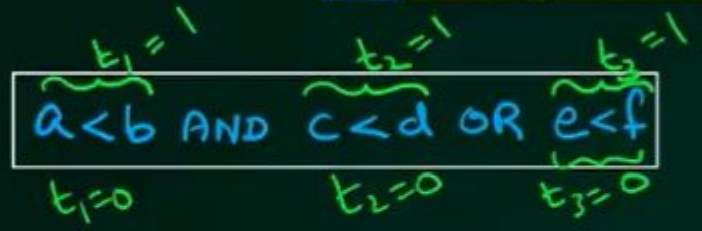
[1] Backpatching and Conversion to 3-Address Code

```
if (a<b) then t = 1
else t = 0
```

3 ADDRESS CODE

```
(i): if a<b goto i+3
(i+1): t=0
(i+2): goto i+4
(i+3): t=1
(i+4): return
```

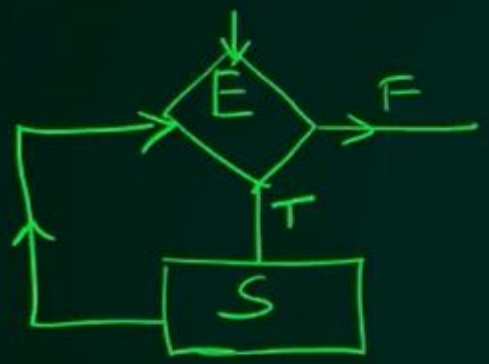
[2] Backpatching and Conversion to 3-Address Code




```
100: if (a<b) goto 103
101: t1 = 0
102: goto 104
103: t1 = 1
104: if(c<d) goto 107
105: t2 = 0
106: goto 108
107: t2 = 1
108: if (e<f) goto 111
109: t3 = 0
110: goto 112
111: t3 = 1
112: t4 = t1 AND t2
113: t5 = t4 OR t3
```

[3] Backpatching and Conversion to 3-Address Code

while E do S




```
L: if(E==0) goto L1
   S
   goto L
L1:
```

OR

```
L: if(E) goto L1
   goto LAST
L1: S
   goto L
LAST:
```

[4] Backpatching and Conversion to 3-Address Code

```
while (a<b) do
  x = y + z
```

```
L:  if a<b goto L1
    goto LAST

L1: t = y + z
    x = t
    goto L

LAST:
```

[5] Backpatching and Conversion to 3-Address Code

```
for(i=0; i<10; i++)
    a = b + c
```

```
    i = 0
L: if(i<10) goto L1
    goto LAST
L1: t1 = b + c
    a = t1
    t = i + 1
    i = t
    goto L
LAST:
```

[6] Backpatching and Conversion to 3-Address Code

```
switch (i+j)
{
  case (1): a = b+c
            break
  case (2): p = q+r
            break
  default : x = y+z
            break
}
```

```
    t = i+j
    if (t==1) goto L1
    if (t==2) goto L2
    goto L3
L1: t1 = b+c
    a = t1
    goto LAST
L2: t2 = q+r
    p = t2
    goto LAST
L3: t3 = y+z
    x = t3
LAST:
```