CT-2 106118036

POCD - Compilers

Date
Page

5) We know 'a' is loaded already in R0.

So, we use the following table:

Line No.	Operation	Assembly Code	Cost
101	x := a + x	ADD x, R0	2
	Now R0 contains x = a+x		
102	y = b*2	MOV b, R1	2
		MUL 2, R1	2
	Now, y in R1 (y = b*2)		
103	a = x - y	SUB R1, R0	1
		MOV R0, a	2
	So, finally, we have a in R0.		9

Total cost = 9 Ans

17

Grammar:-

$S' \rightarrow S$

$S \rightarrow CB$

$S \rightarrow BC$

$C \rightarrow Cad$

$C \rightarrow d$

$B \rightarrow BaC$

$B \rightarrow a$

LR(1) Table:-

goto	State	Closure
	0	{[S' → .S, $] [S → .CB, $] [S → .BC, $], [C → .Cad, a] [C → .d, a] [B → .BaC, d/a] [B → .a, d/a]}
goto(0,S)	1	{[S' → S., $]}
goto(0,C)	2	{[S → C.B, $], [C → C.ad, a], [B → .BaC, $/a] [B → .a, $/a]}
goto(0,B)	3	{[S → B.C, $], [B → B.aC, d/a], [C → .Cad, $/a] [C → .d, $/a]}
goto(0,d)	4	{[C → d., a]}
goto(0,a)	5	{[B → a., d/a]}

goto(2,S) 6 {[S→cB.,$], [B→B.ac$/a]}

goto(3,a) 7 {[C→ca.d,a], [B→a.c,$/a]}

goto(3,C) 8 {[S→BC.,$], [C→C.ad,$/a]}

goto(3,d) 10 {[C→d.,$/a]}

goto(6,c) 11 {[B→Bac.,$/a]
[C→c.ad,$/a], [C→.d/$/a]}

goto(7,d) 12 {[C→cad.,a]}

goto(8,a) 13 {C→ca.d,$/a]}

goto(9,c) 14 {[B→Bac.,d/a]
, [C→c.ad,d/a]}

goto(9,d) 15 {[C→d.,d/a]}

goto(11,c) 16 {[B→Bac.,$/a]
, [C→C.ad,$/a]}

goto(11,d) 16

goto(13,d) 17 {[C→Cad.,$/a]}

goto(14,a) 18 {[C→Ca.d,d/a]}

goto(16,a) 13

goto(18,d) 19 {[C→Cad.,d/a]}

Now, firsts ⇒ $S' \to \{b, a\}$ $C \to \{d\}$

$S \to \{b, a\}$ $B \to \{a\}$

LR Table

State	Action			GOTO			
	a	d	$	S'	S	C	B
				1	2	3	
0	S5	S4					
1			acc				
2	S7						6
3	S9	S10				8	
4	r_4						
5	r_6	r_6					
6	S11		r_1				
7	r_6	S12	r_6				
8	S13		r_2				
9		S15	r_4			14	
10	r_4		r_4				
11		S10				16	
12	r_3						
13		S17					
14	S_8/r_5	r_5					
15	r_4	r_4					
16	S_{13}/r_5		r_5				
17	r_3		r_3				
18		S19					
19	r_3	r_3					

2) 3 address code for the given C-code can be written as:

```
i := 0                                  ---- leader

L1 : if i < n goto S1                   ---- leader
     goto L2                            ---- leader
S1 : t1 := 4 * i                        ---- leader
     t2 := a[t1]
     t3 := t2 / 2
     t3 := t3 % 10
     t3 := t3 * 4
     t1 := count
     t2 := t1[t3]
     t2 := t2 + 1
     t1[t3] := t2
     i := i + 1
     goto L1

L2 : i = 1                              ---- leader
L3 : if i < 10 goto S2                  ---- leader
     goto L4                            ---- leader
S2 : t1 := i - 1                        ---- leader
     t1 := t1 * 4
     t2 = count
     t3 := t2 [t1]
     t6 = 4 * 4
     t4 = t2[t1]
     t4 = t4 + t3
     t2 [t1] = t4
     i := i + 1
```

goto L3

L4: i := n — — — — leader
i := i-1

L5: if i>=0 goto S3 — — — — leader
goto L6 — — — — leader

S3: t1 := 4*i — — — — leader
t2 := addr
t3 := t2[t1]
t3 := t3 / emp
t3 := t3 % 10
t1 := count
t2 := t1[t3]
t2 := t2 - 1
t2 := t2 * 4
t1 := i * 4
t3 := addr
t4 := t3[t1]
t1 := output
t1[t2] := t4
t1 := 4*1
t2 := addr
t3 := t2[t1]
t3 := t3 / emp
t3 := t3 % 10
t1 := count
t2 := t1[t3]
t2 := t2 - 1
t1[t3] := t2
i := i - 1
goto L5

L6: i := 0 ——→ leader

L7: if i<n goto S4 ---- leader

goto L8 - - - - leader

S4: t1 := i*4 - - - - - leader

t2 := output

t3 := t2[t1]

t2 := arr

t2[t1] := t3

i := i+1

goto L7

L8:

Now, we have 16 Blocks

3) $x := b[i] + c \hat{} d * t - 2$

TAC can be written as:-

$t_1 = i$
$t_2 = t_1 * 4$
$t_3 = b[t_2]$
$t_4 = d * t$
$t_5 = t_3 + c$
$t_6 = t_4 - 2$
$t_7 = t_5 \text{ xor } t_6$
$x = t_7$

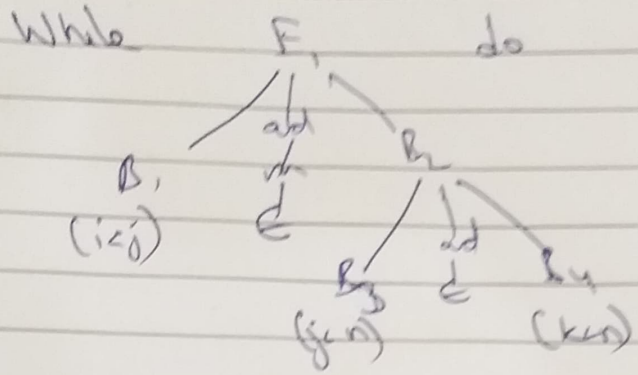
Triples Representation:-

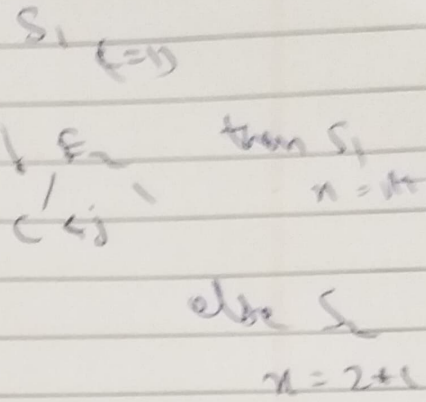
#	Op	Arg1	Arg2
(0)	Assign	i	
(1)	*	4	(0)
(2)	[]	b	(1)
(3)	*	d	t
(4)	+	c	(2)
(5)	-	(3)	2
(6)	xor	(4)	(5)
(7)	assign	x	(6)

4) TAC without backpatching

		Filling goto
1:	if i<j goto ___	3
2:	goto ___	21
3:	if j<n goto ___	7
4:	goto ___	5
5:	if k<n goto ___	7
6:	goto ___	21
7:	c = 1	
8:	goto ___	9
9:	if c<j goto ___	11
10:	goto ___	13
11:	x = x+c	
12:	goto ___	15
13:	x = 2*c	
14:	goto ___	15
15:	c = c+1	
16:	goto ___	17
17:	j = j+1	
18:	goto ___	19
19:	k = k*2	
20:	goto ___	1
21:	a = x+n	

Annotated Tree:





S_1 a = x+n

S_2 c = c+1

S_3 f = j+1

S_4 k = k+1

Using these, we can fill the gst