

CMSC 430 Practice Problem 6 Solution

Solutions

1. Value Numbering

(a) What are in the CODE, SYMBOLS, AVAIL, and CONSTANTS tables?

CODE			
Instructions	resultValNum	isConst	
(1) $i + 1$	3	no	
(2) $a := (1)$	3	no	
(3) $a + b$	5	no	
(4) $c := (3)$	5	no	
(5) $d := 1$	2	yes	
(6) $i + d$	3	no	deleted
(7) $e := (6)$	3	no	
(8) $d + 2$	7	yes	deleted
(9) $f := (8)$	7	yes	

SYMBOLS		
name	ValNum	isConst
i	1	no
c1	2	yes
a	3	no
b	4	no
c	5	no
d	2	yes
e	3	no
c2	6	yes
f	7	yes

CONSTANTS	
ValNum	bits
2	1
6	2
7	3

AVAIL				
lhsValNum	op	rhsValNum	resultValNum	instr
1	+	2	3	(1)
3	+	4	5	(3)

(b) What is the output code?

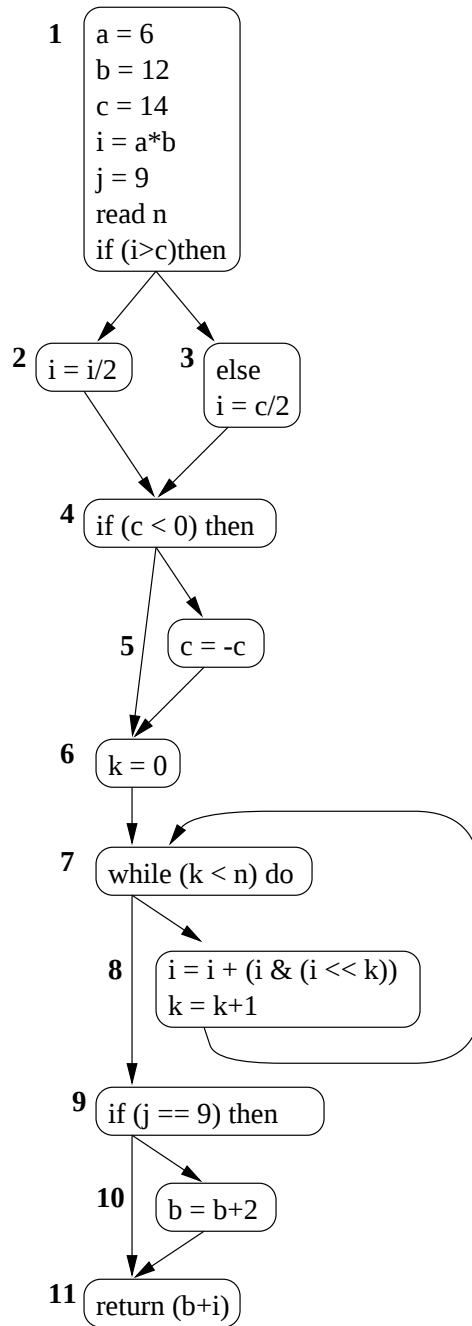
```

(1)    i + 1                //  a := i + 1
(2)    a := (1)
(3)    a + b                //  c := a + b
(4)    c := (3)
(5)    d := 1              //  d := 1
(6)    e := (1)            //  e := i + 1
(7)    f := 3              //  f := 3

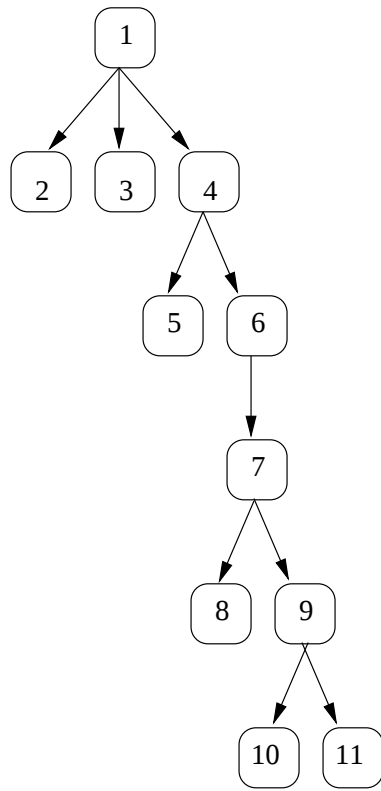
```

2. **Static Single Assignment.** Consider the following program

(a) Construct a control flow graph for the program.



(b) Construct a dominator tree for the resulting CFG.



(c) What is the iterated dominance frontier for variables `c` and `i`?

$$DF(c) = DF(1,5) = \{ 6 \}$$

$$DF(i) = DF(1,2,3,8) = \{ 4,7 \}$$

(d) Generate the SSA form for this program.

```
procedure foo
  a, b, c, i, j, k, n : integer

  a = 6
  b1 = 12
  c1 = 14
  i1 = a * b1
  j = 9
  read n

  if (i1 > c1) then
    i2 = i/2
  else
    i3 = c1/2

  i4 = phi(i2,i3)
  if (c1 < 0) then
    c2 = -c1

  c3 = phi(c1,c2)
  k1 = 0

L: i5 = phi(i4,i6)
  k2 = phi(k1,k3)
  if (k2 < n) then
    i6 = i5 + (i5 & (1 << k2))
    k3 = k2 + 1
    goto L

  if (j == 9) then
    b2 = b1 + a

  b3 = phi(b1,b2)
  return (b3 + i5)
end foo
```

(e) Show the result of conditional constant propagation.

```
procedure foo
  a, b, c, i, j, k, n : integer

  a = 6
  b = 12
  c = 14
  i = 72          // a * b
  j = 9
  read n

  if (true) then   // 72 > 14
    i = 36         // 72 / 2
  else
    i = c/2

  if (false) then  // 14 < 0
    c = -c

  k = 0
  while (k < n) do
    i = i + (i & (1 << k))
    k = k + 1
  end while

  if (true) then   // 9 == 9
    b = 18         // 12 + 6

  return (18 + i)  // b + i
end foo
```