BT18CSE021
USHAK
SHARMA

Date. _____
Page No. _____

LANGUAGE PROCESSORS

ASSIGNMENT - 4 - II

LIVE VARIABLE ANALYSIS
AND AVALABLE VARIABLE ANALYSIS

PROGRAM :

```
int A, B, C;
int main() {
    int a, b, c, d;
    b = 8;
    a = b + c;
    d = a * b;
    if (A<B) {
        b = a - c;
    } else {
        do {
            c = b + c;
            if (B>A) {
                do {
                    d = a+b;
                    func1( b * c);
                } while (B>A)
            } else {
                c = a*b;
                func1( a-b);
            }
            func2 (a+b);
        } while (C>A);
}
```

func3(a-c);
func1(b+c);

CONTROL FLOW GRAPH

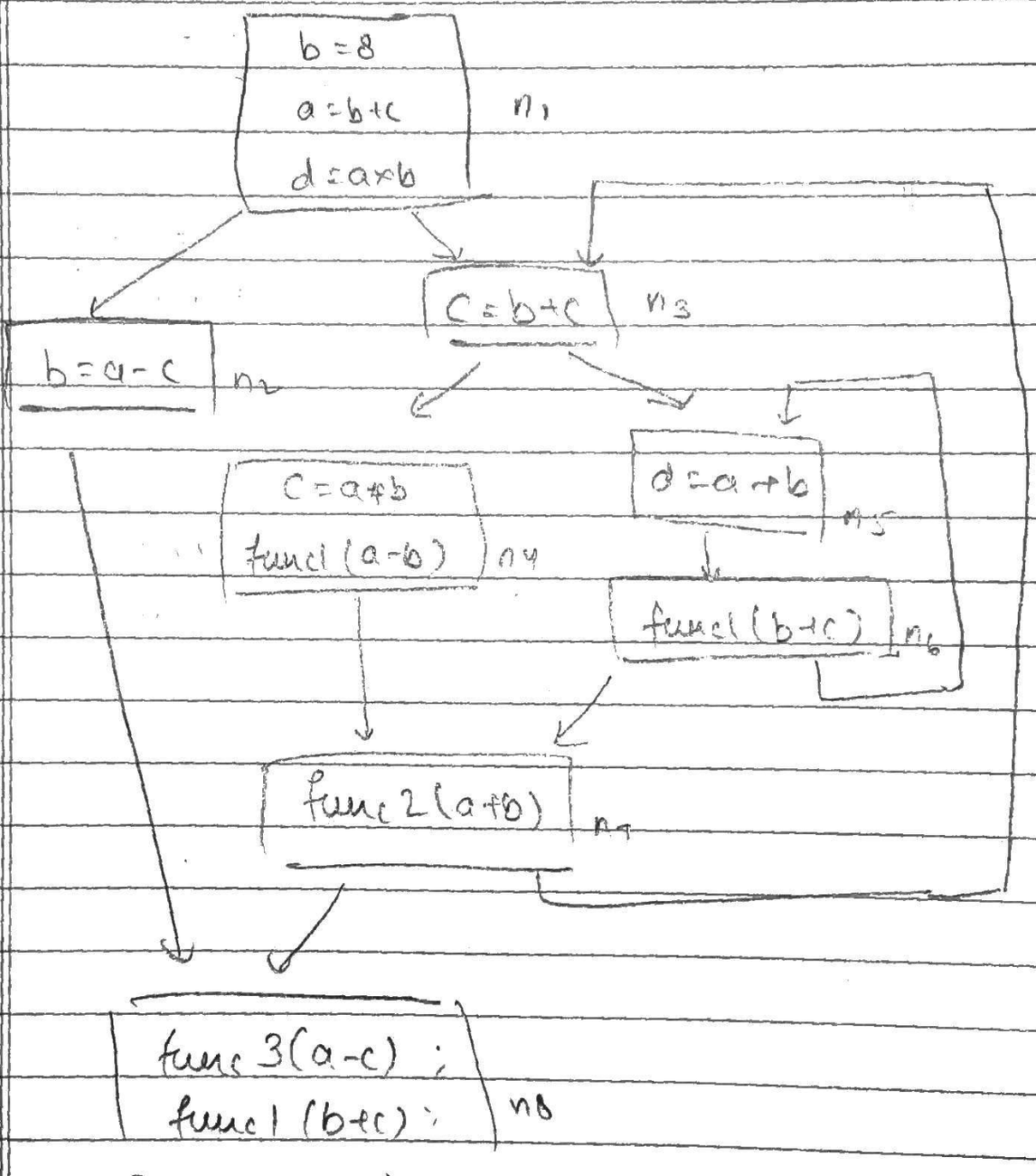


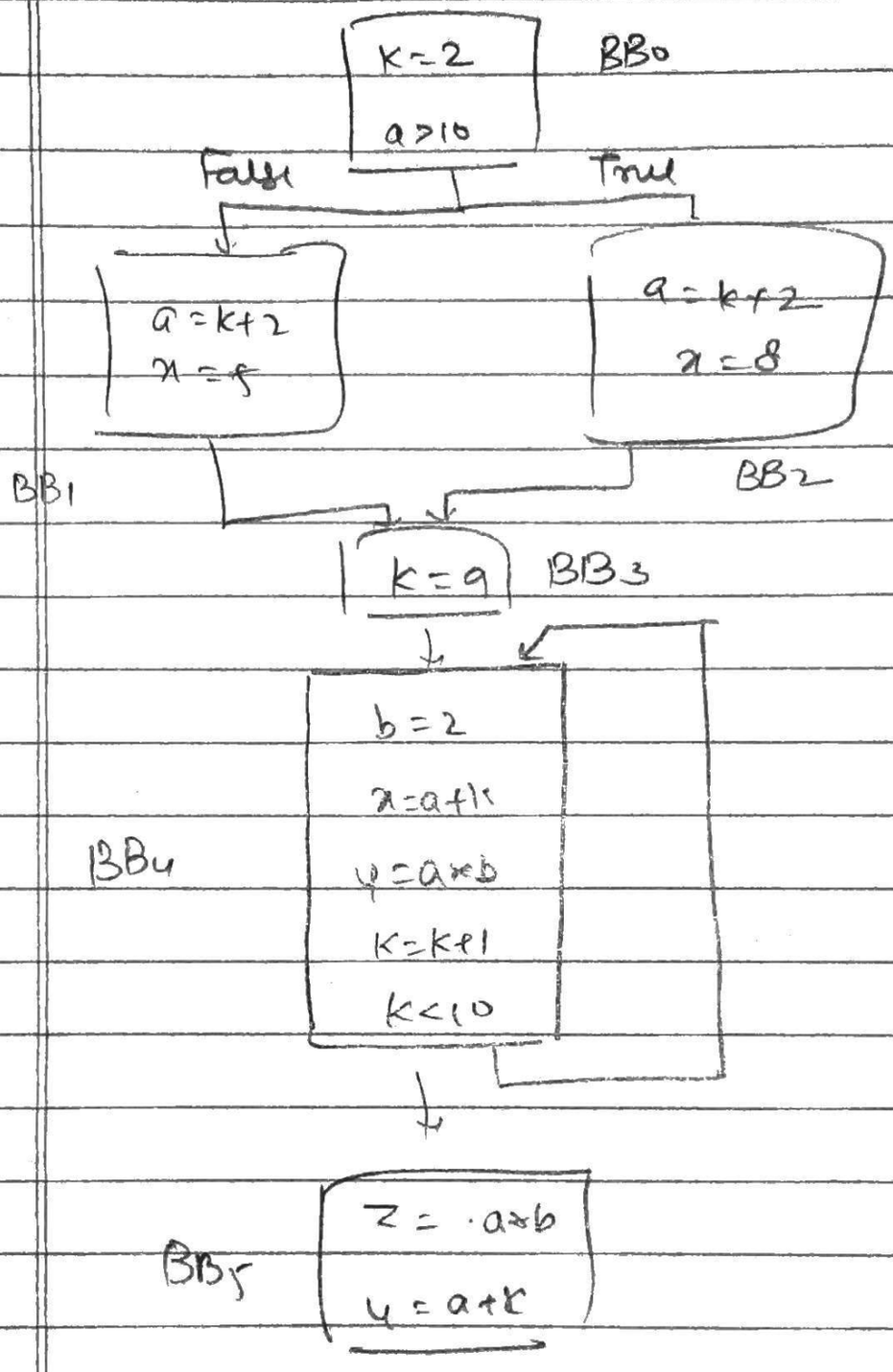
TABLE FOR LIVE VARIABLE ANALYSIS

BLOCK	GEN	KILL	1st Iteration		2nd Iteration	
			OUT	IN	OUT	IN
n8	{a,b,c}	ϕ	ϕ	{a,b,c}	ϕ	{a,b,c}
n7	{a,b}	ϕ	{a,b,c}	{a,b,c}	{a,b,c}	{a,b,c}
n6	{b,c}	ϕ	{a,b,c}	{a,b,c}	{a,b,c}	{a,b,c}
n5	{a,b}	{d}	{a,b,c}	{a,b,c}	{a,b,c}	{a,b,c}
n4	{a,b}	{c}	{a,b,c}	{a,b}	{a,b,c}	{a,b}
n3	{b,c}	{c}	{a,b,c}	{a,b,c}	{a,b,c}	{a,b,c}
n2	{a,c}	{b}	{a,b,c}	{a,c}	{a,b,c}	{a,c}
n1	{c}	{a,b,d}	{a,b,c}	{c}	{a,b,c}	{c}

ANOTHER EXAMPLE

		LEADERS
1.	K=2	Yes
2.	if a>10 goto 6	
3.	a=K+2	Yes
4.	x=5	
5.	goto 8	
6.	a=k*2	Yes
7.	n=8	
8.	K=a	Yes
9.	b=2	Yes
10.	x=a-k	
11.	y=a+b	
12.	c=b+1	
13.	if k<10 goto 9	
14.	z=a-b	Yes.
15	y=a+k.	

CONTROL FLOW GRAPH



LIVE VARIABLE ANALYSIS

	SUCCESOR	GEN	KILL	1st Iteration IN	1st Iteration OUT	2nd Iteration IN	2nd Iteration OUT
BB0	1, 2	a	k	ϕ	ϕ	a	ϕ
BB1	3	k	a, x	ϕ	ϕ	k	ϕ
BB2	3	k	a, x	ϕ	ϕ	k	ϕ
BB3	4	a,	k	ϕ	ϕ	a	ϕ
BB4	4, 5	a, k	b, a, 4, k	ϕ	ϕ	a, x	ϕ
BB5	ϕ	a, b, k	z, y	ϕ	ϕ	a, b, k	ϕ

	3rd Iteration IN	3rd Iteration OUT	4th Iteration IN	4th Iteration OUT	5th Iteration IN	5th Iteration OUT
BB0	a	k	a	k	a	k
BB1	k	a	k	a	k	a
BB2	k	a	k	a	k	a
BB3	a	a, k	a	a, k	a	a, k,
BB4	a, k	a, b, k	a, b	a, b, k	a, k	a, b, k
BB5	a, b, k	ϕ	a, b, k	ϕ	a, b, k	ϕ

Date. _____

Page No. _____

AVAILABLE EXPRESSION ANALYSIS (7)

Transfer Equations

GEN(B) = { the exp x op y with no sub. def of x | y in B }

KILL(B) = { rest other than Gen(B) } → union

Pred(B) = predecessors

OUT(B) = (IN(B) − KILL(B)) ∪ GEN(B)

IN(B) = ∪ OUT(P) where P ∈ Pred(B)

S1 = (k+2, k+2, a+k, a*b, k+1)

1st iteration

	PRED	GEN	KILL	IN	OUT
BB0	ϕ	ϕ	k+2, k+2, a+k, k+1	ϕ	ϕ
BB1	BB0	k+1	a+k, a*b	k+2, k+2, a+k, a*b, k+1	a
BB2	BB0	k<2	a+k, a*b	S1	ϕ
BB3	BB1, BB2	ϕ	k+2, k+2, a+k, k+1	S1	ϕ
BB4	BB3, BB4	a+k, a*b, k+1	(a+b) ∪ (KILL BB0)	S1	ϕ
BB5	BB4	a*b, a+k	ϕ	S1	ϕ

	IN	OUT	IN	OUT	IN	OUT
BB0	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
BB1	ϕ	k+1	ϕ	k+1	ϕ	k+2
BB2	ϕ	k+1	ϕ	k+1	ϕ	k+2
BB3	k+2, k+2, k+1	ϕ	ϕ	ϕ	ϕ	ϕ
BB4	a*b	a+k, a*b, k+1	ϕ	a+k, a*b, k+1	ϕ	a+k, a*b, k+1
BB5	a+k, a*b, k+1	a+k, a*b, k+1	a+k, a*b, k<1	" "	a+k, a*b, k+1	a+k, a*b, k<1

Date. _____

Page No. _____

CONSTANT FOLDING

Example

```
void f() {
    int A[10];
    A[2] = 3*5;
}
```

After applying constant folding

⇓

```
void f() {
    int A[10];
    A[2] = 15;
}
```

```
void g() {
int A[10], a, b, *c;
char* d, *e;
a = 3*5;
d = (char*) A
b = 2*4;
e = d+b;
c = (int*) e;
*c = a;
}
```

After applying Constant folding

⇓

```
void g() {
int A[10], a, b, *c;
char* d, *e;
a = 15;
d = (char*) A
b = 8;
e = d+b;
c = (int*) e
*c = a;
}
```