CSEP501 : Compiler Construction: Homework 4

Karuna Sagar Krishna

November 27, 2023

Question 1

1a

The following table (hash map) captures the value number mapping for different expressions encountered in the given basic block. To construct this we look at each statement in order, for each expression get or assign a value number. If the expression can be broken down, we first sort the operands if the operation is commutative, calculate the value number for sub expressions and look up the hash table for key ($op\ o1\ o2$) to get or assign a value number.

$$
\begin{aligned}
a &\rightarrow 1 \\
b &\rightarrow 2 \\
(+1\ 2) &\rightarrow 3 \\
t_1 &\rightarrow 3 \\
c &\rightarrow 4 \\
(+3\ 4) &\rightarrow 5 \\
t_2 &\rightarrow 5 \\
d &\rightarrow 6 \\
(+5\ 6) &\rightarrow 7 \\
t_3 &\rightarrow 7 \\
t_4 &\rightarrow 3 \\
e &\rightarrow 8 \\
(+7\ 8) &\rightarrow 9 \\
t_5 &\rightarrow 9 \\
f &\rightarrow 10 \\
(+3\ 10) &\rightarrow 11 \\
t_6 &\rightarrow 11 \\
t_7 &\rightarrow 3
\end{aligned}
$$

Now with the above table, we rewrite the statements with value number in super script.

$$
\begin{aligned}
t_1^3 &= a^1 + b^2 \\
t_2^5 &= t_1^3 + c^4 \\
t_3^7 &= t_2^5 + d^6 \\
t_4^3 &= b^2 + a^1 \\
t_5^9 &= t_3^7 + e^8 \\
t_6^{11} &= t_4^3 + f^{10} \\
t_7^3 &= a^1 + b^2
\end{aligned}
$$

1b

Redundant expressions $a + b$ can be eliminated as shown below. Since addition is commutative t_4 get the same value number as (+1 2). This is because we sort the operands for commutative operations before looking up the hash table. Similarly, t_7 gets the same value number as (+1 2).

$$
\begin{aligned}
t_1^3 &= a^1 + b^2 \\
t_2^5 &= t_1^3 + c^4 \\
t_3^7 &= t_2^5 + d^6 \\
t_4^3 &= t_1^3 \\
t_5^9 &= t_3^7 + e^8 \\
t_6^{11} &= t_4^3 + f^{10} \\
t_7^3 &= t_1^3
\end{aligned}
$$

Question 2

2a

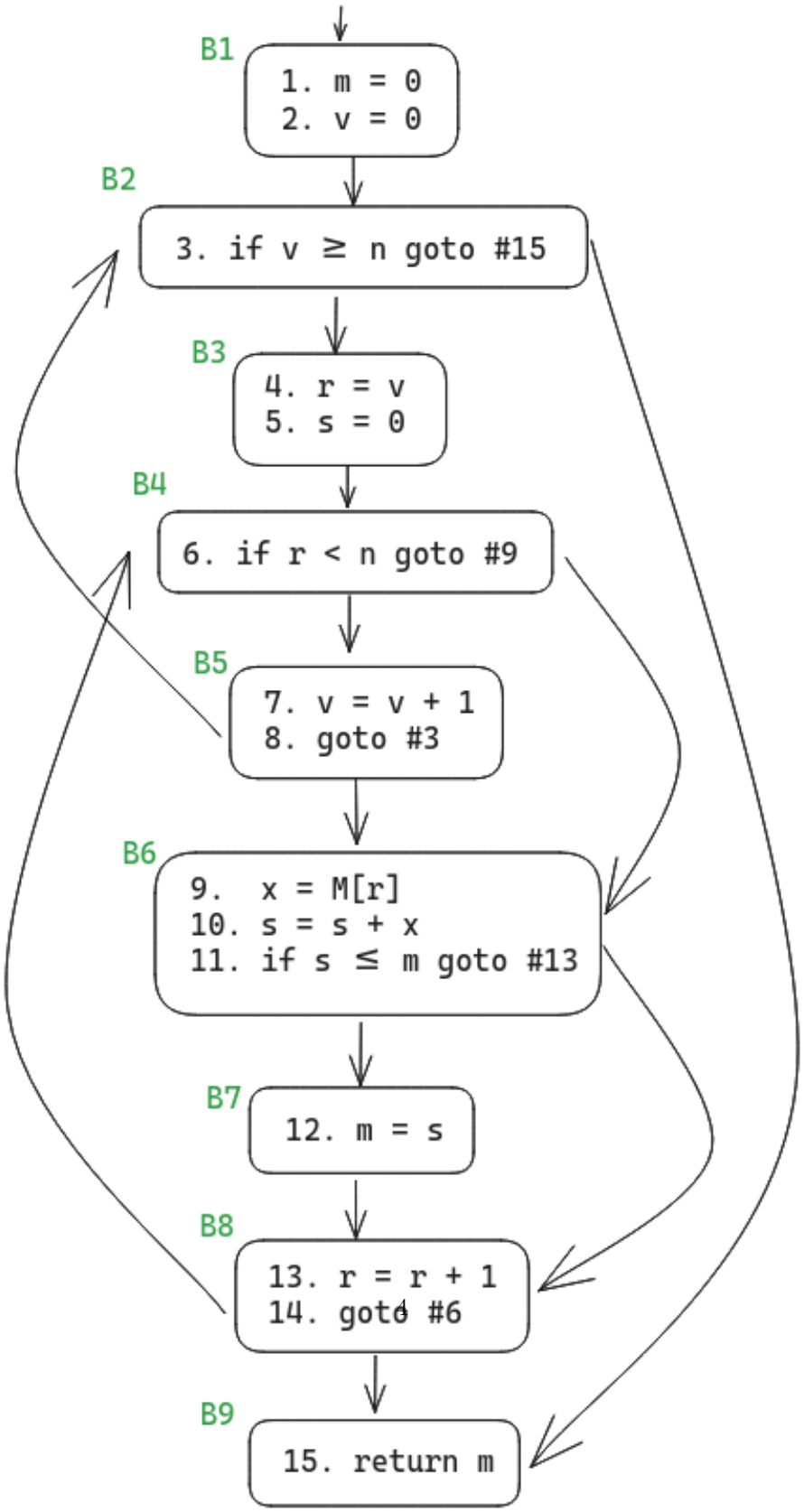
A leader instruction is identified by the following conditions:

1. First instruction of the program
2. Instruction that is the target of a jump
3. Instruction that immediately follows a jump

Below we identify the leader instructions in the given basic block. Note, $leader_i$ represents the instruction is leader due to condition i as stated above.

1	$m = 0$	$leader_1$
2	$v = 0$	
3	if $v >= n$ goto #15	$leader_2$
4	$r = v$	$leader_3$
5	$s = 0$	
6	if $r < n$ goto #9	$leader_2$
7	$v = v + 1$	$leader_3$
8	goto #3	
9	$x = M[r]$	$leader_2(leader_3)$
10	$s = s + x$	
11	if $s <= m$ goto #13	
12	$m = s$	$leader_3$
13	$r = r + 1$	$leader_2$
14	goto #6	
15	return m	$leader_2(leader_3)$

2b


B1
1. m = 0
2. v = 0
B2
3. if v ≥ n goto #15
B3
4. r = v
5. s = 0
B4
6. if r < n goto #9
B5
7. v = v + 1
8. goto #3
B6
9. x = M[r]
10. s = s + x
11. if s ≤ m goto #13
B7
12. m = s
B8
13. r = r + 1
14. goto #6
B9
15. return m

Question 3

3a

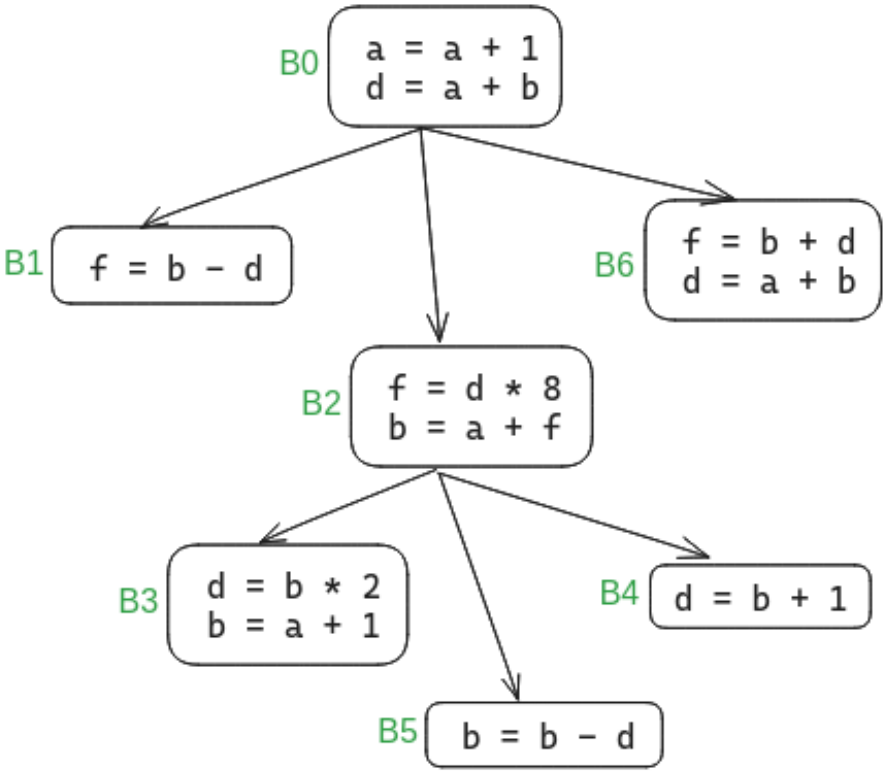
block	use	def	out	in
9	m			m
8	r	r	r,n,s,x,m,v	r,n,s,x,m,v
7	s	m	r,n,s,x,m,v	r,n,s,x,v
6	r,s,x,m	x,s	r,n,s,x,m,v	r,n,s,x,m,v
5	v	v	r,s,x,m,v,n	r,s,x,m,v,n
4	r,n		r,s,x,m,v,n	r,s,x,m,v,n
3	v	r,s	r,n,s,x,m,v	v,n,x,m
2	v,n		v,n,x,m	v,n,x,m
1		m,v	v,n,x,m	n,x

3b

Variables n and x are uninitialized because we see that these 2 variables are live on entry to block 1 which is at the beginning of the program. This means that these variables are used before they are defined or initialized.

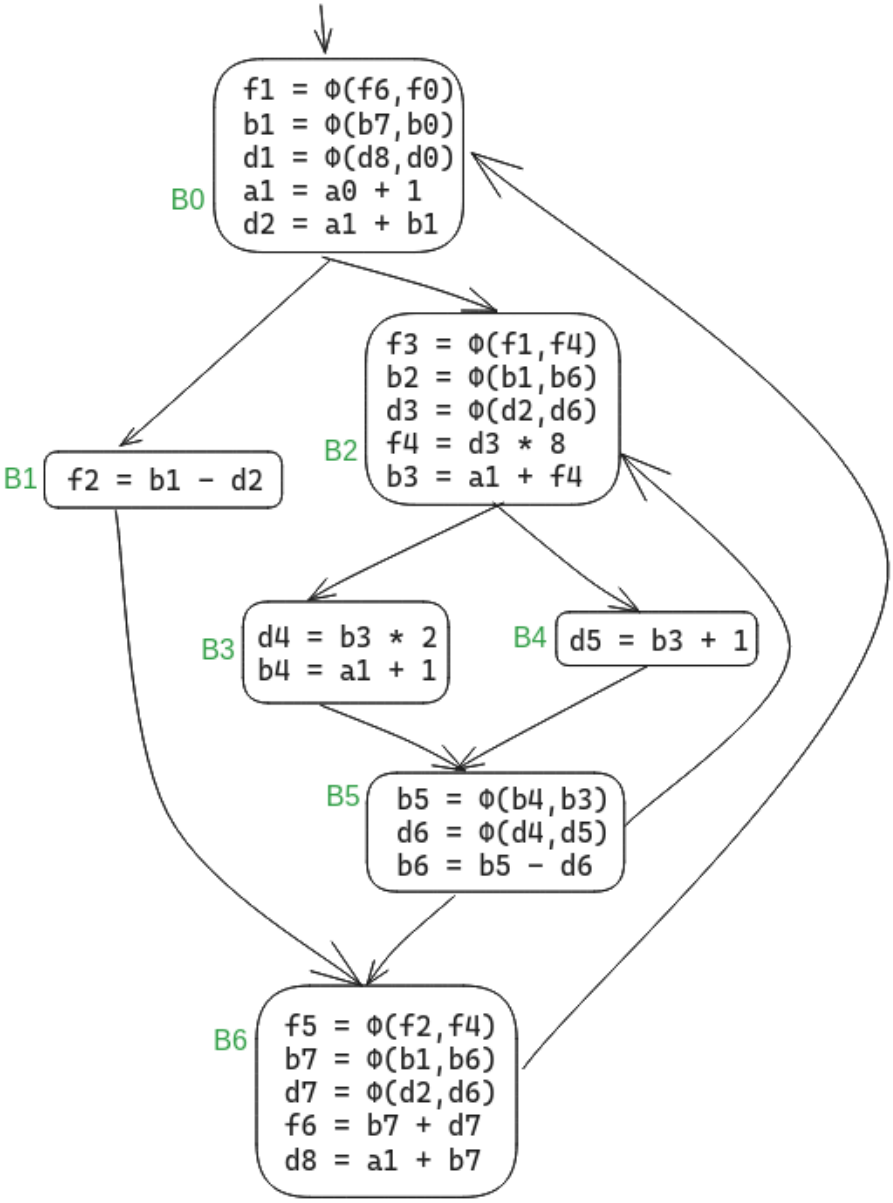
Question 4

block	dom	idom
0	0	
1	1,0	0
2	2,0	0
3	3,2,0	2
4	4,2,0	2
5	5,2,0	2
6	6,0	0



block	dominance frontier	strictly dominates
0		1,2,3,4,5,6
1	6	
2	2,6	3,4,5
3	5	
4	5	
5	2,6	
6	0	

Question 5


B0
f1 = Φ(f6,f0)
b1 = Φ(b7,b0)
d1 = Φ(d8,d0)
a1 = a0 + 1
d2 = a1 + b1
B1
f2 = b1 - d2
B2
f3 = Φ(f1,f4)
b2 = Φ(b1,b6)
d3 = Φ(d2,d6)
f4 = d3 * 8
b3 = a1 + f4
B3
d4 = b3 * 2
b4 = a1 + 1
B4
d5 = b3 + 1
B5
b5 = Φ(b4,b3)
d6 = Φ(d4,d5)
b6 = b5 - d6
B6
f5 = Φ(f2,f4)
b7 = Φ(b1,b6)
d7 = Φ(d2,d6)
f6 = b7 + d7
d8 = a1 + b7