

Register Allocation

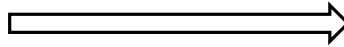
TEACHING ASSISTANT: DAVID TRABISH

Control Flow Graph

- Create a node for each IR instruction
- Create an edge between an instruction and its next instruction
- If the instruction is a **branch**:
 - Connect it to the instruction that comes after the **target label**

Control Flow Graph

```
a = x * (y - z)
if (a) {
    a = a + 1;
}
b = a
```



```
t1 = x
t2 = y
t3 = z
t4 = sub t2, t3
t5 = mult, t1, t4
a = t5
t6 = a
bne t6, 1, end
t7 = a
t8 = 1
t9 = add t7, t8
a = t9
end:
t10 = a
b = t10
```

`t1 = x`

`t2 = y`

`t3 = z`

`t4 = sub t2, t3`

`t5 = mult t1, t4`

`a = t5`

`t6 = a`

`bne t6, 1, end`

`t7 = a`

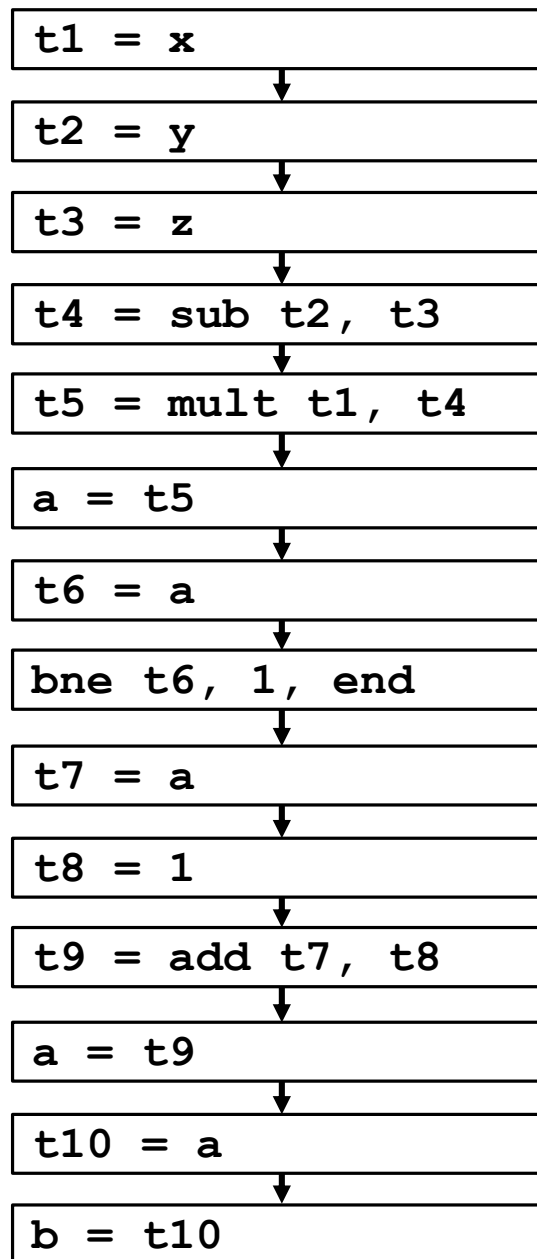
`t8 = 1`

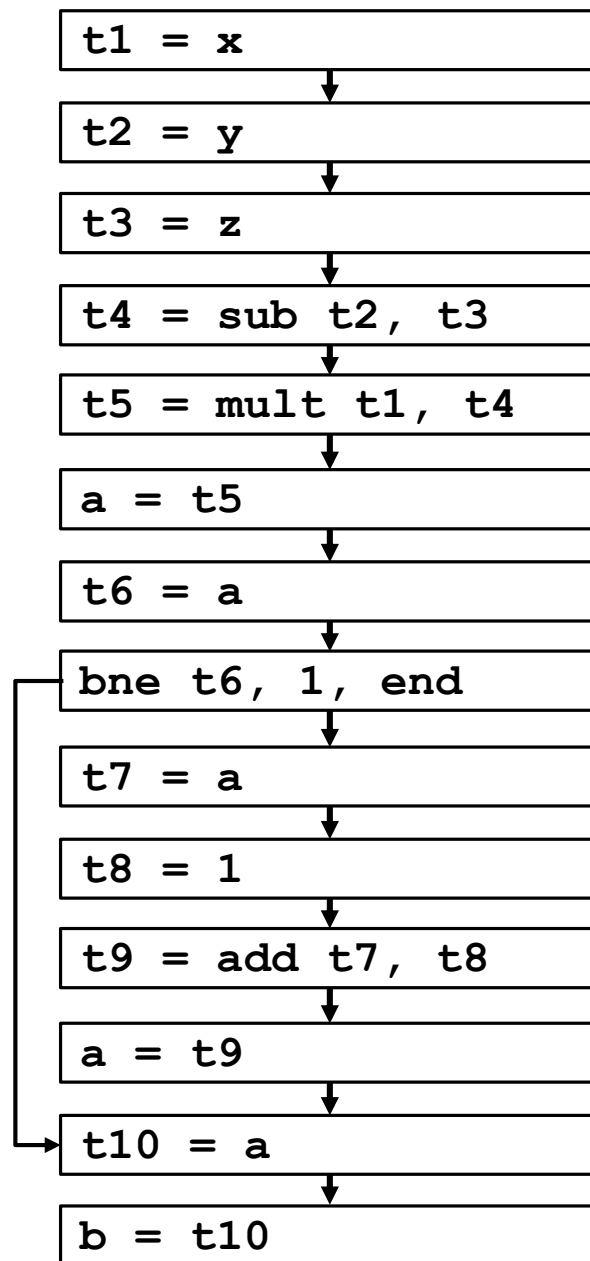
`t9 = add t7, t8`

`a = t9`

`t10 = a`

`b = t10`





Liveness Analysis

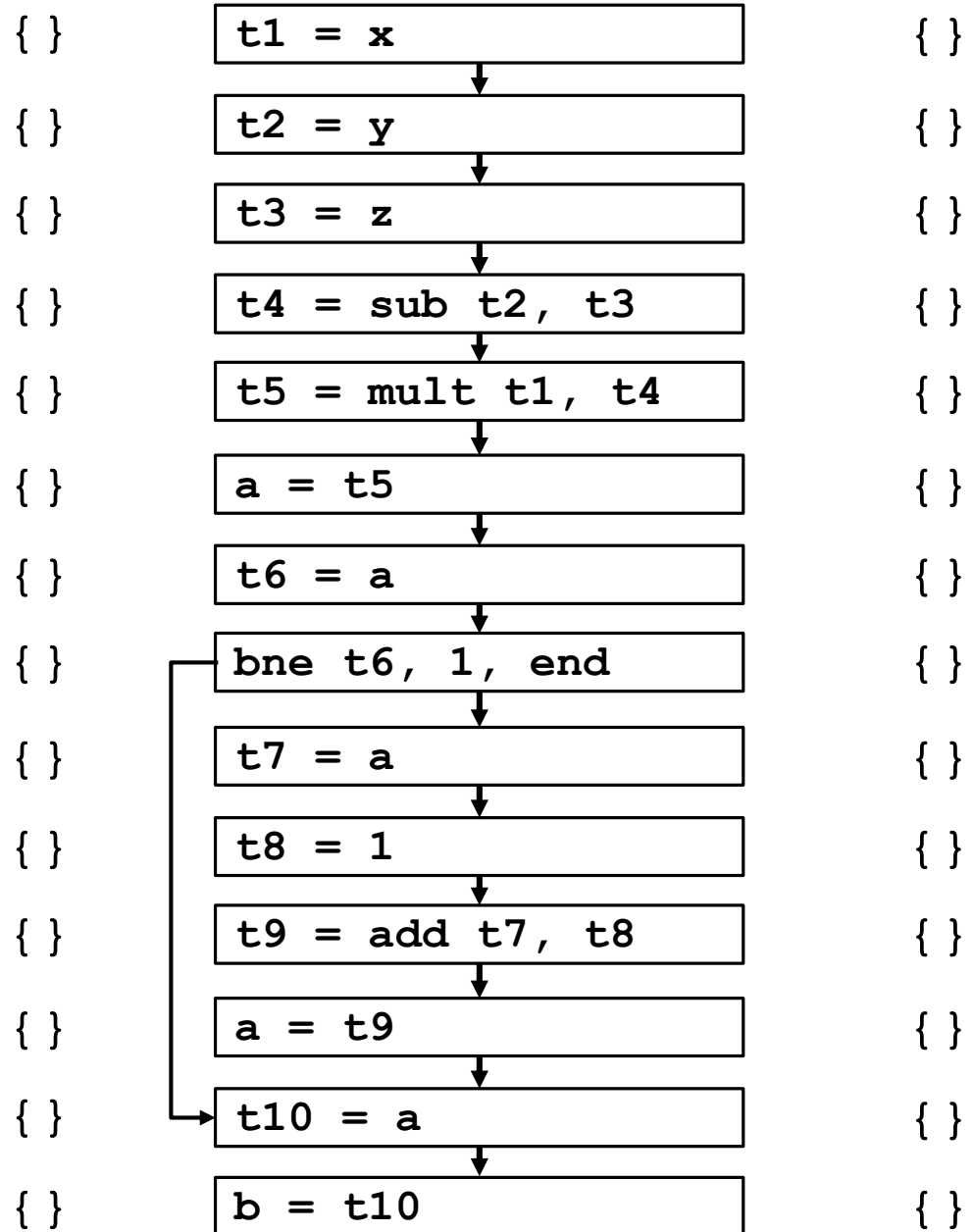
- TODO

Liveness Analysis

First Iteration...

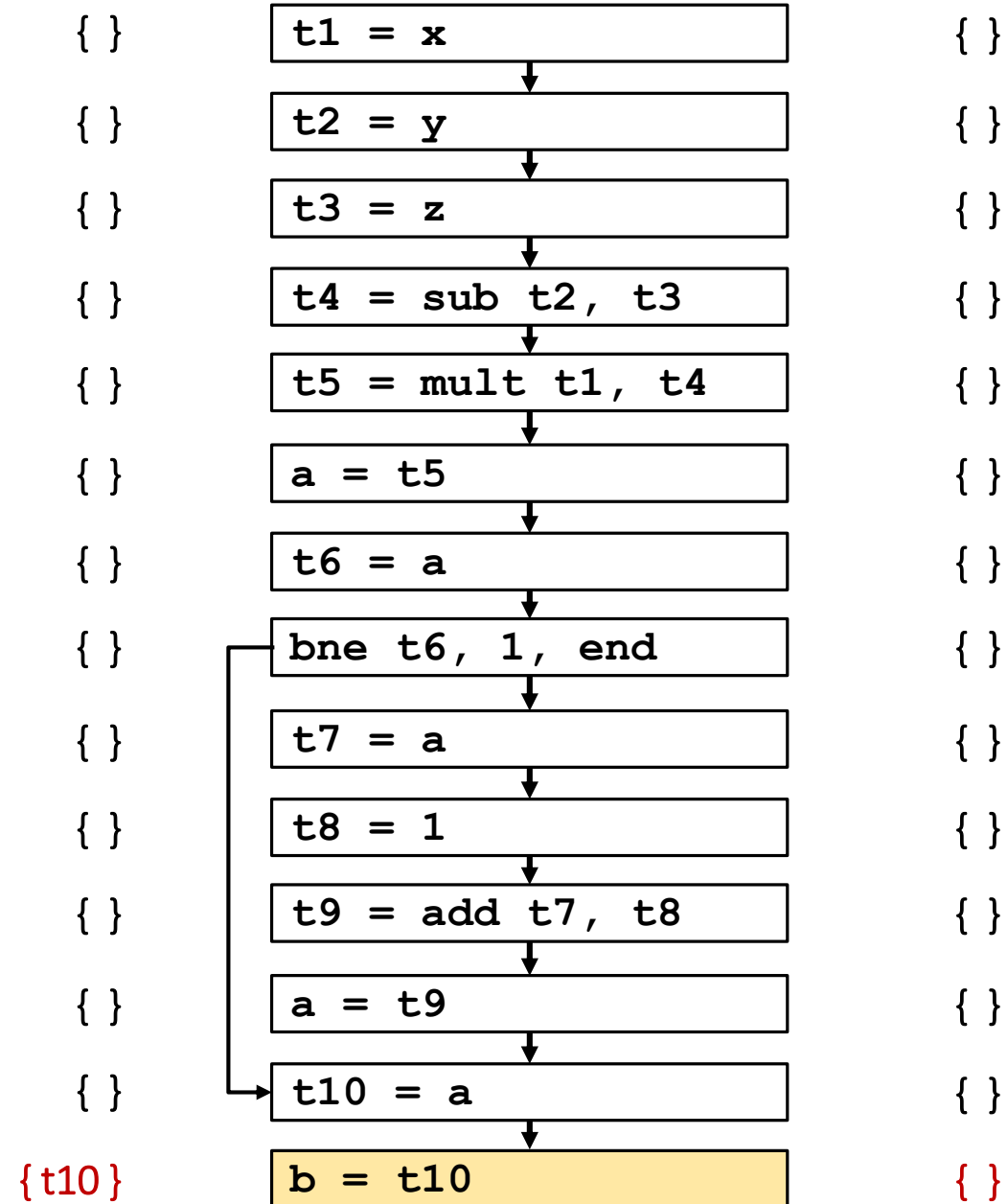
IN

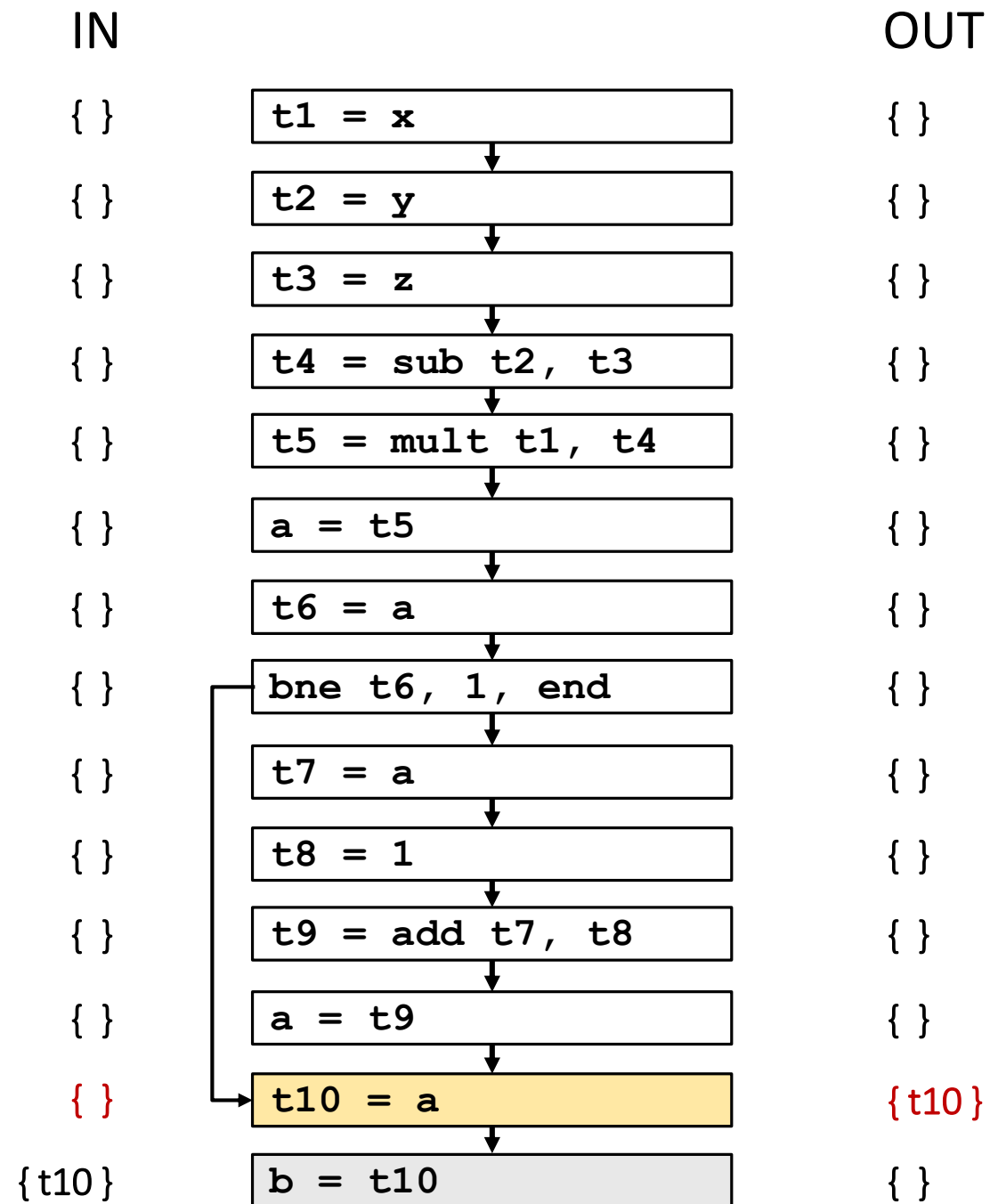
OUT

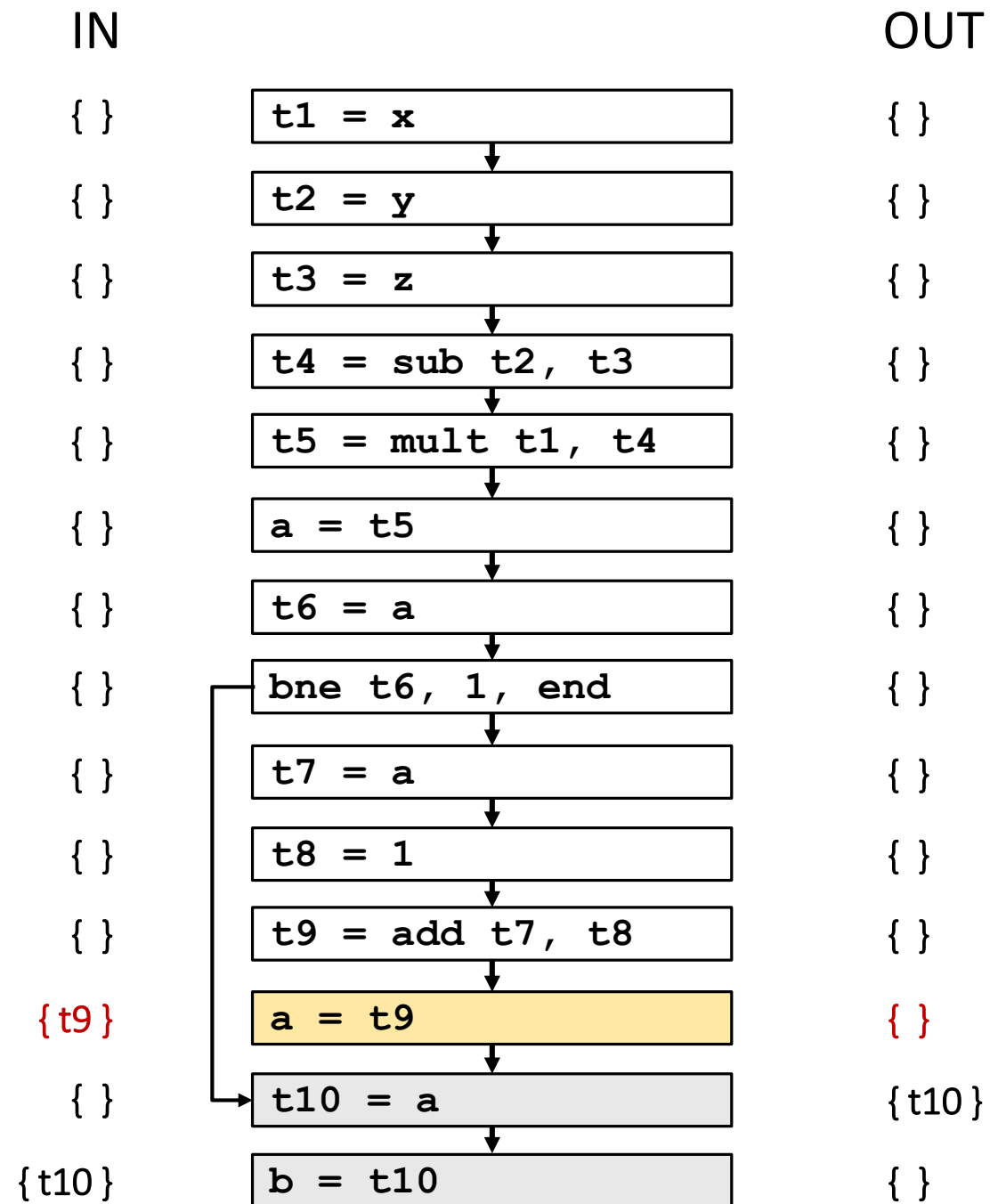


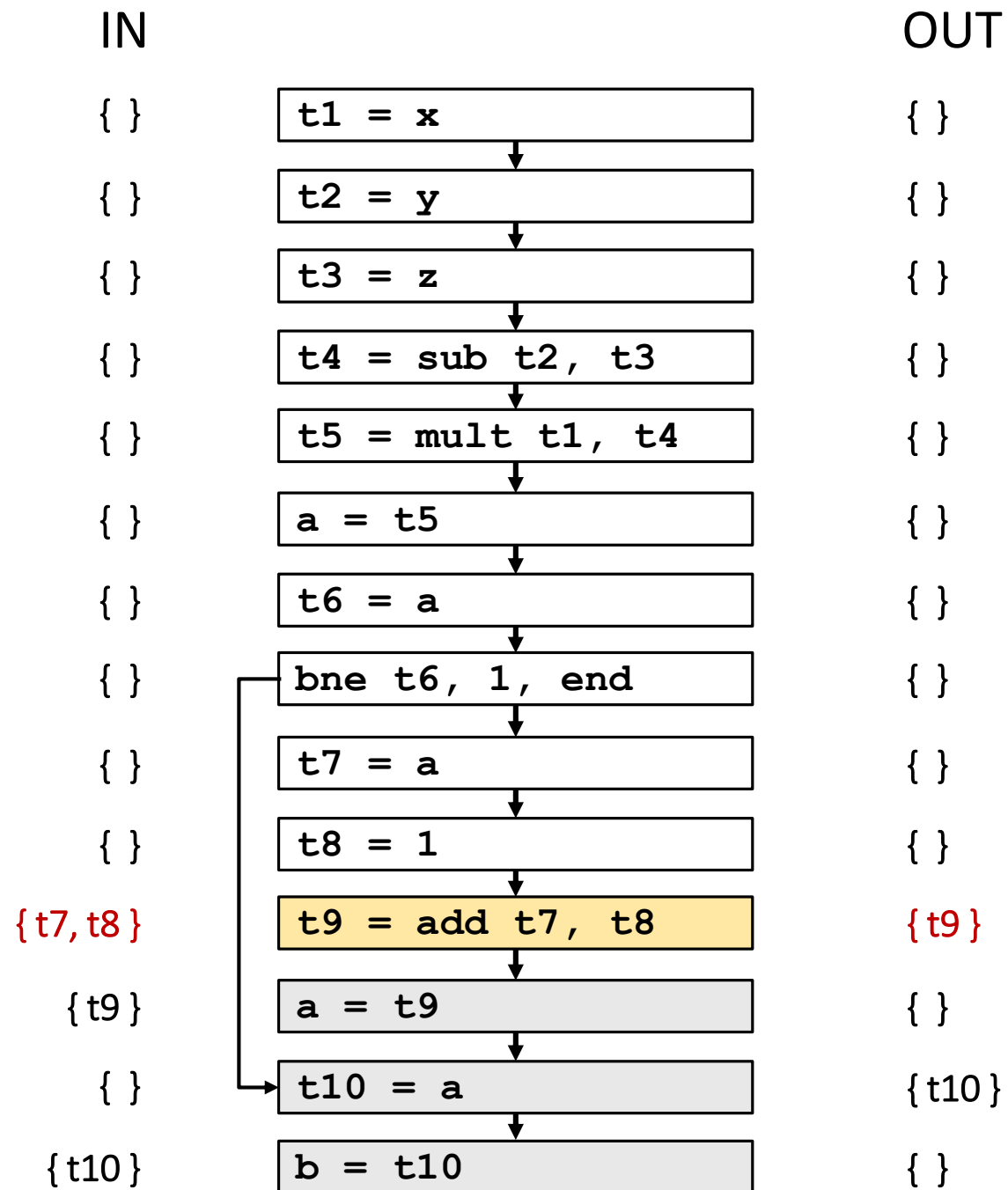
IN

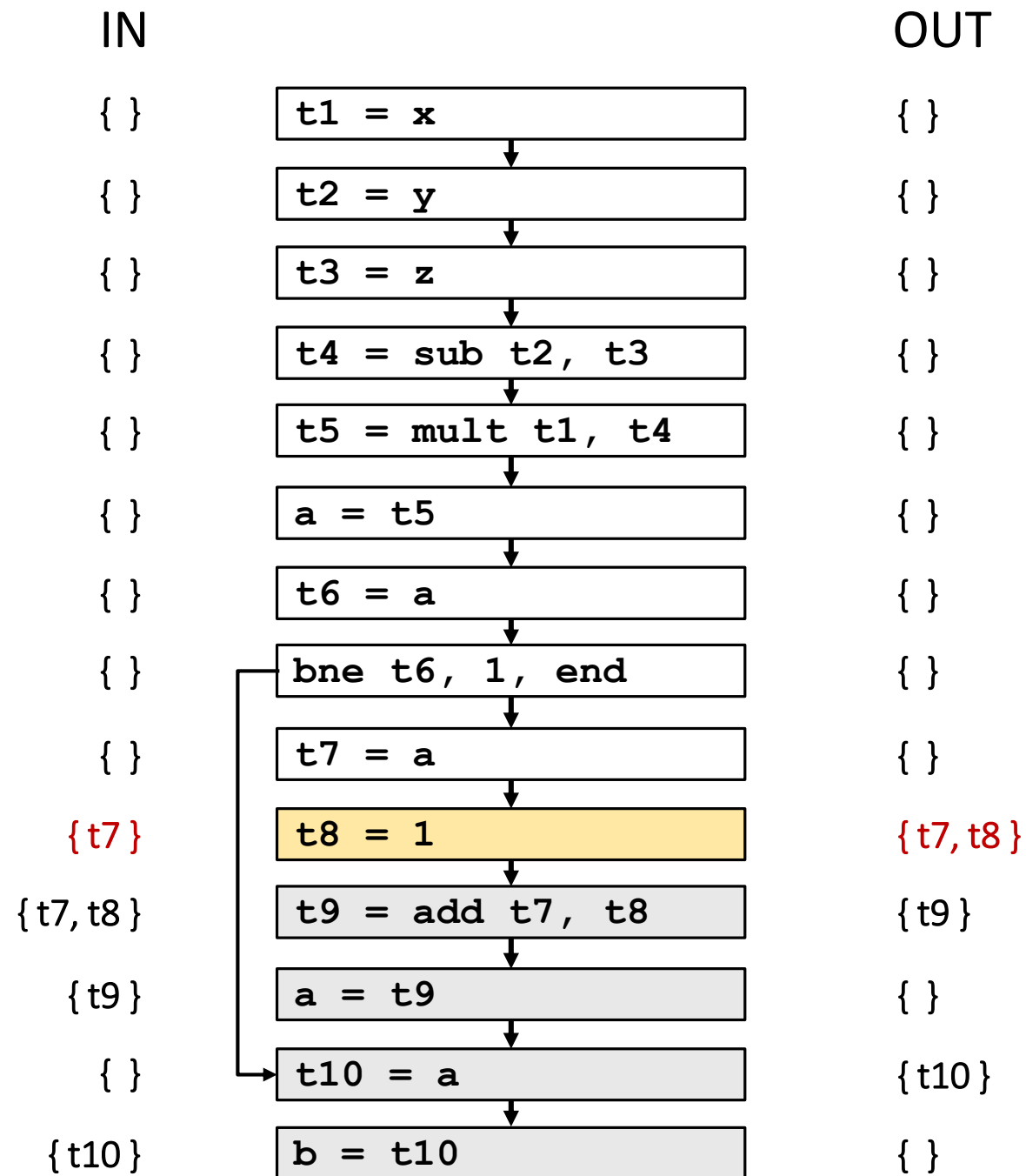
OUT

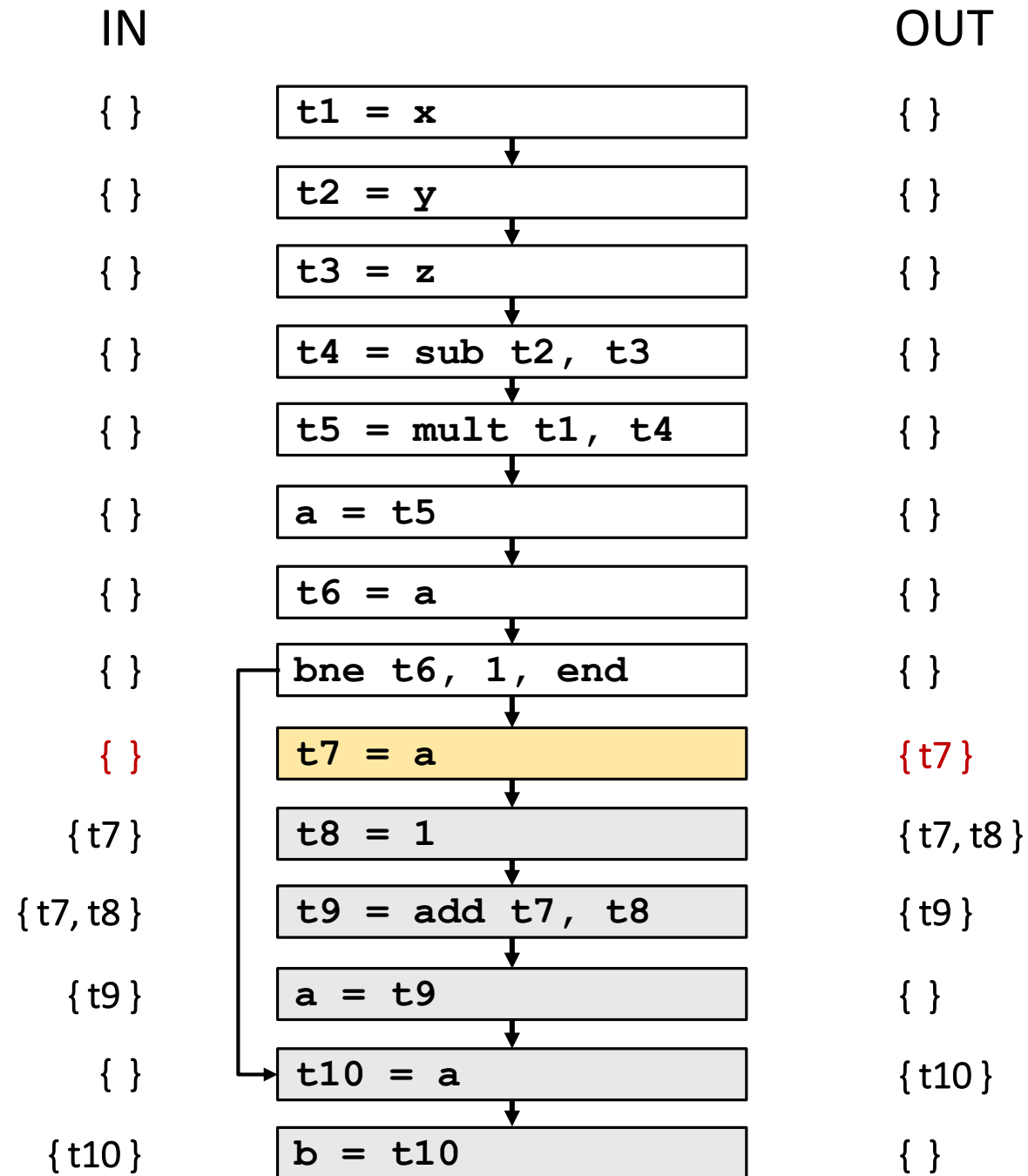


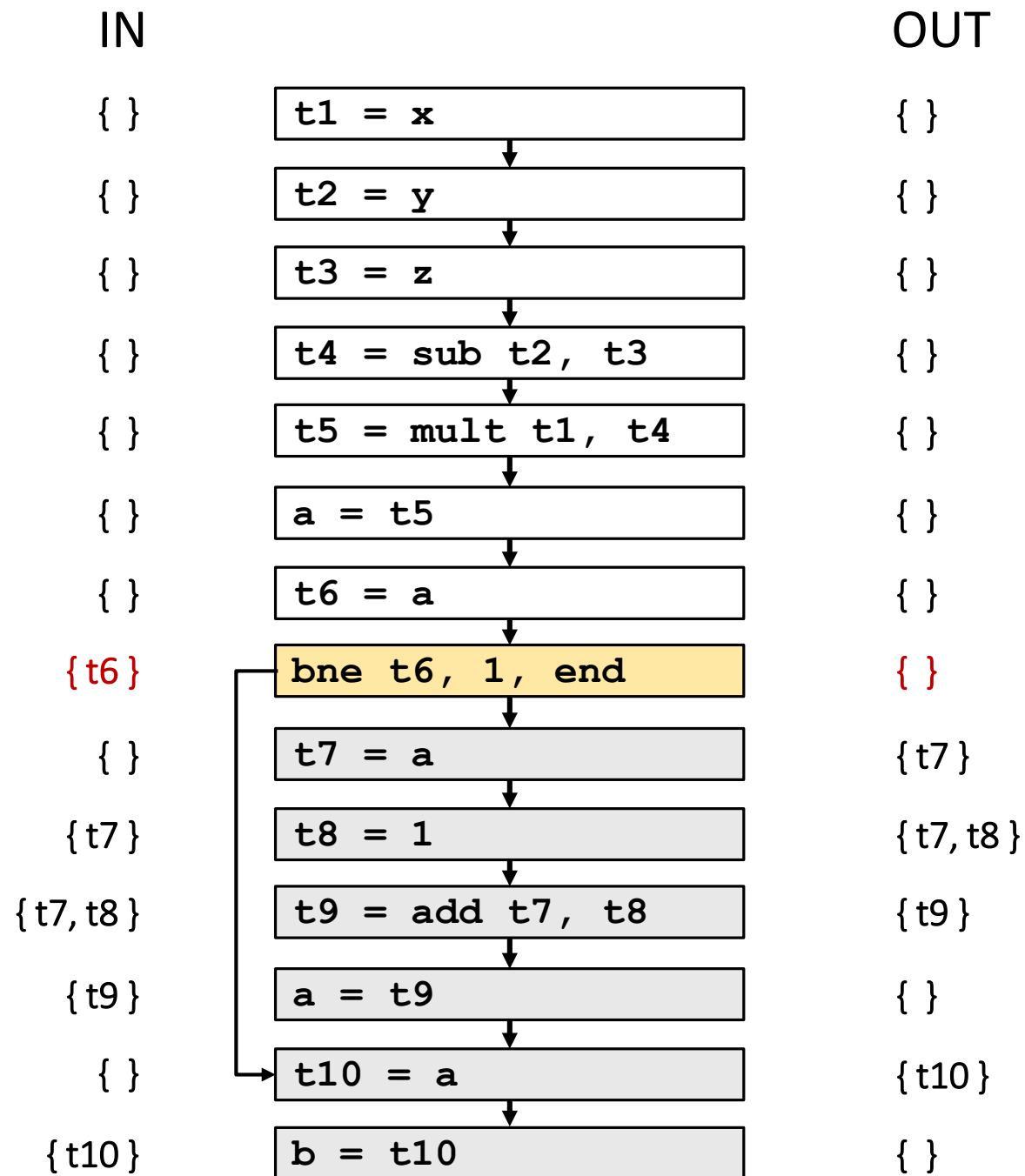


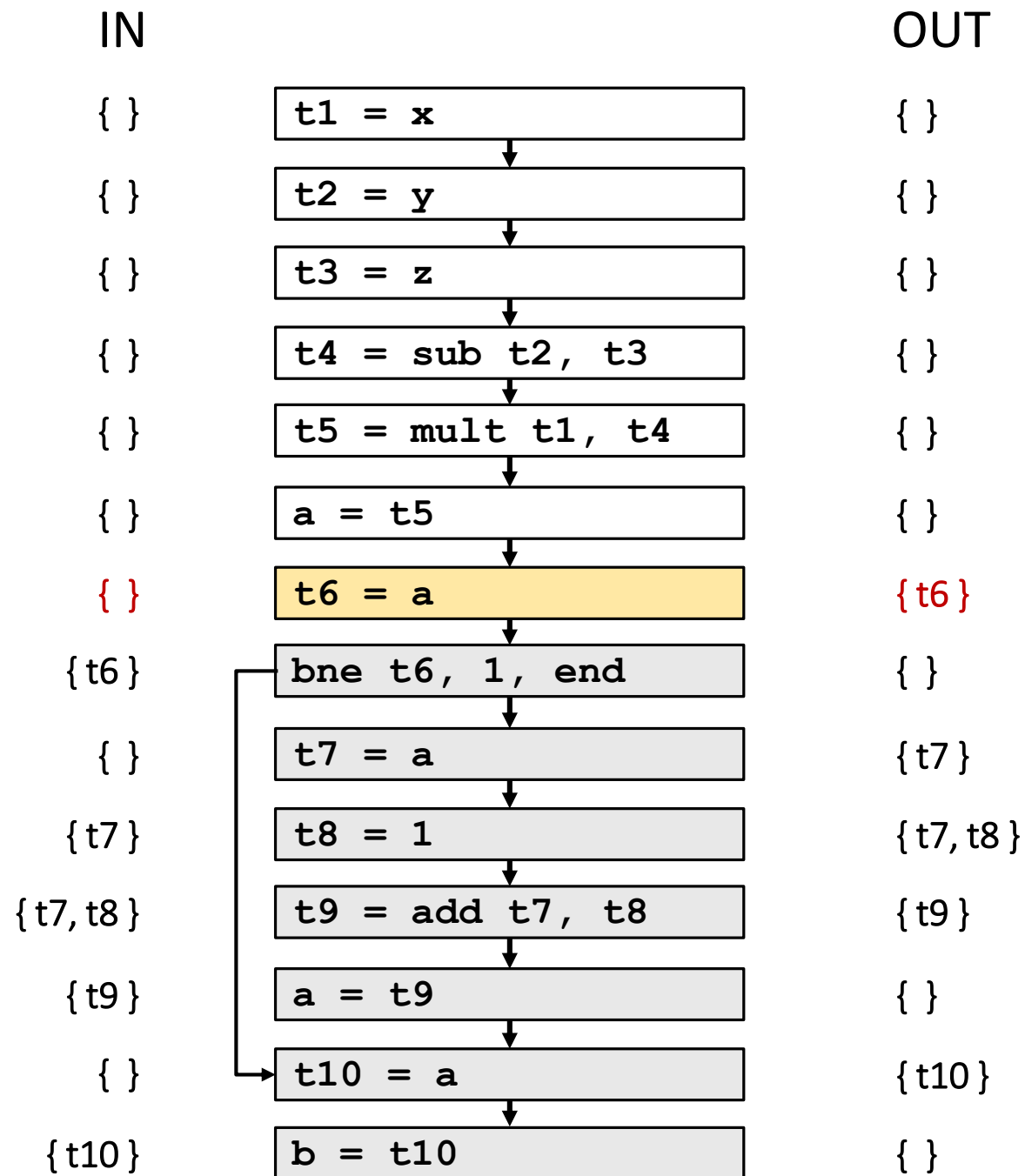


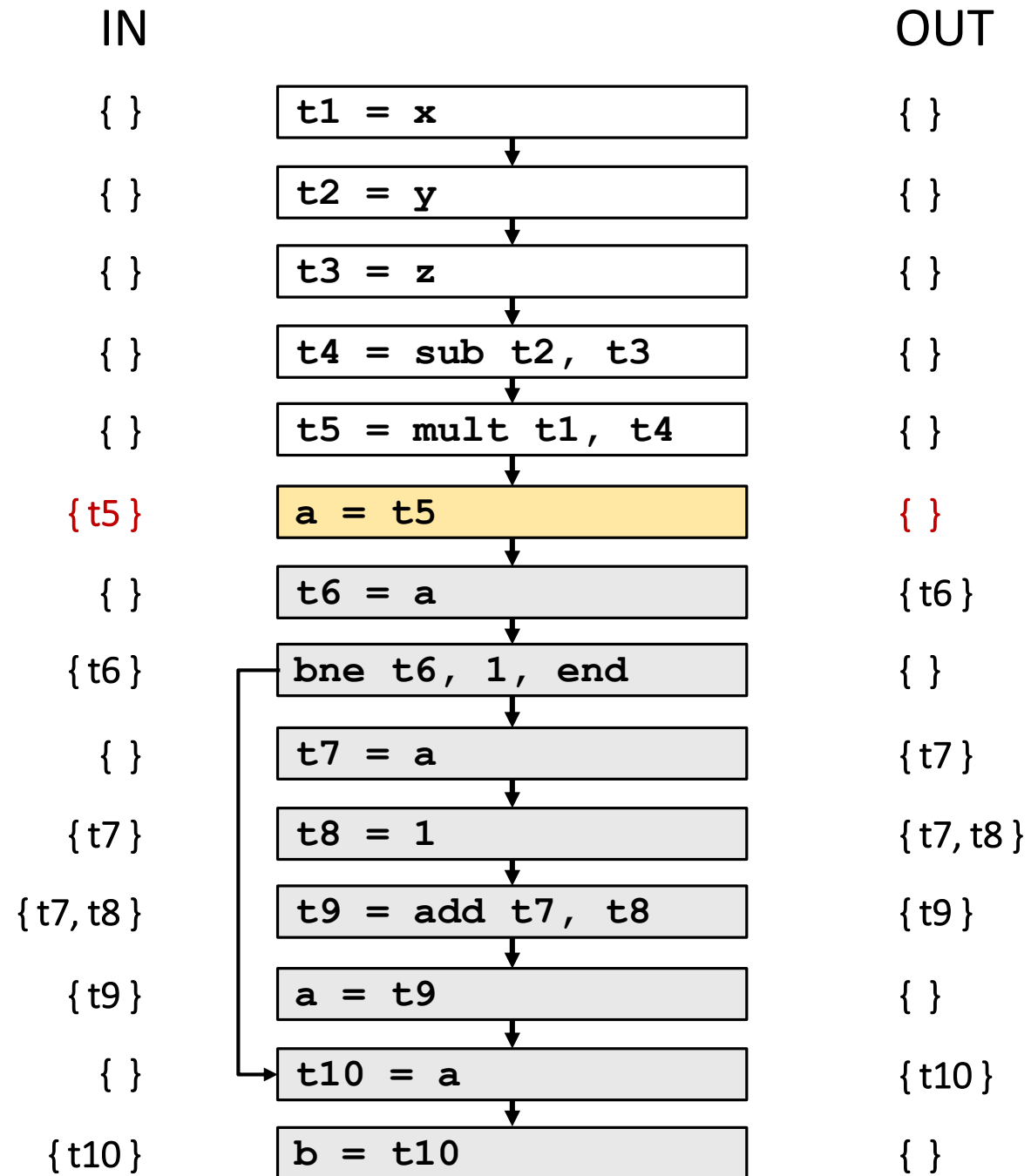


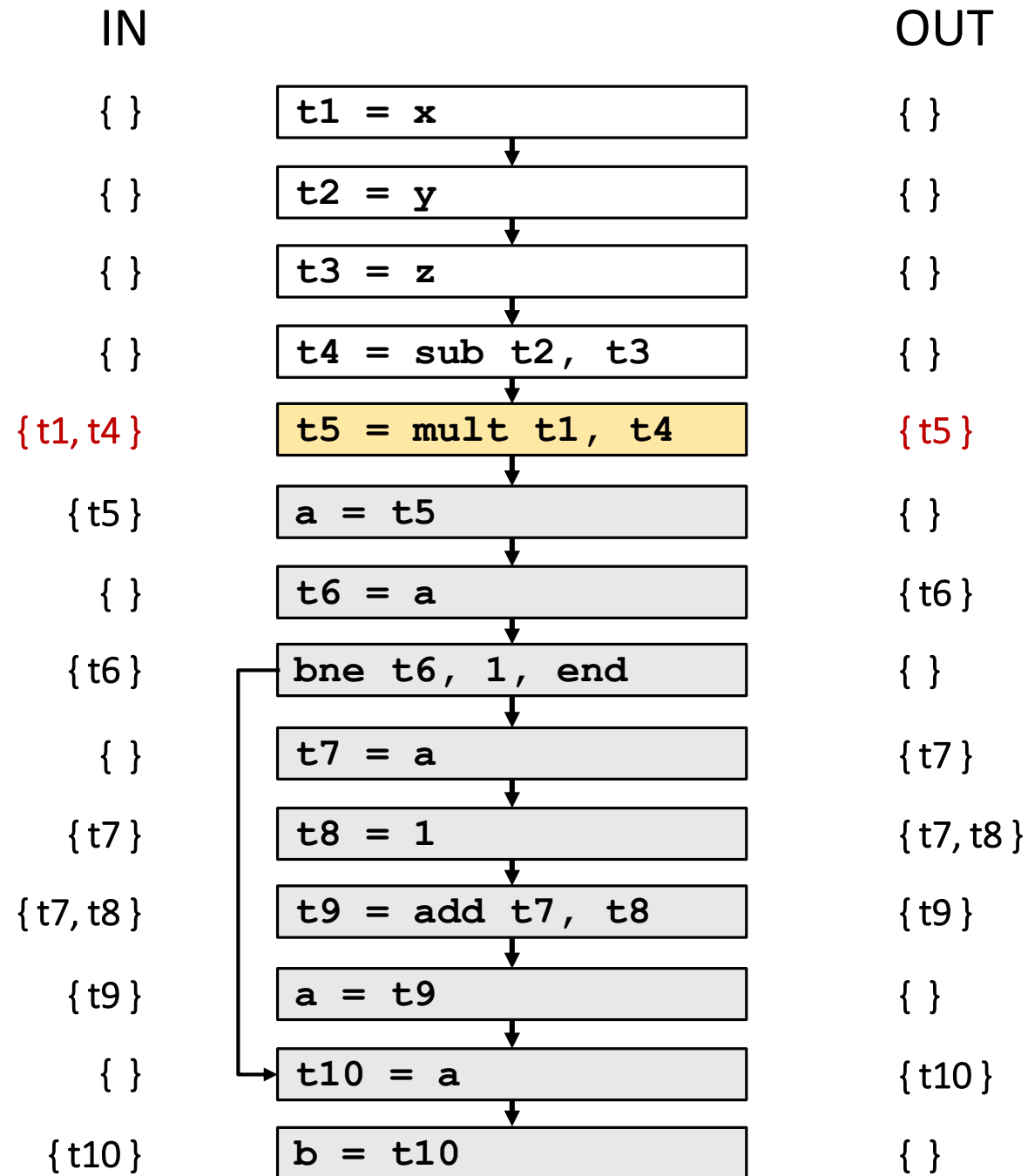


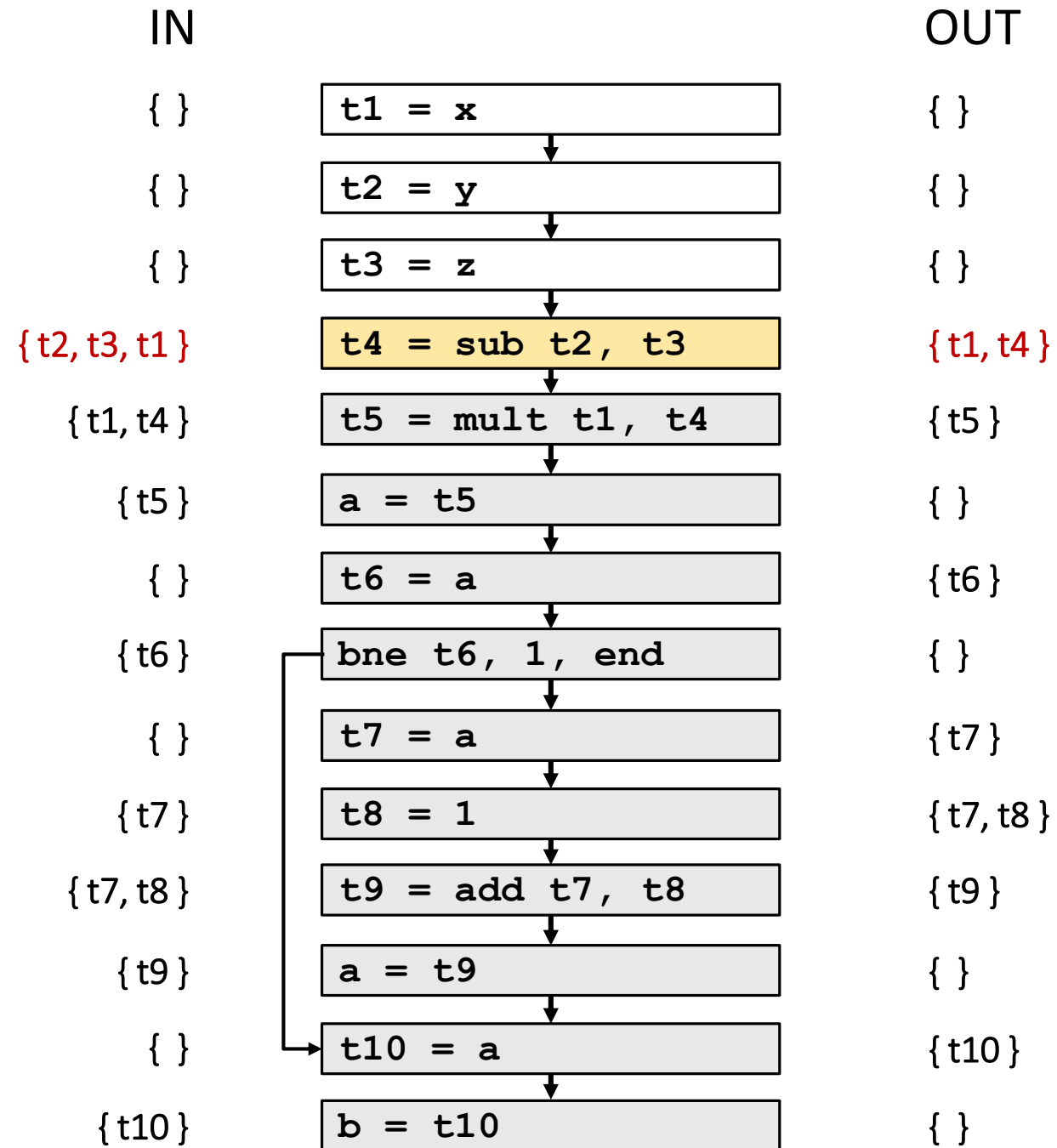


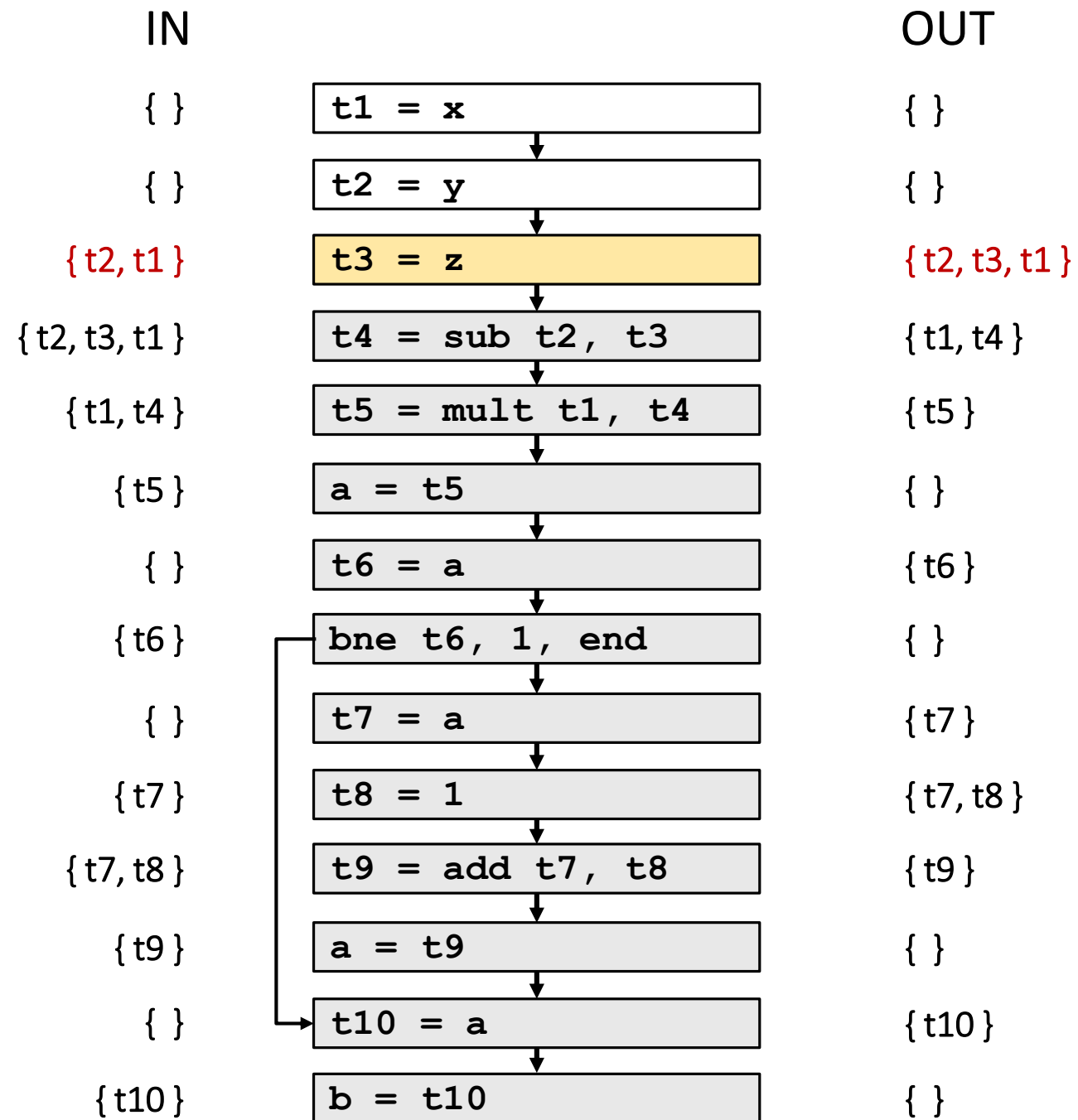


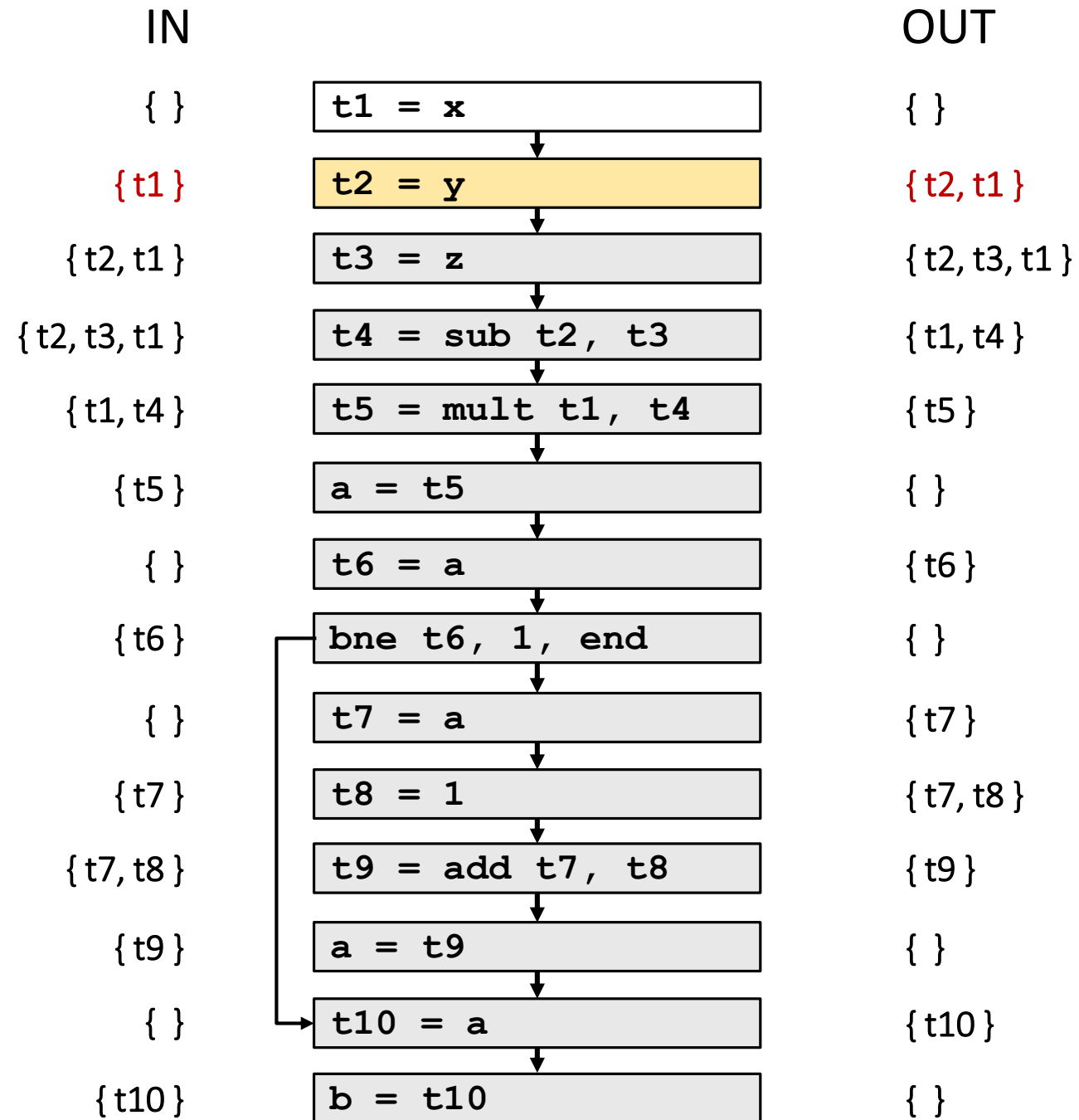






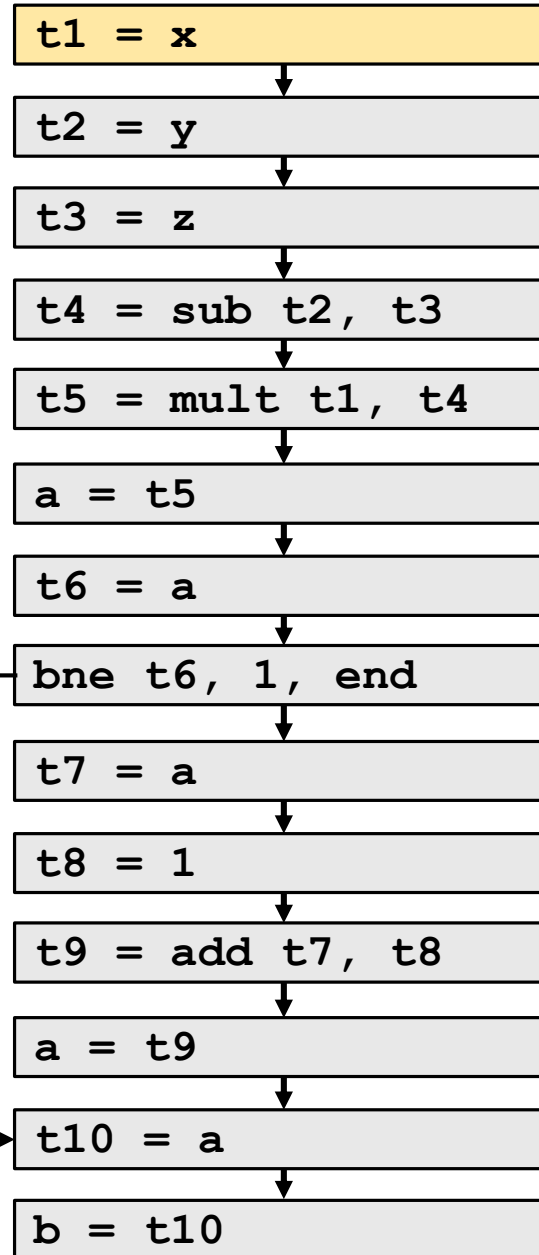






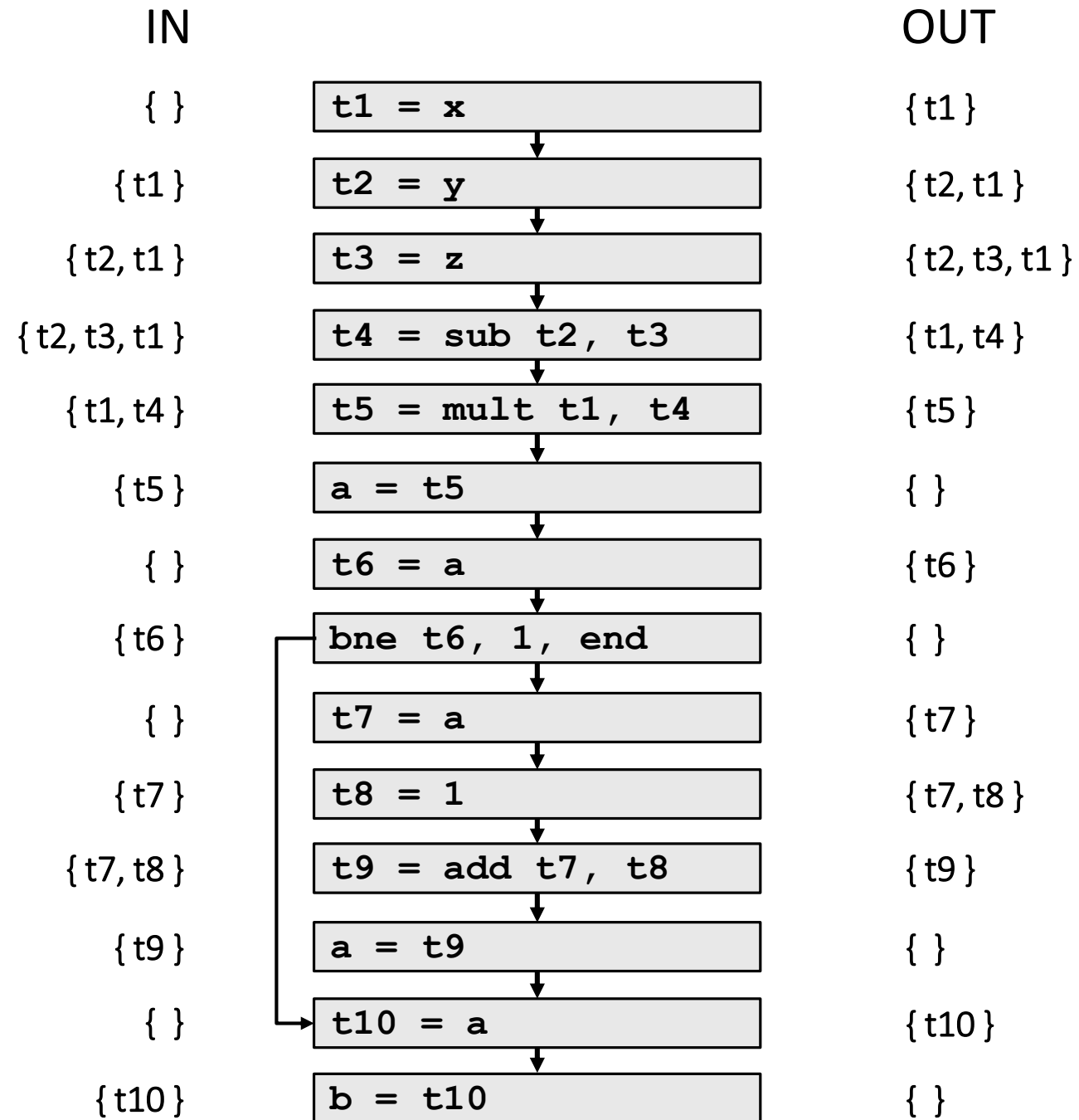
IN

{ }
{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ }
{ t6 }
{ }
{ t7 }
{ t7, t8 }
{ t9 }
{ }
{ t10 }



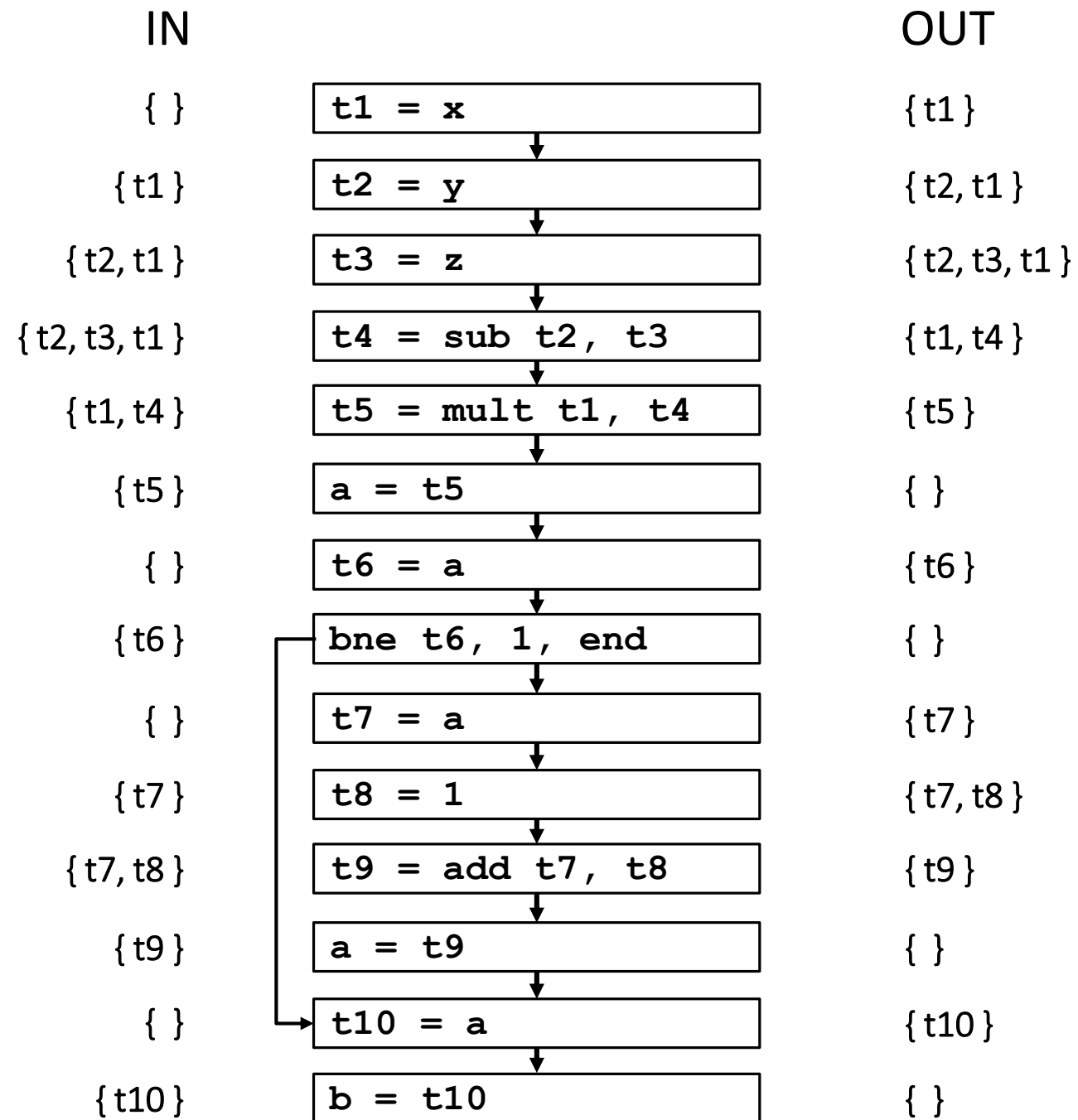
OUT

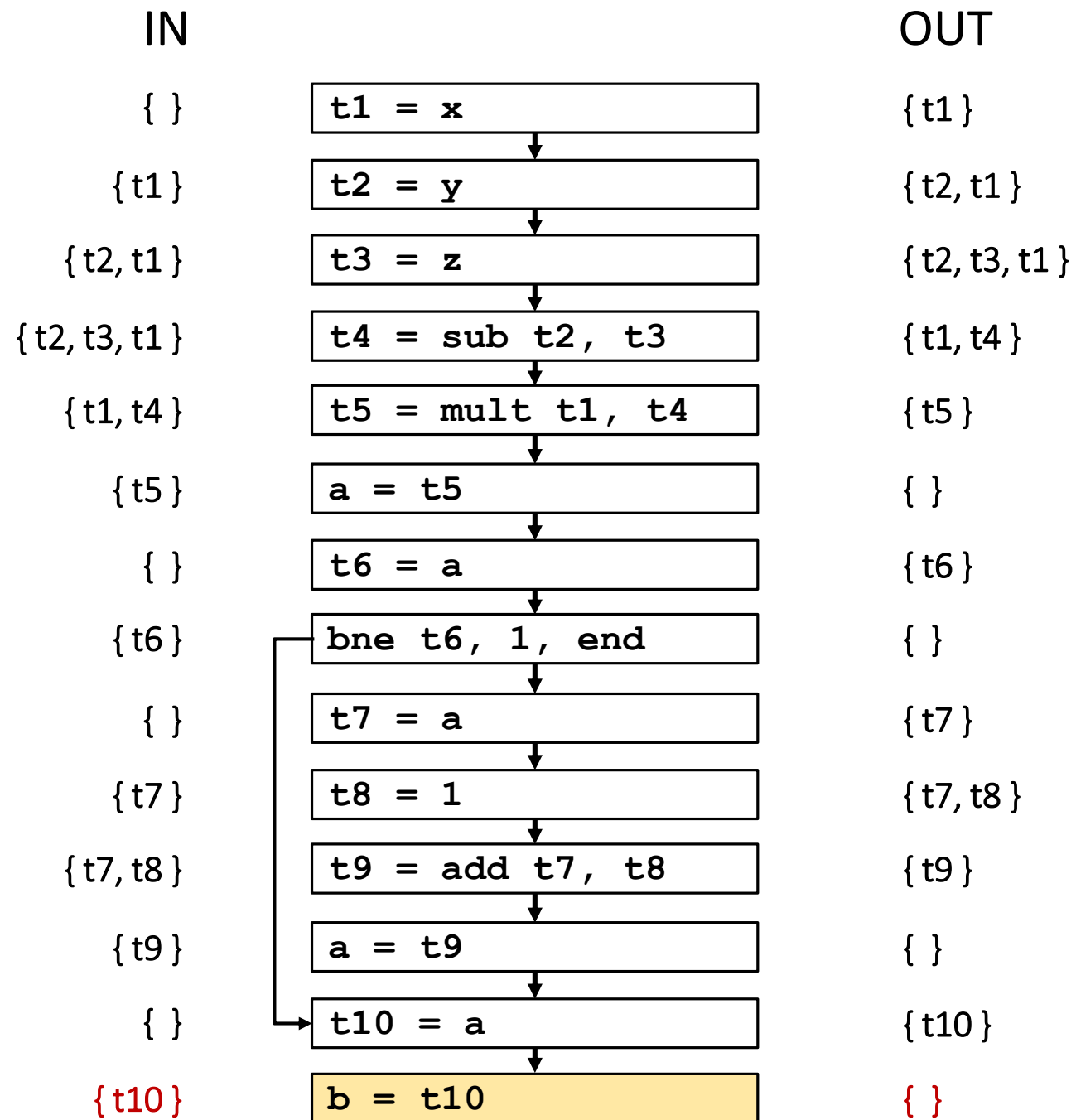
{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ }
{ t6 }
{ }
{ t7 }
{ t7, t8 }
{ t9 }
{ }
{ t10 }
{ }

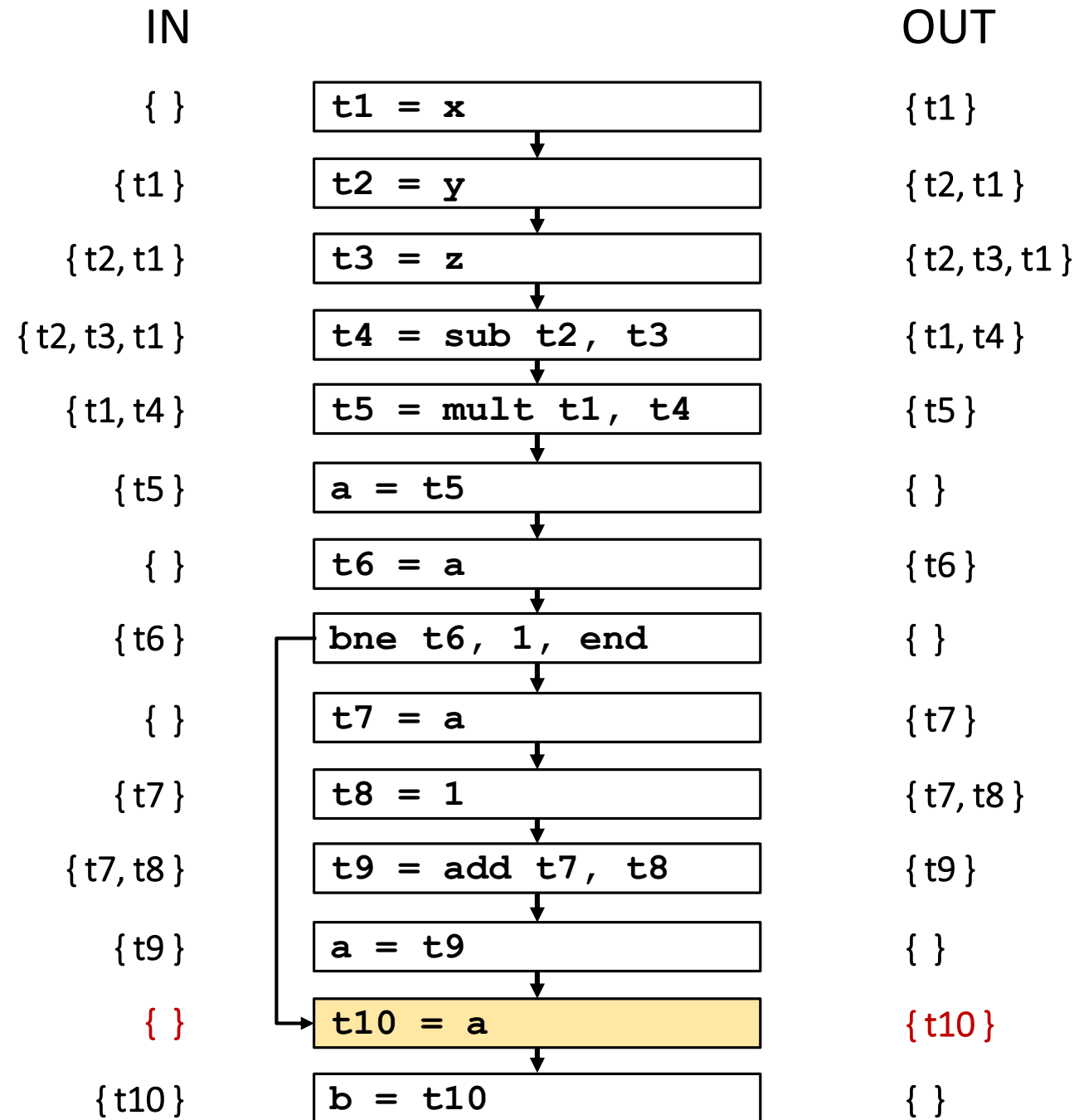


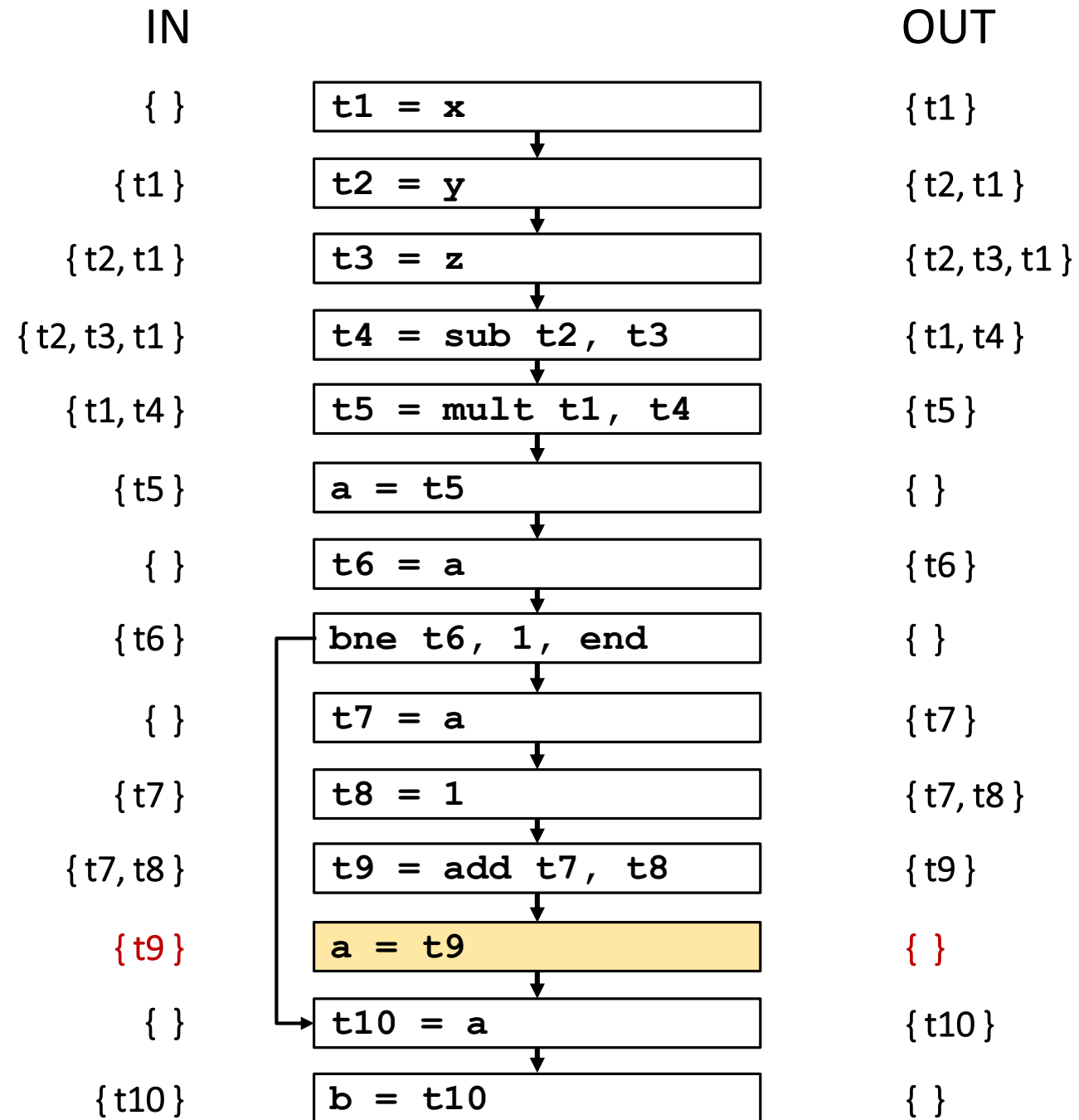
Liveness Analysis

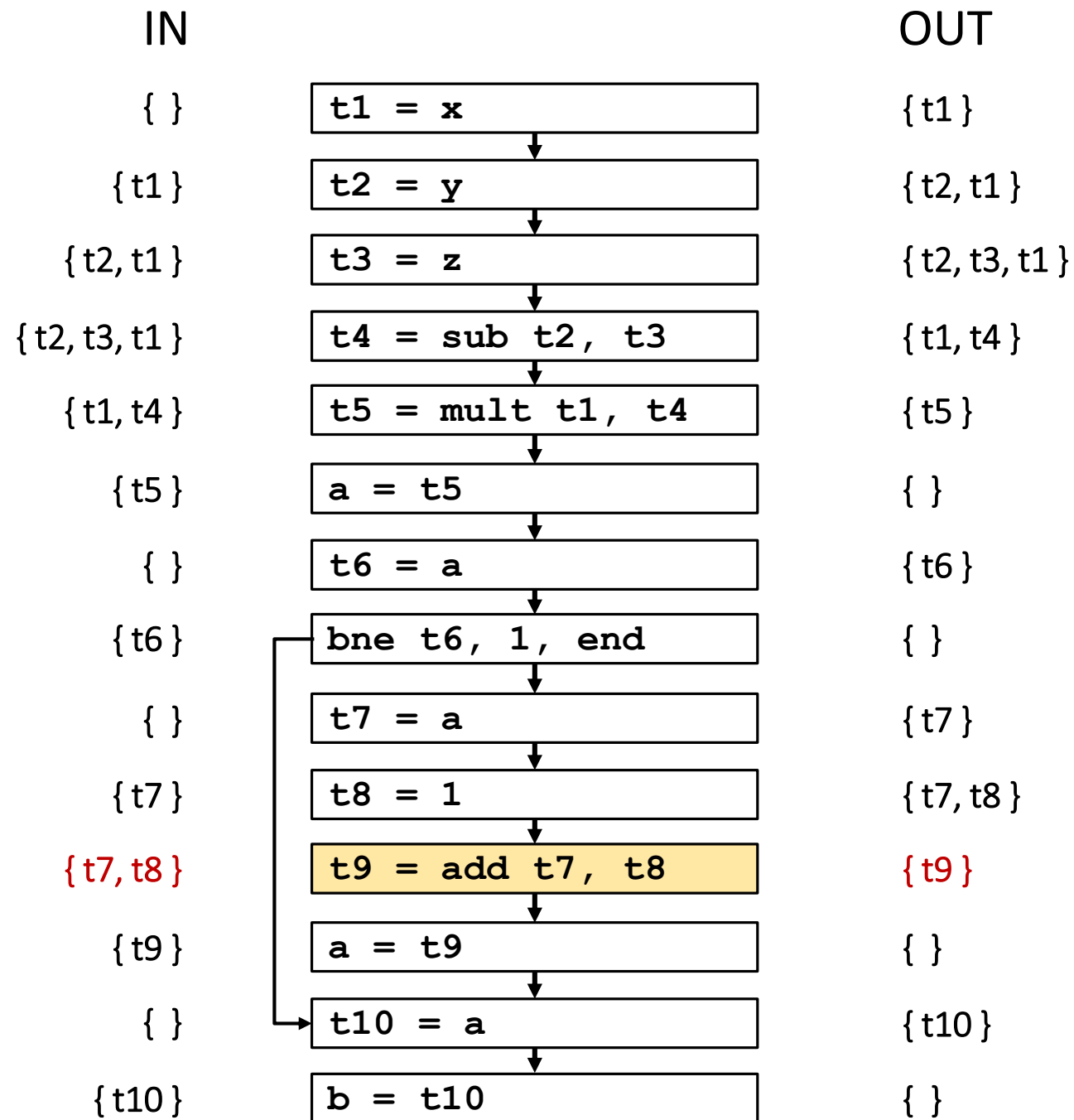
Second Iteration...

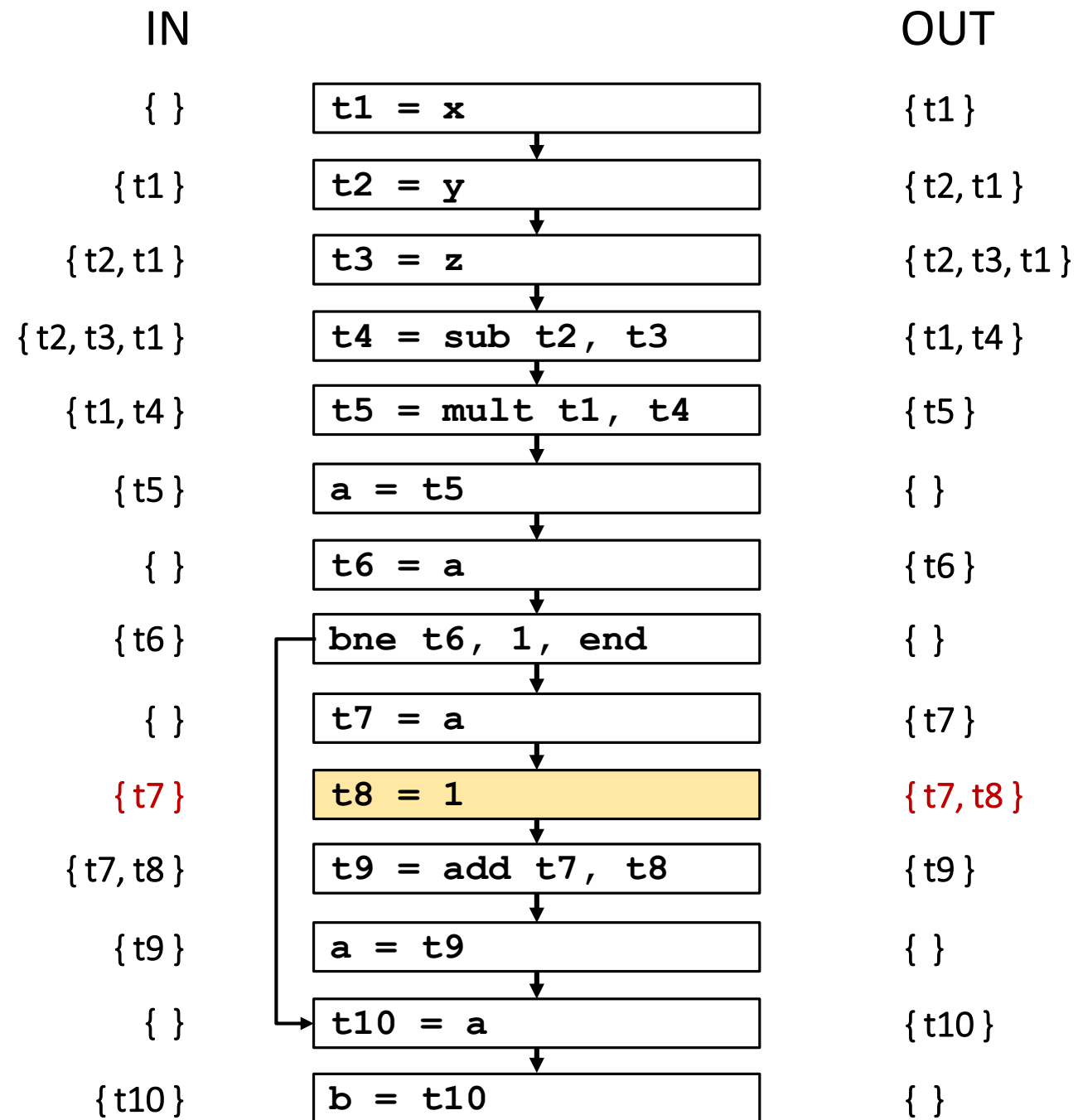


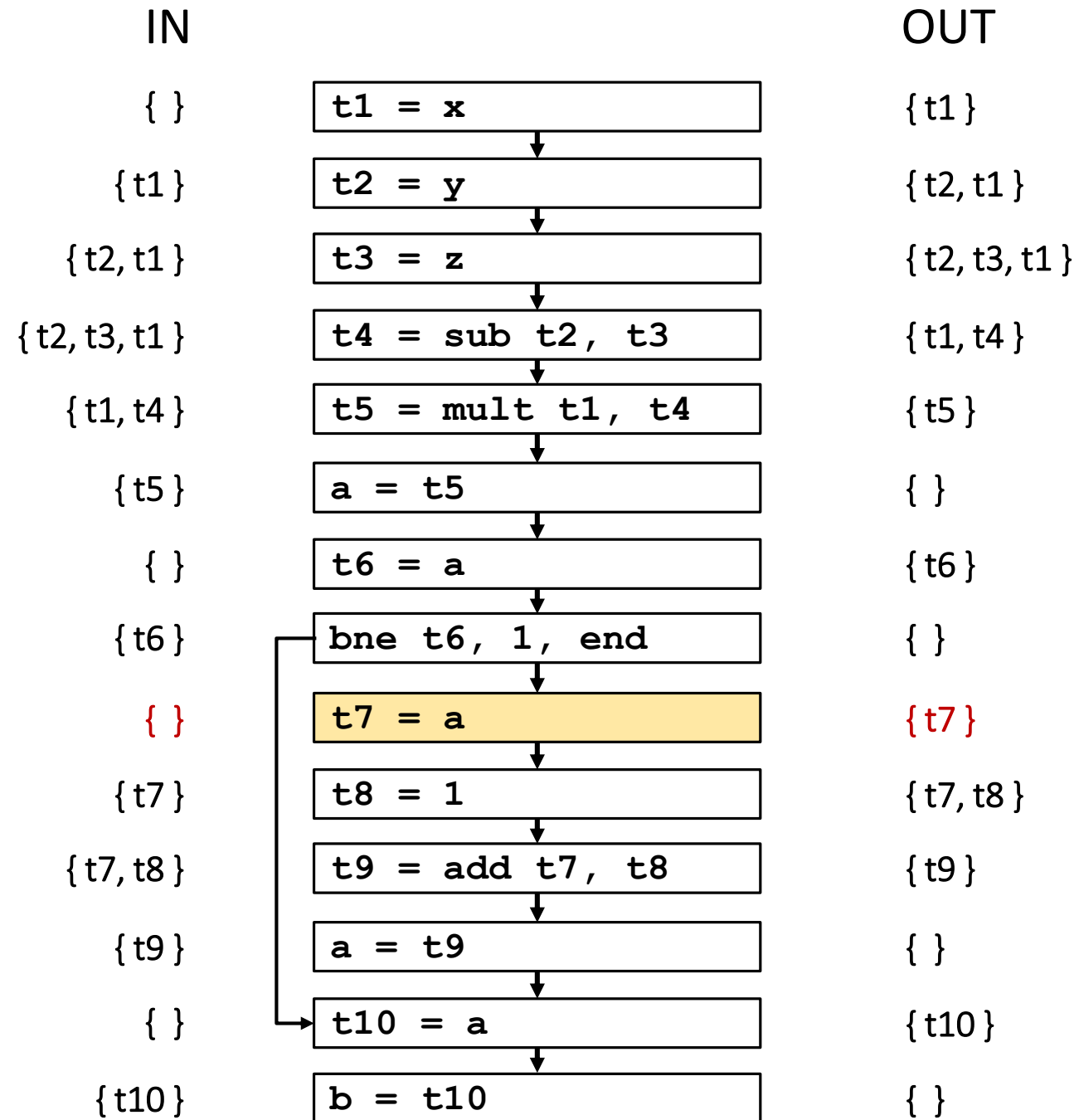


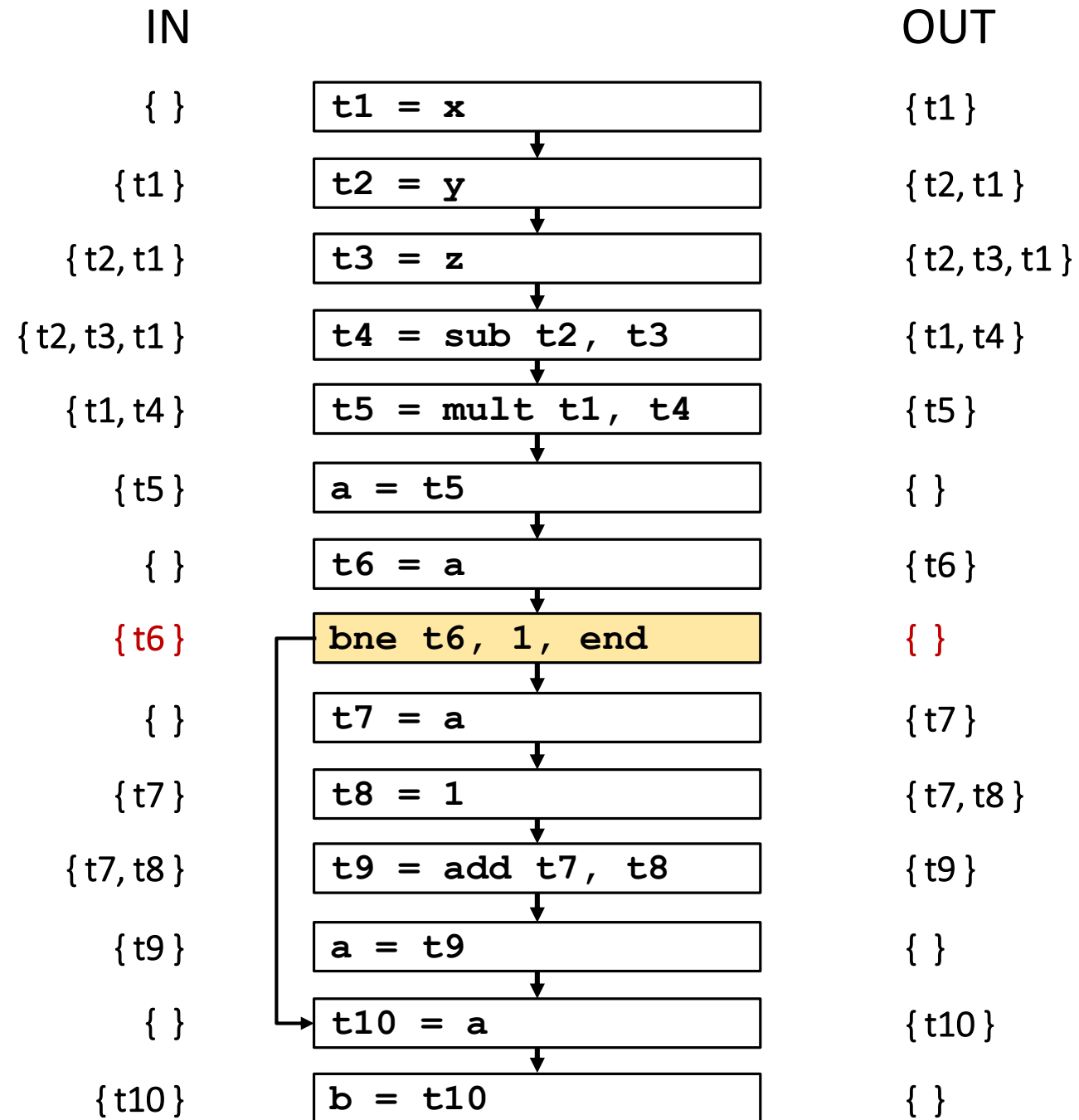


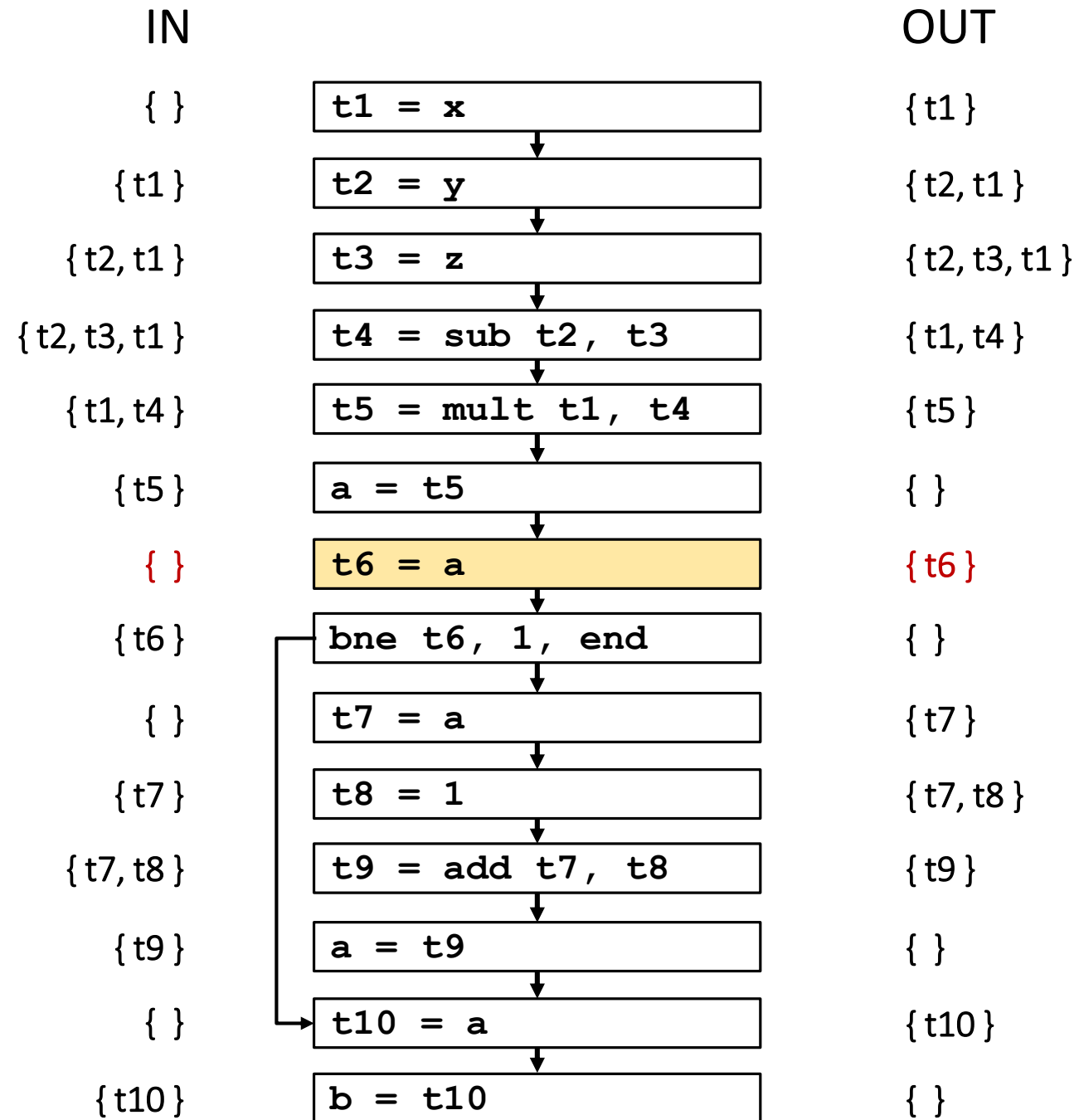


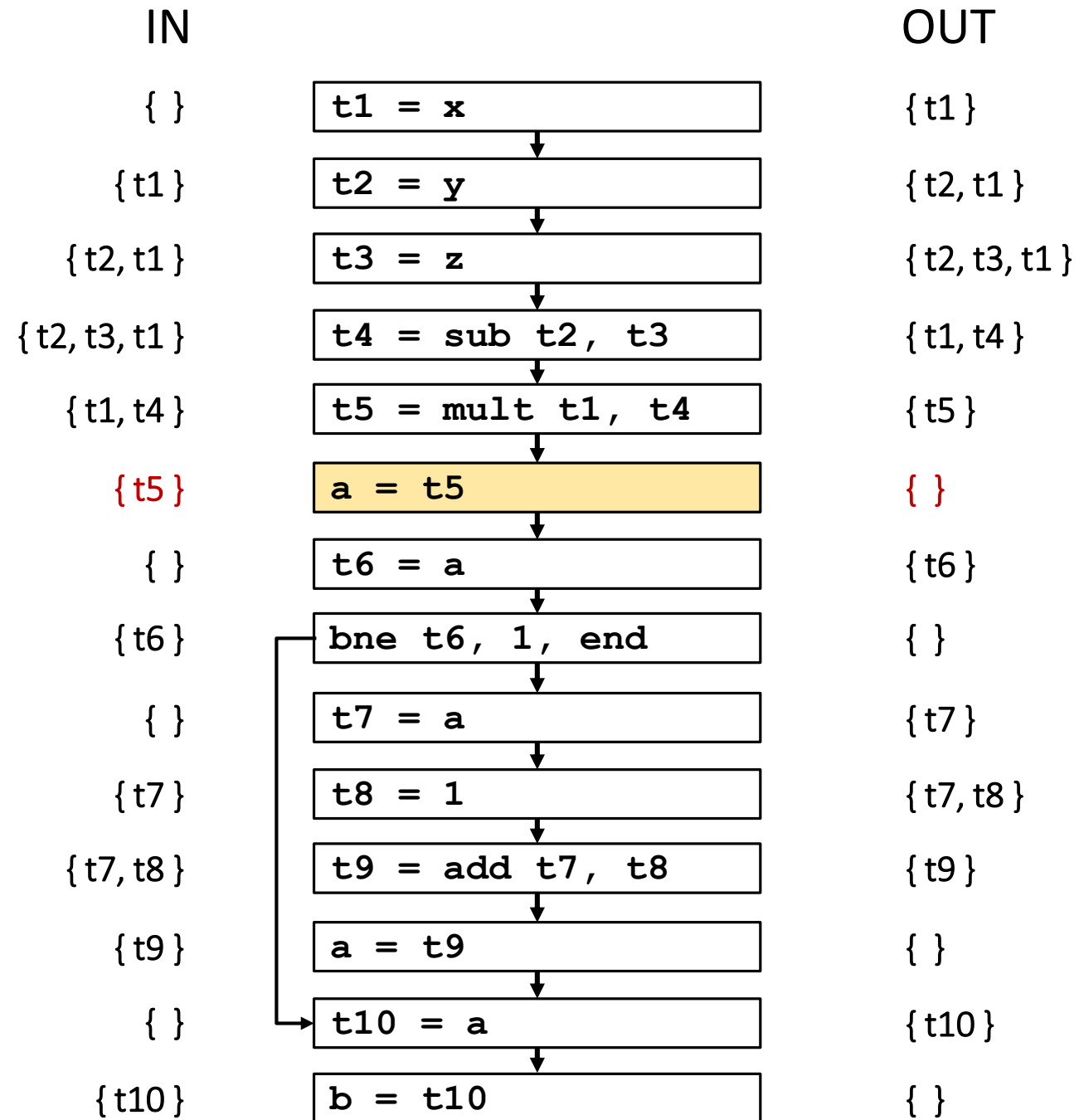


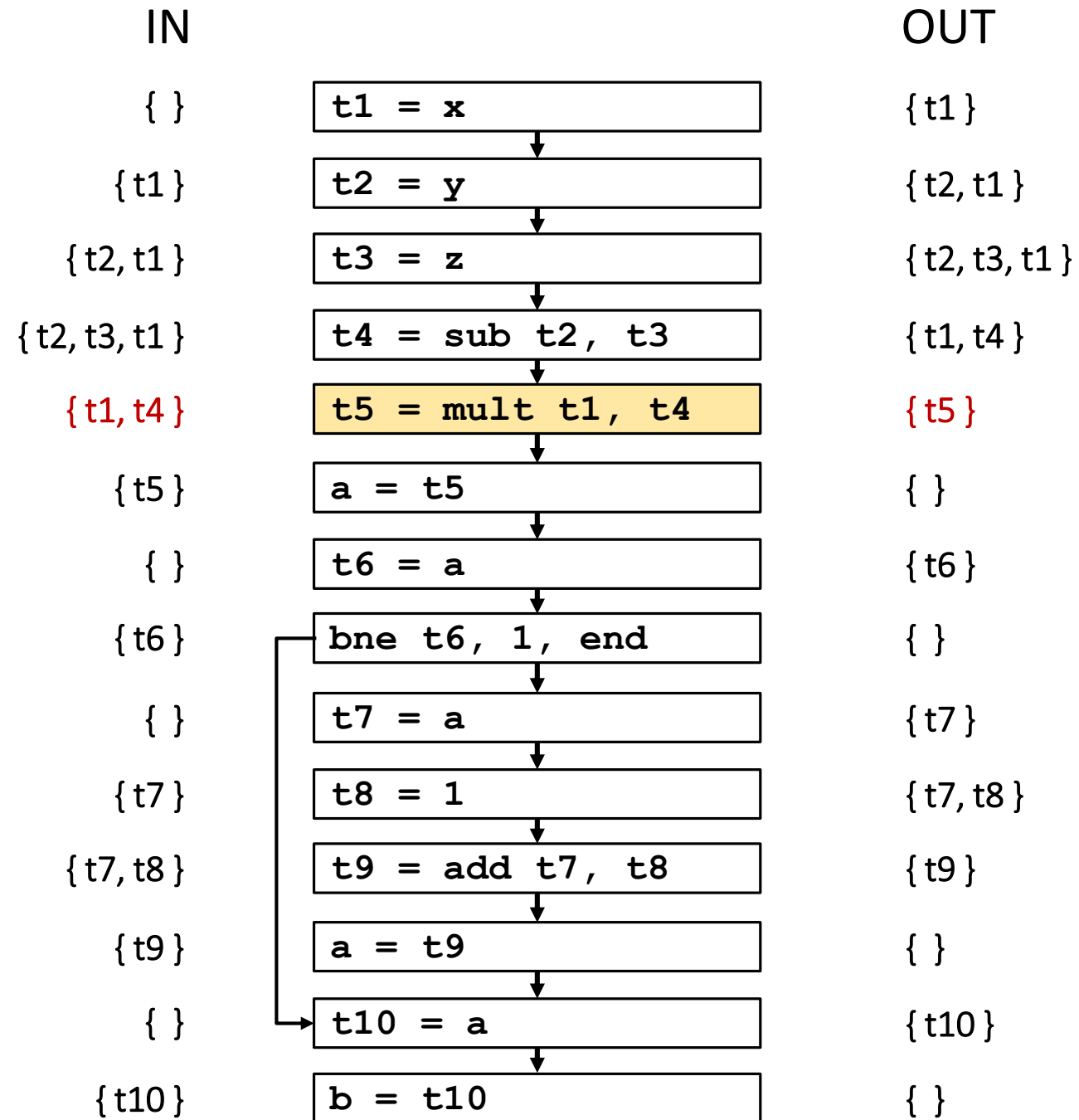


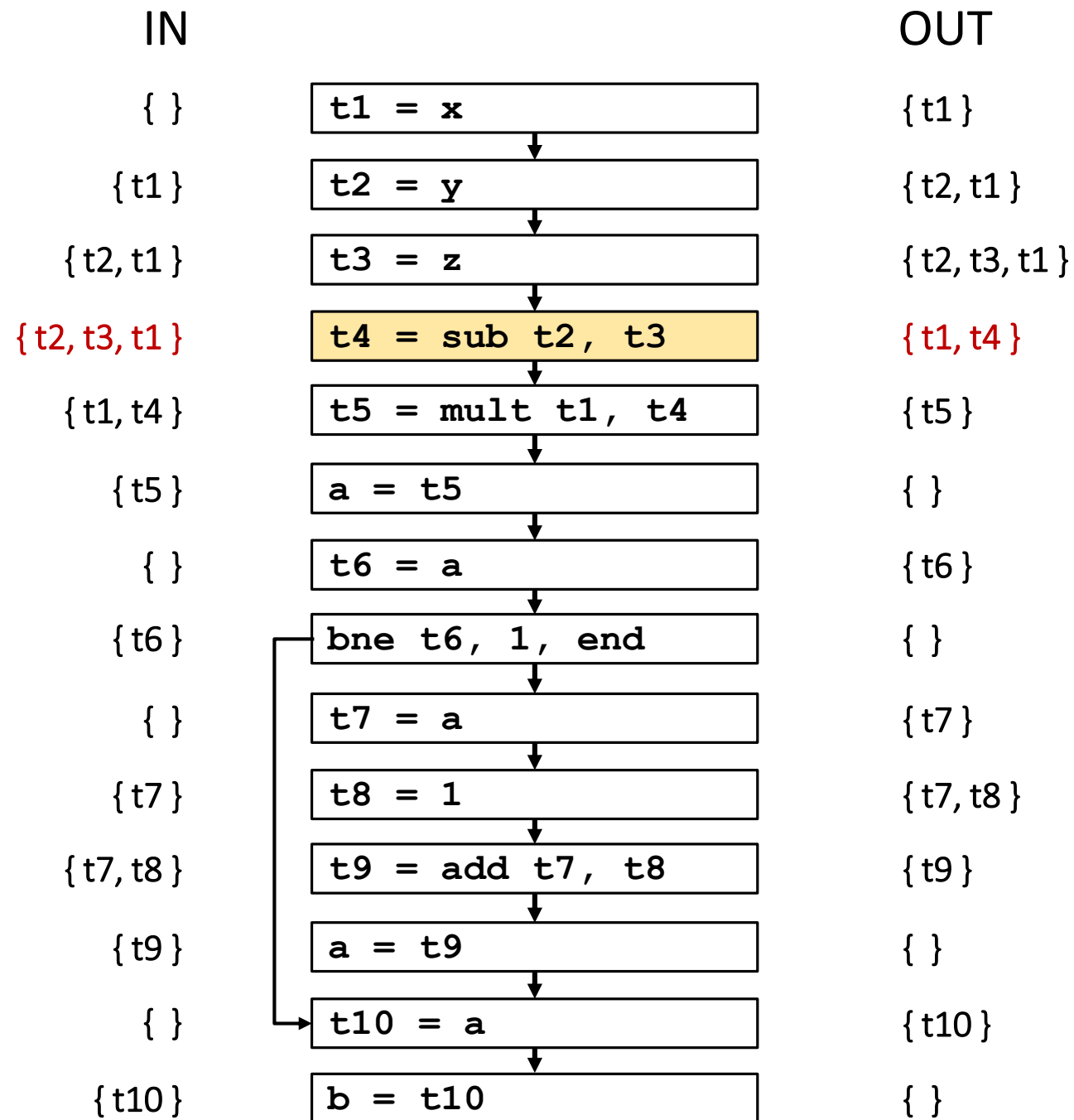


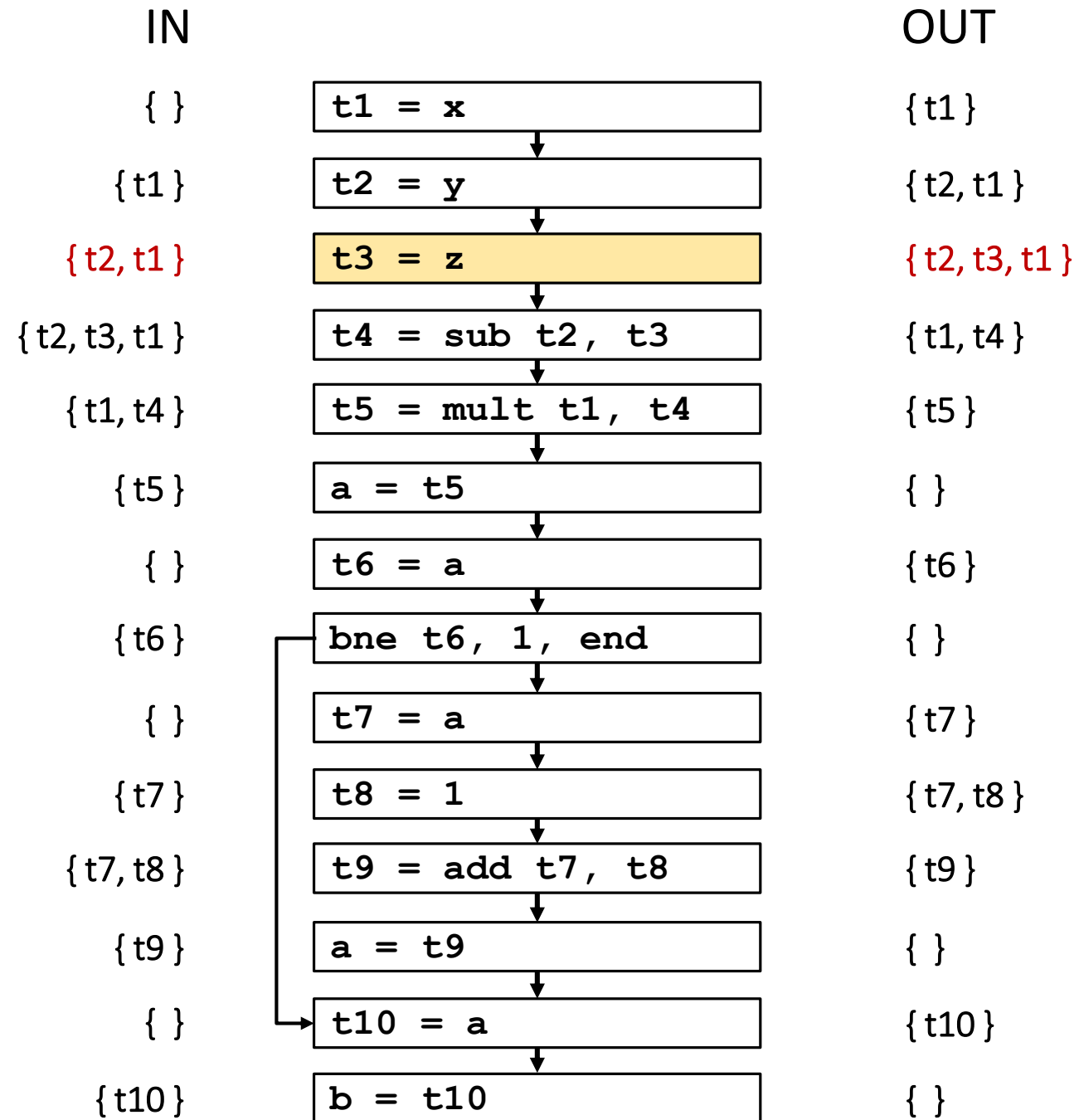


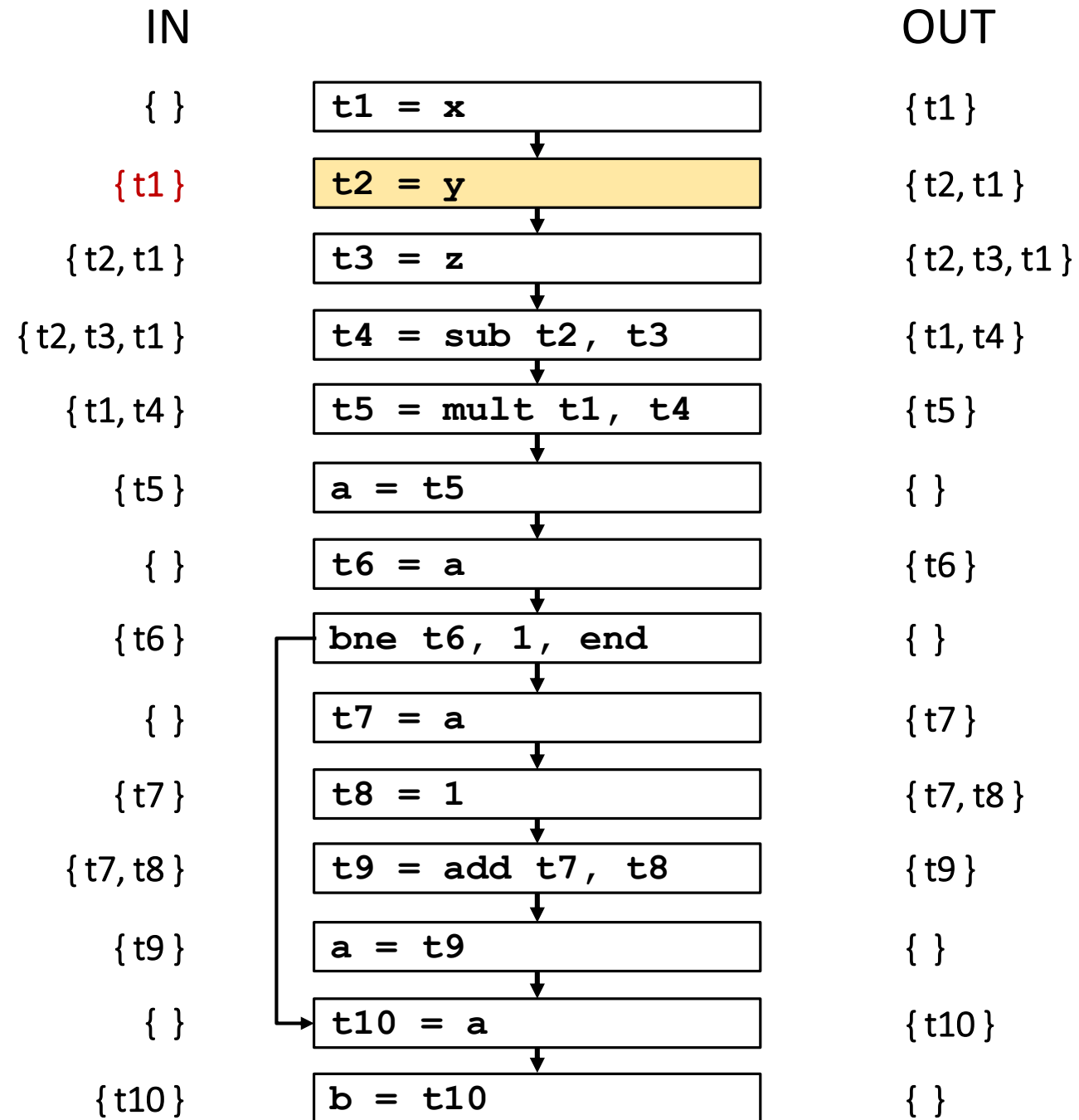






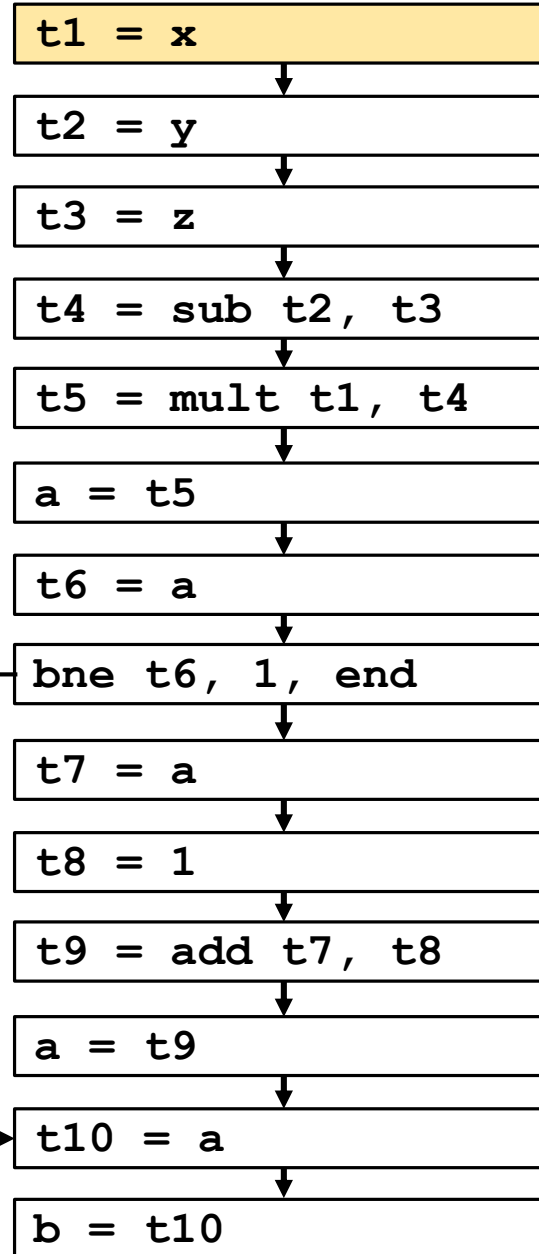






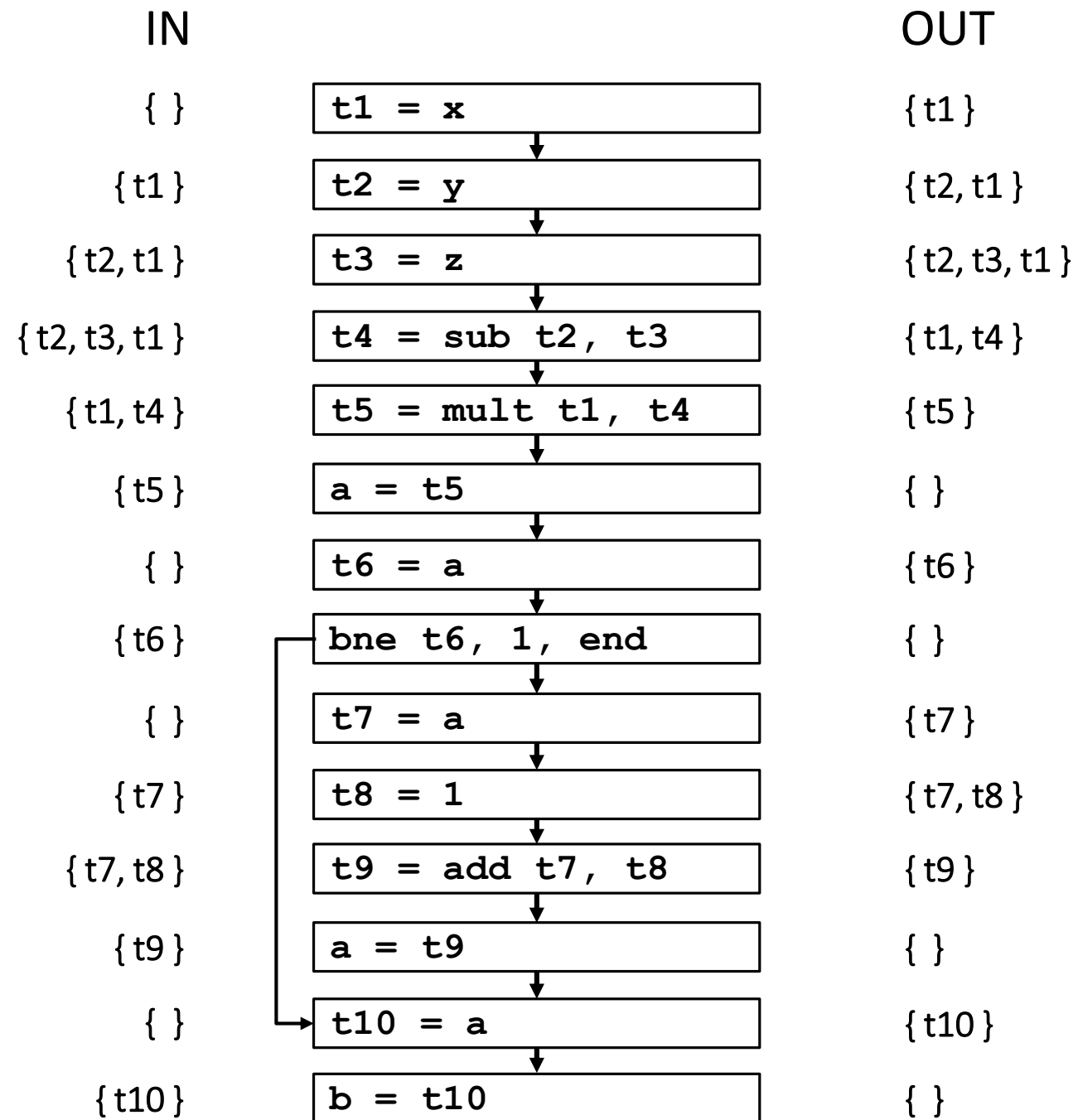
IN

{ }
{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ }
{ t6 }
{ }
{ t7 }
{ t7, t8 }
{ t9 }
{ }
{ t10 }



OUT

{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ }
{ t6 }
{ }
{ t7 }
{ t7, t8 }
{ t9 }
{ }
{ t10 }
{ }



Interference Graph

- Use liveness analysis to construct the **interference graph**
- Create a node for each IR register ($t1$, $t2$, ...)
- If $t1$ and $t2$ appear together in one of the liveness sets:
 - Create an edge between $t1$ and $t2$

Interference Graph

{ t1 }

{ t2, t1 }

{ t2, t3, t1 }

{ t1, t4 }

{ t5 }

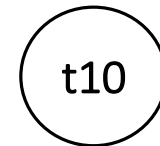
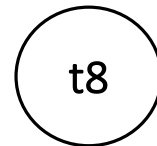
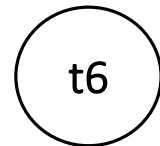
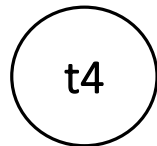
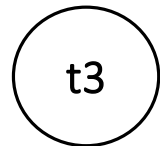
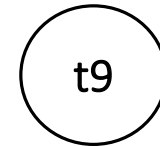
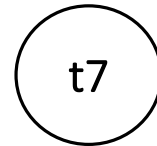
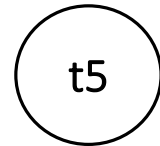
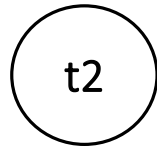
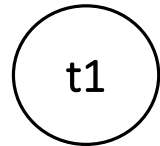
{ t6 }

{ t7 }

{ t7, t8 }

{ t9 }

{ t10 }



Interference Graph

$\{t1\}$

$\{t2, t1\}$

$\{t2, t3, t1\}$

$\{t1, t4\}$

$\{t5\}$

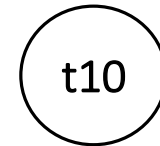
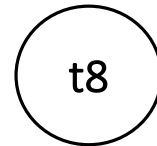
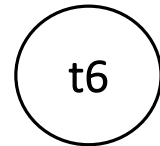
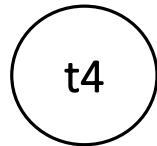
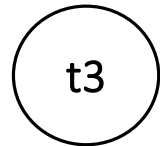
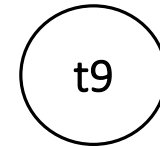
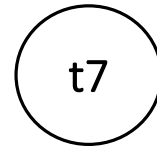
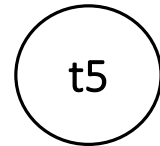
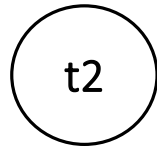
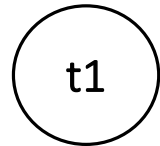
$\{t6\}$

$\{t7\}$

$\{t7, t8\}$

$\{t9\}$

$\{t10\}$



Interference Graph

{ t1 }

{ t2, t1 }

{ t2, t3, t1 }

{ t1, t4 }

{ t5 }

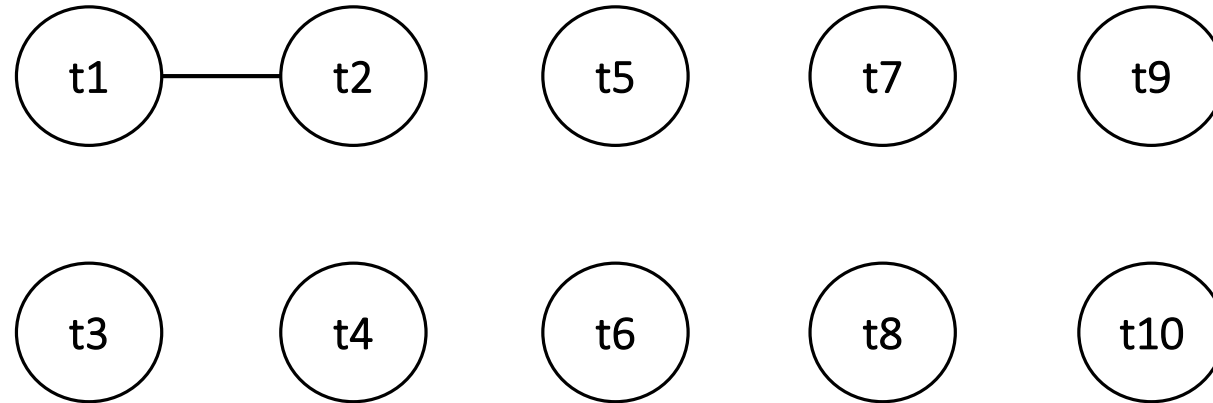
{ t6 }

{ t7 }

{ t7, t8 }

{ t9 }

{ t10 }



Interference Graph

{ t1 }

{ t2, t1 }

{ t2, t3, t1 }

{ t1, t4 }

{ t5 }

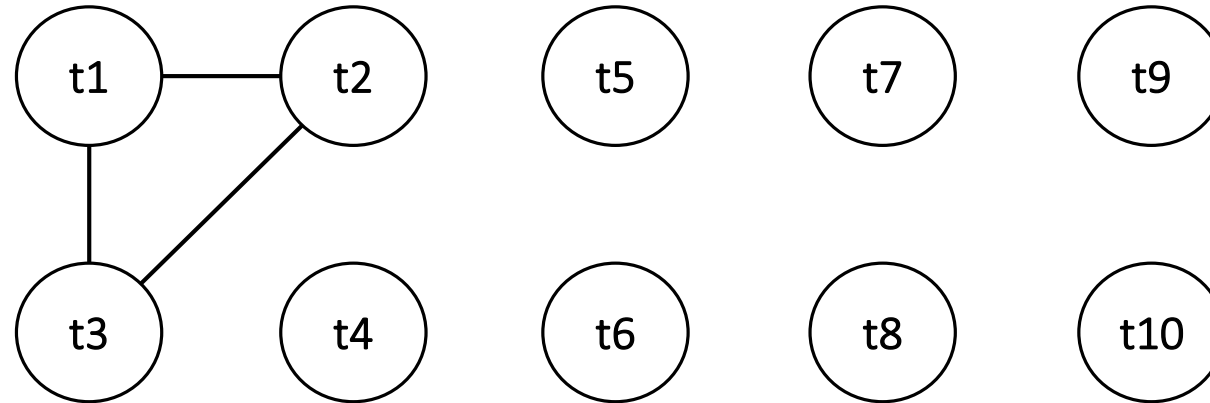
{ t6 }

{ t7 }

{ t7, t8 }

{ t9 }

{ t10 }



Interference Graph

{ t1 }

{ t2, t1 }

{ t2, t3, t1 }

{ t1, t4 }

{ t5 }

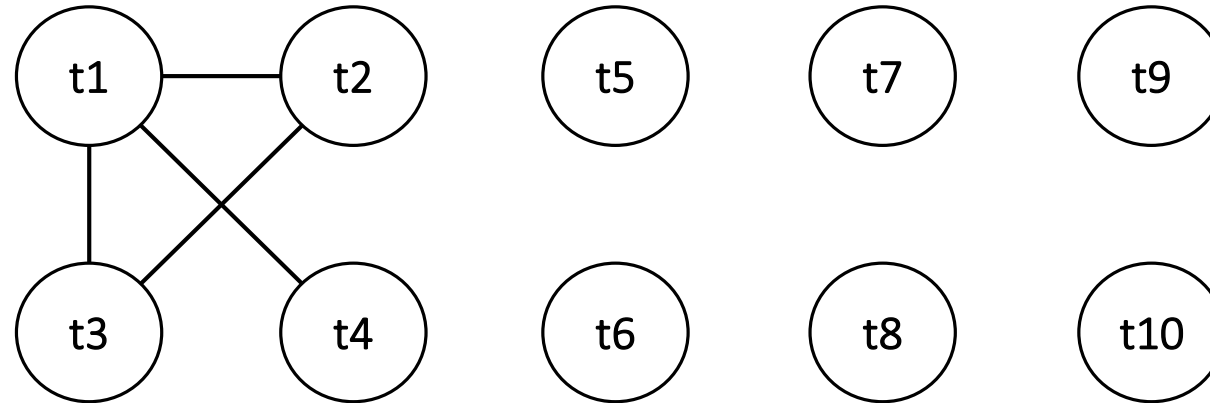
{ t6 }

{ t7 }

{ t7, t8 }

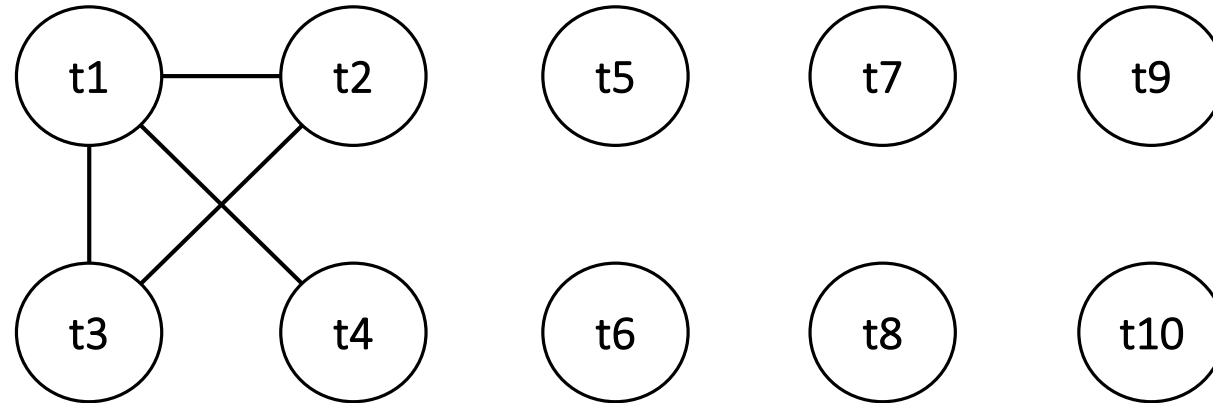
{ t9 }

{ t10 }



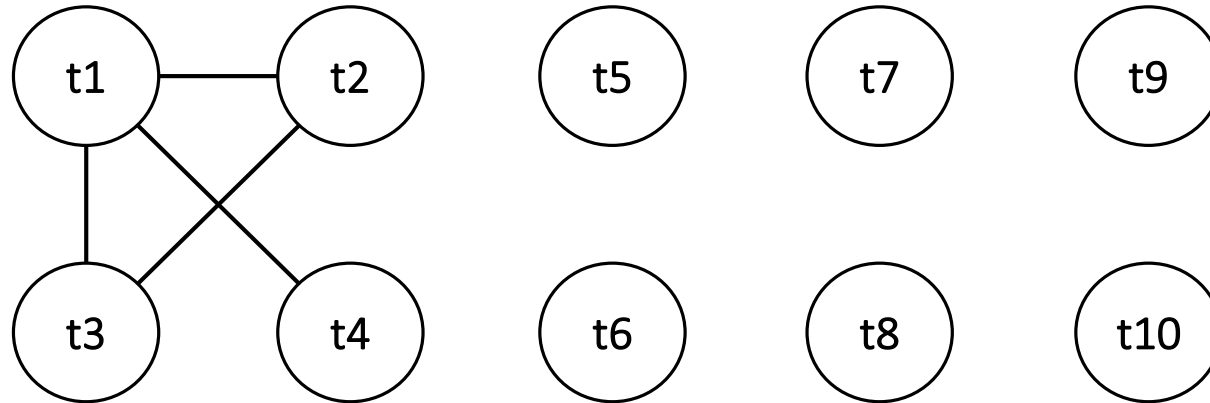
Interference Graph

{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ t6 }
{ t7 }
{ t7, t8 }
{ t9 }
{ t10 }



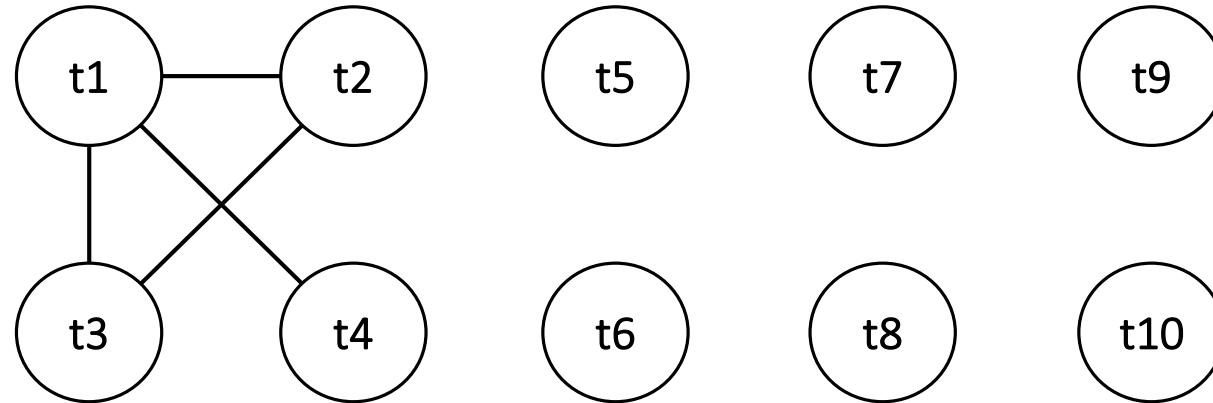
Interference Graph

{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ t6 }
{ t7 }
{ t7, t8 }
{ t9 }
{ t10 }



Interference Graph

{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ t6 }
{ t7 }
{ t7, t8 }
{ t9 }
{ t10 }



Interference Graph

{ t1 }

{ t2, t1 }

{ t2, t3, t1 }

{ t1, t4 }

{ t5 }

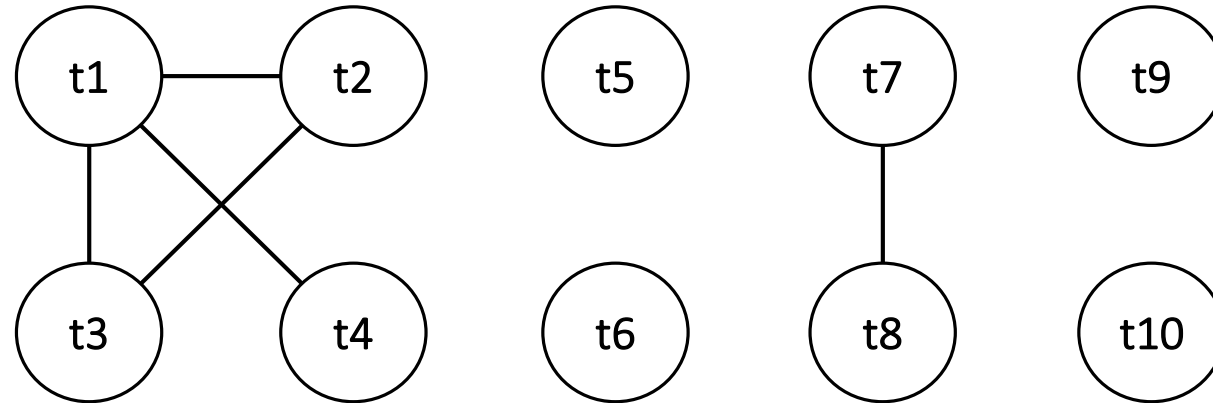
{ t6 }

{ t7 }

{ t7, t8 }

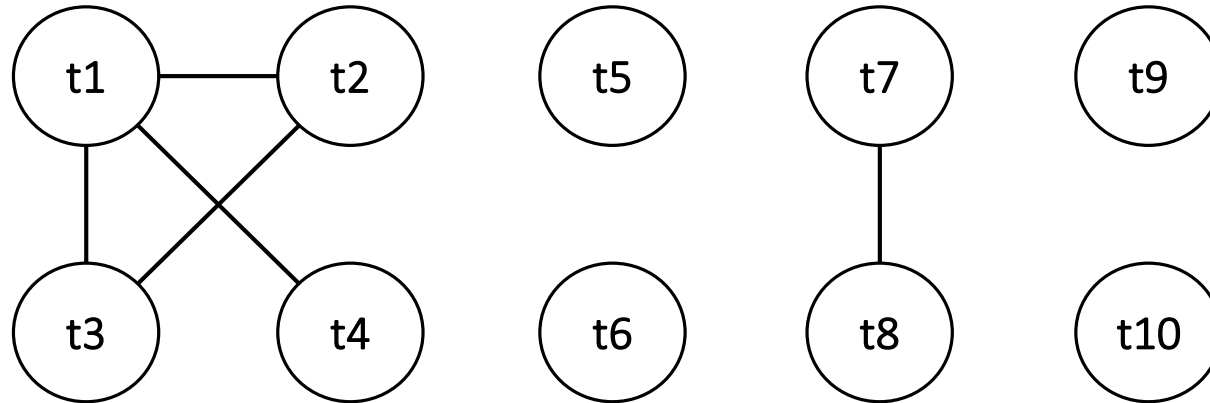
{ t9 }

{ t10 }



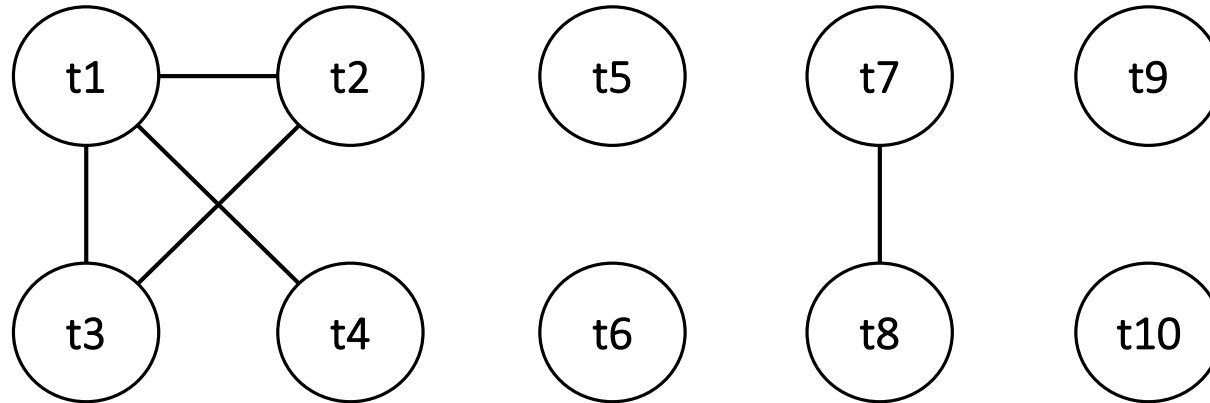
Interference Graph

{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ t6 }
{ t7 }
{ t7, t8 }
{ t9 }
{ t10 }



Interference Graph

{ t1 }
{ t2, t1 }
{ t2, t3, t1 }
{ t1, t4 }
{ t5 }
{ t6 }
{ t7 }
{ t7, t8 }
{ t9 }
{ t10 }

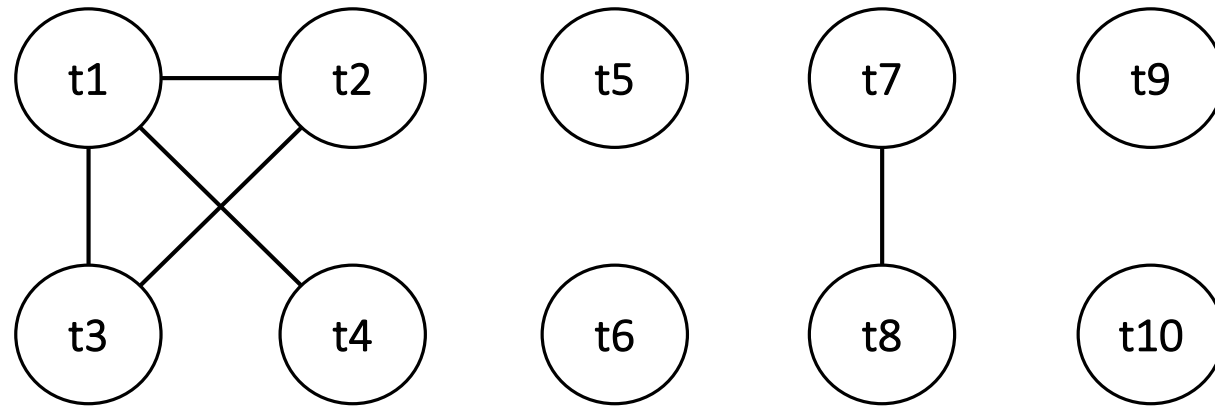


Graph Coloring

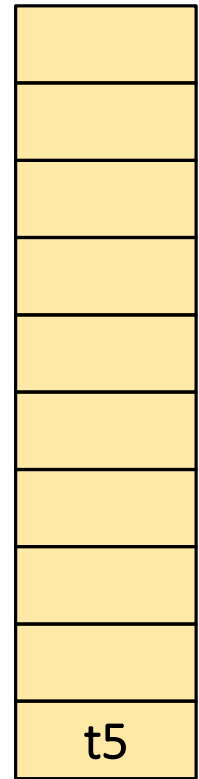
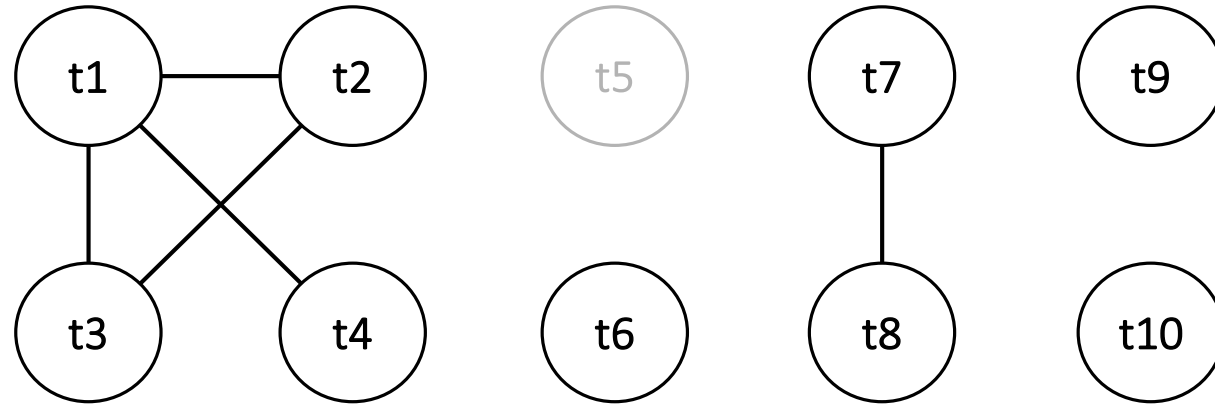
Chaitin's Algorithm:

- TODO

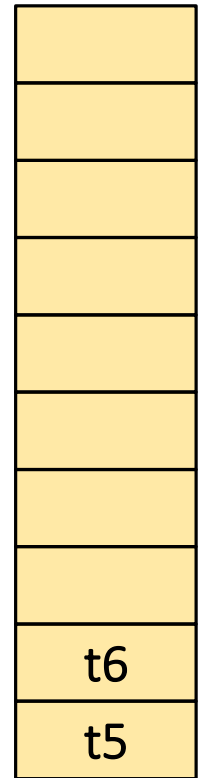
Graph Coloring ($k = 3$)



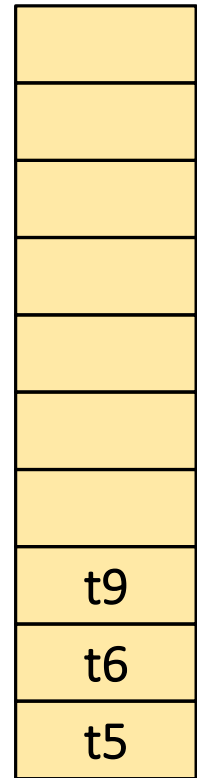
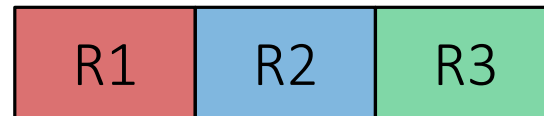
Graph Coloring ($k = 3$)



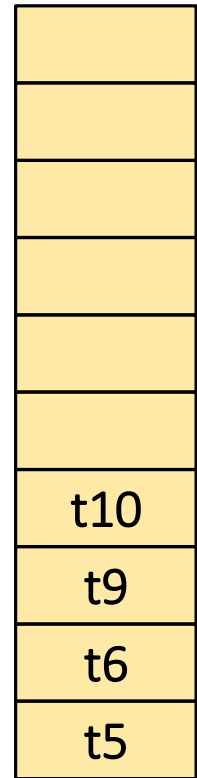
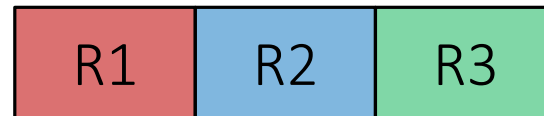
Graph Coloring ($k = 3$)



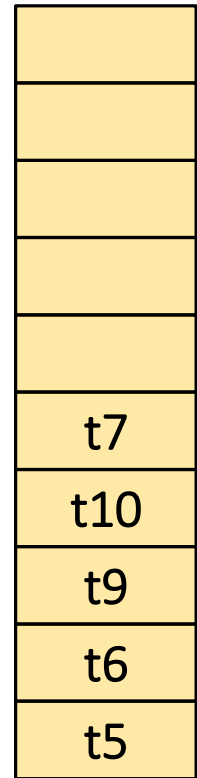
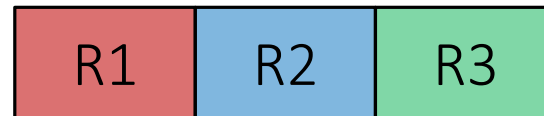
Graph Coloring ($k = 3$)



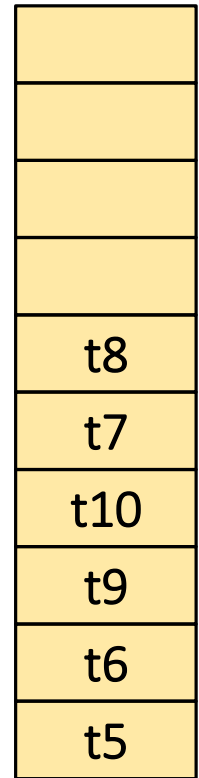
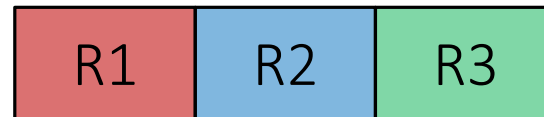
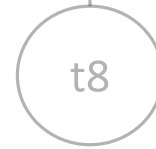
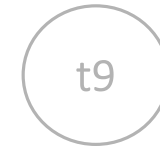
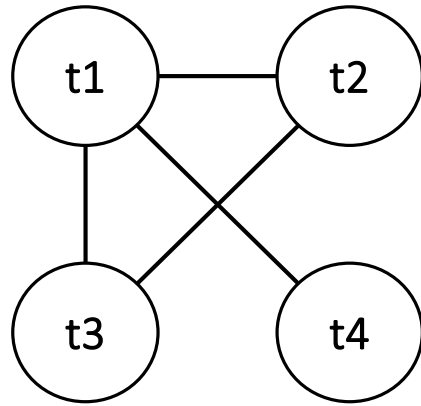
Graph Coloring ($k = 3$)



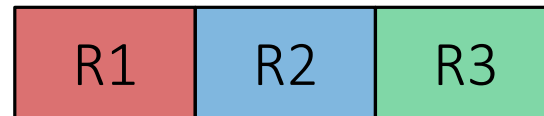
Graph Coloring ($k = 3$)



Graph Coloring ($k = 3$)

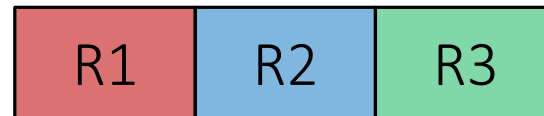


Graph Coloring ($k = 3$)



t4
t8
t7
t10
t9
t6
t5

Graph Coloring ($k = 3$)



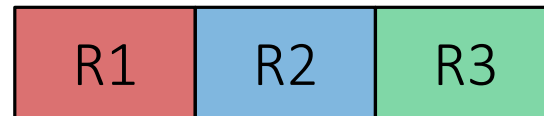
t1
t4
t8
t7
t10
t9
t6
t5

Graph Coloring ($k = 3$)



t2
t1
t4
t8
t7
t10
t9
t6
t5

Graph Coloring ($k = 3$)

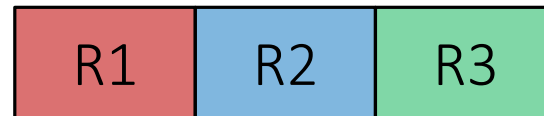


t_3
t_2
t_1
t_4
t_8
t_7
t_{10}
t_9
t_6
t_5

Graph Coloring ($k = 3$)

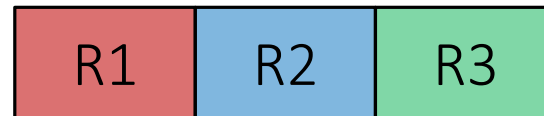
Pop nodes and assign colors...

Graph Coloring ($k = 3$)



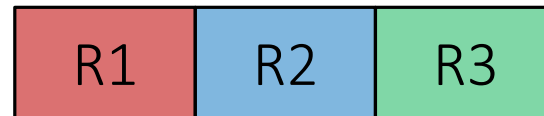
t2
t1
t4
t8
t7
t10
t9
t6
t5

Graph Coloring ($k = 3$)



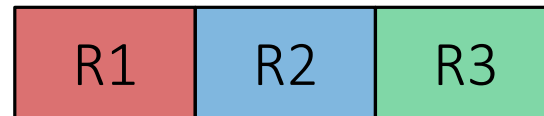
t2
t1
t4
t8
t7
t10
t9
t6
t5

Graph Coloring ($k = 3$)



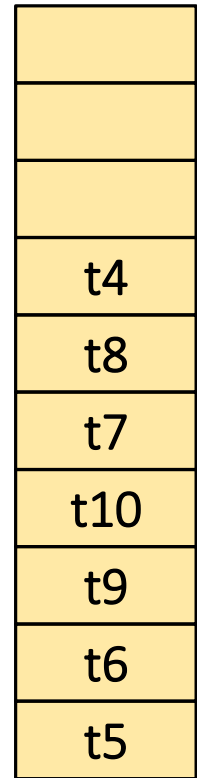
t1
t4
t8
t7
t10
t9
t6
t5

Graph Coloring ($k = 3$)

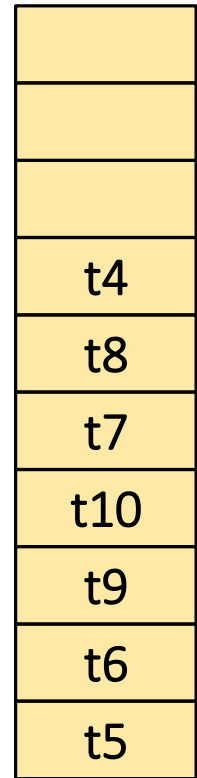
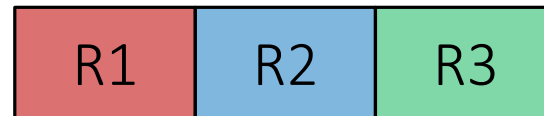


t1
t4
t8
t7
t10
t9
t6
t5

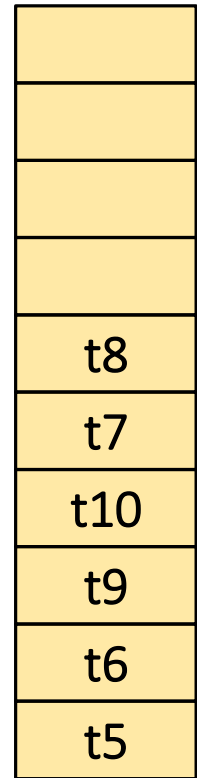
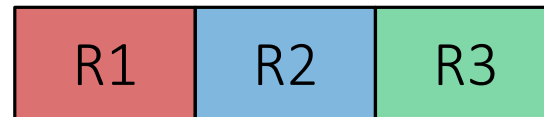
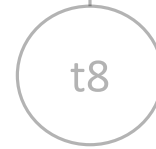
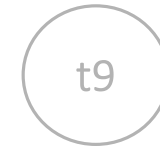
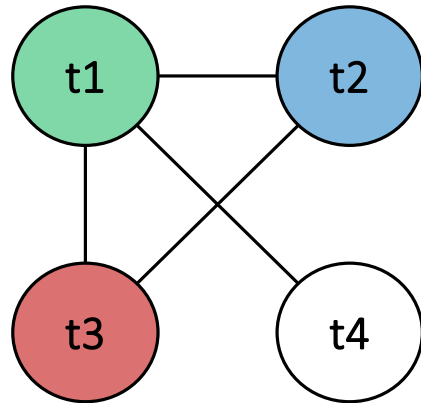
Graph Coloring ($k = 3$)



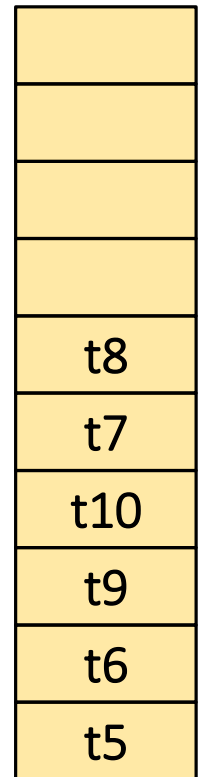
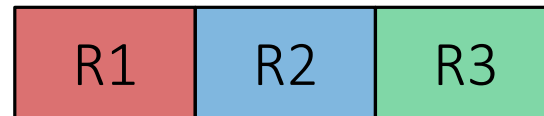
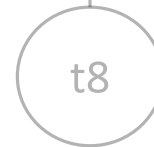
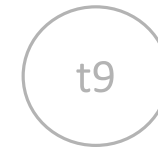
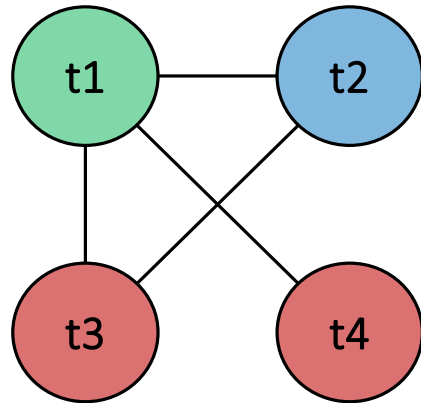
Graph Coloring ($k = 3$)



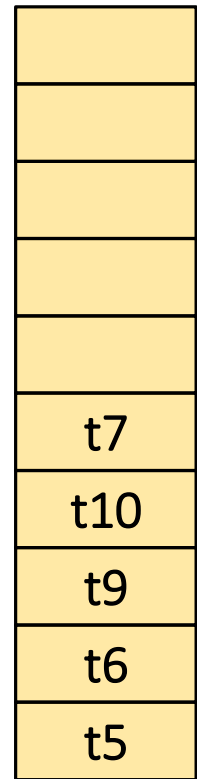
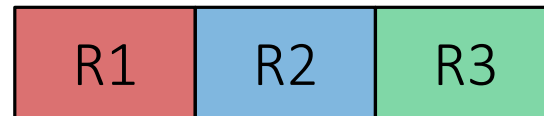
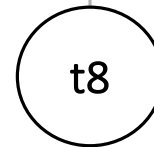
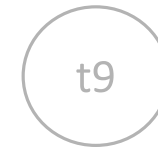
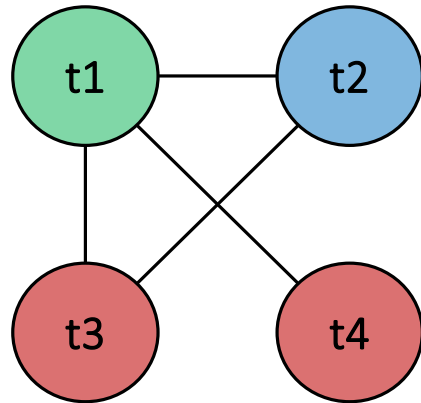
Graph Coloring ($k = 3$)



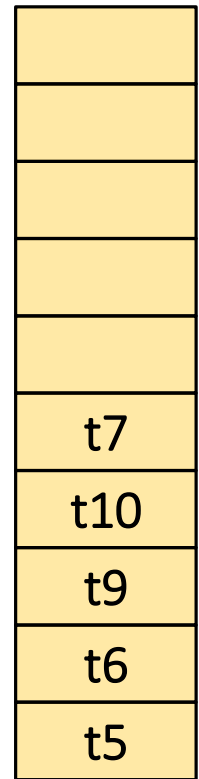
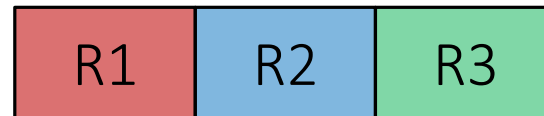
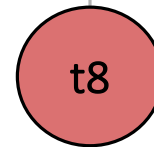
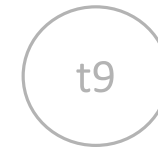
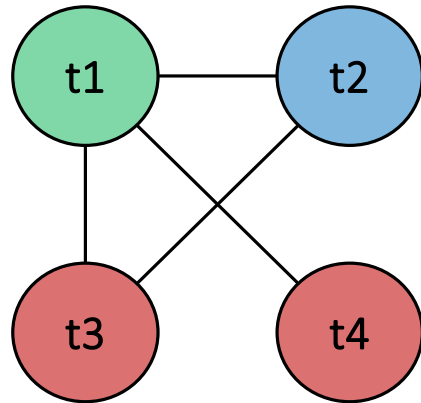
Graph Coloring ($k = 3$)



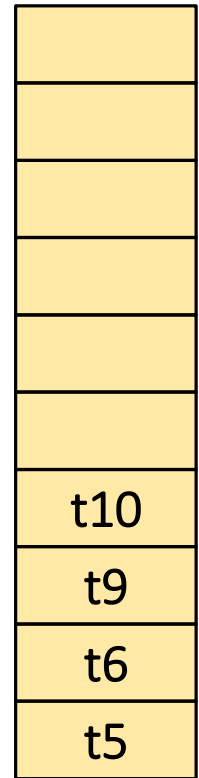
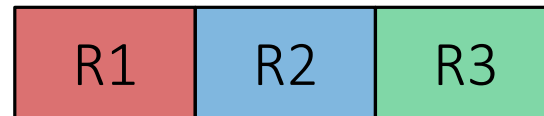
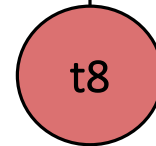
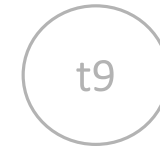
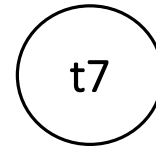
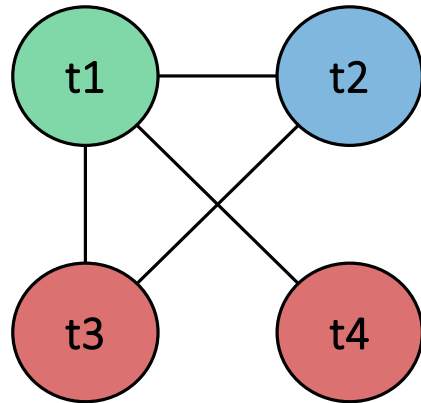
Graph Coloring ($k = 3$)



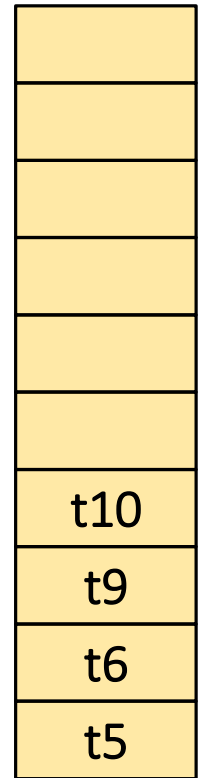
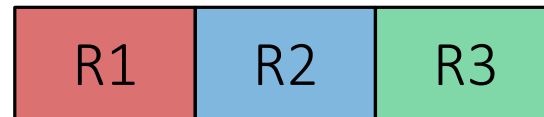
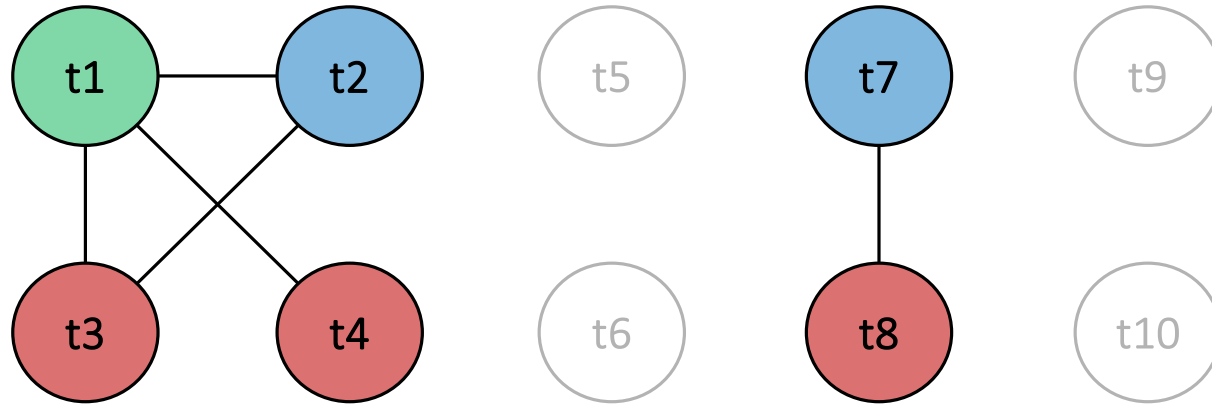
Graph Coloring ($k = 3$)



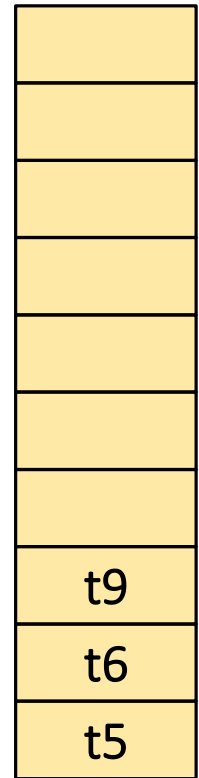
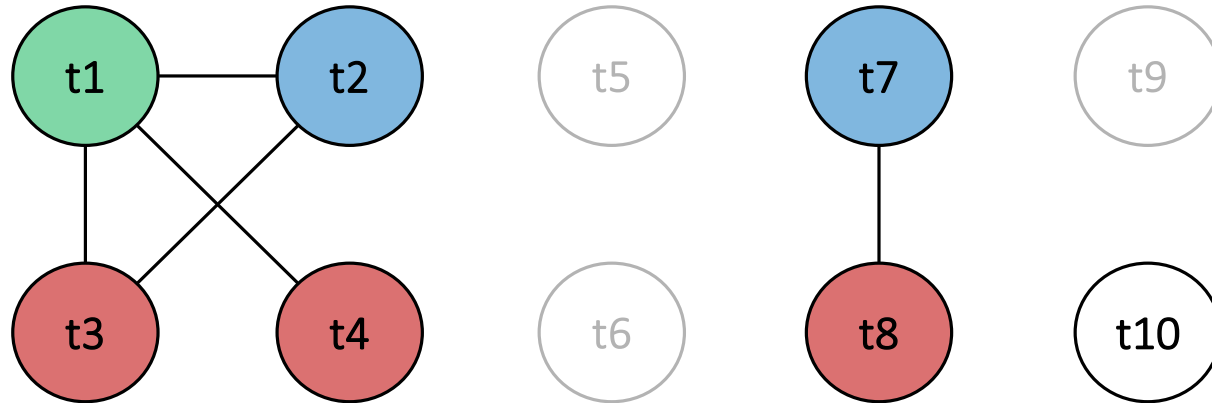
Graph Coloring ($k = 3$)



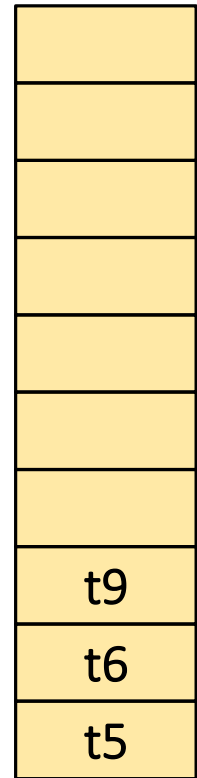
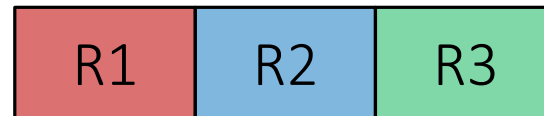
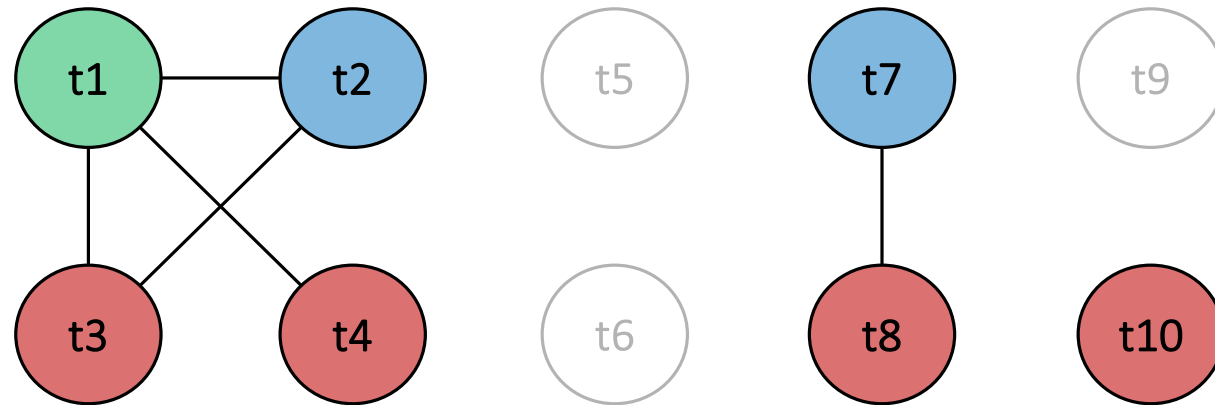
Graph Coloring ($k = 3$)



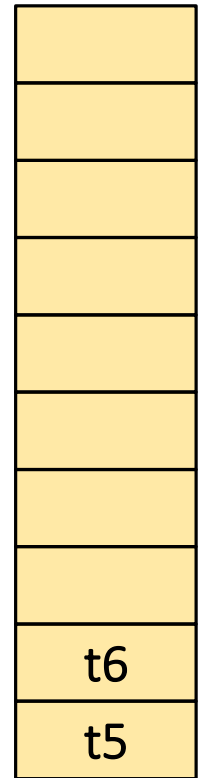
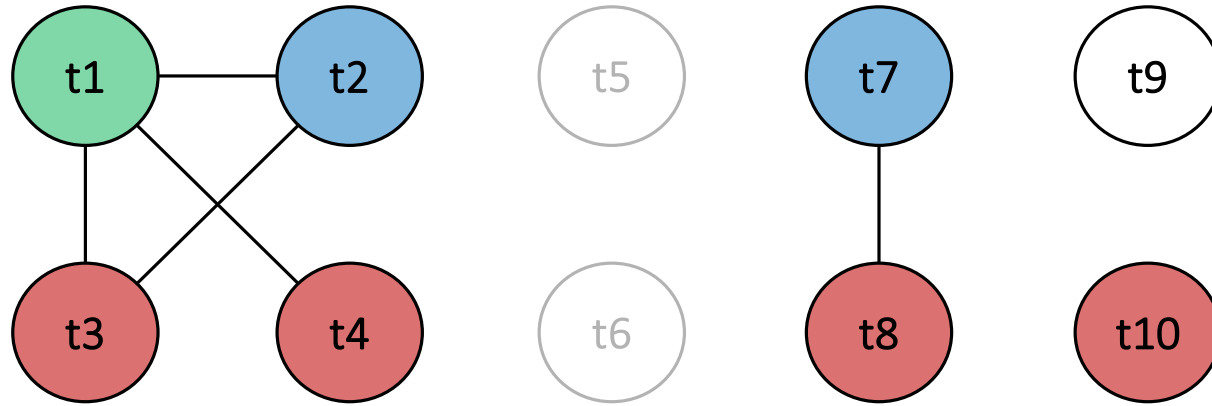
Graph Coloring ($k = 3$)



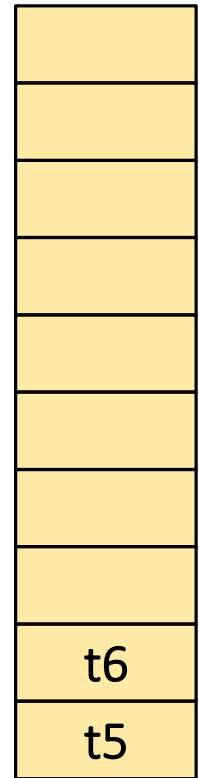
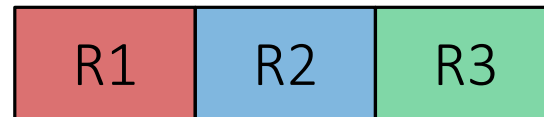
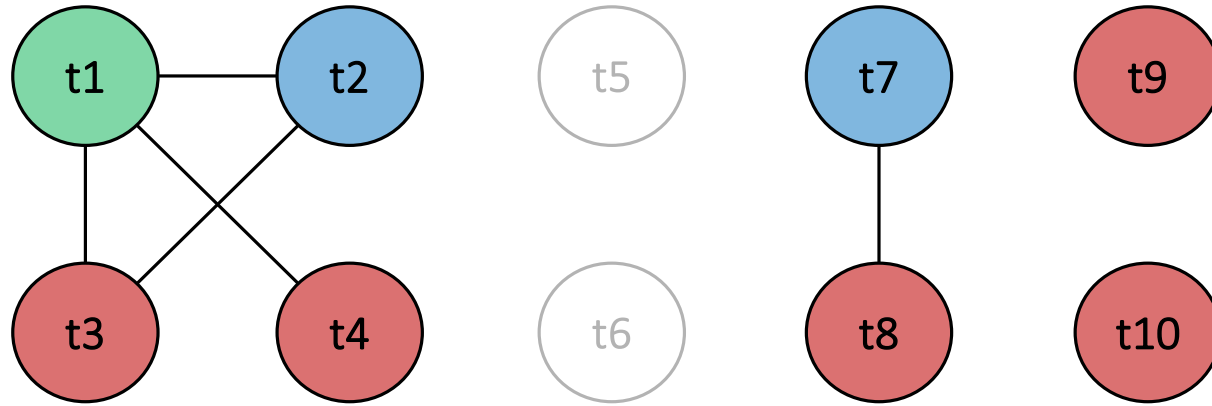
Graph Coloring ($k = 3$)



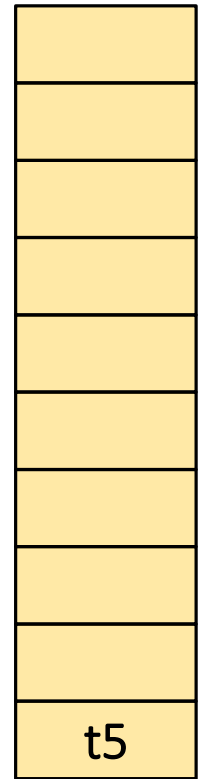
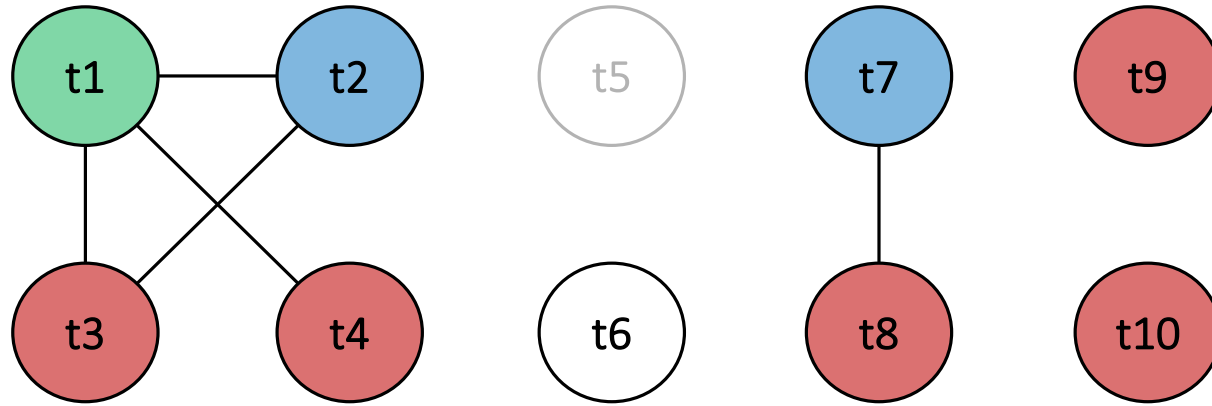
Graph Coloring ($k = 3$)



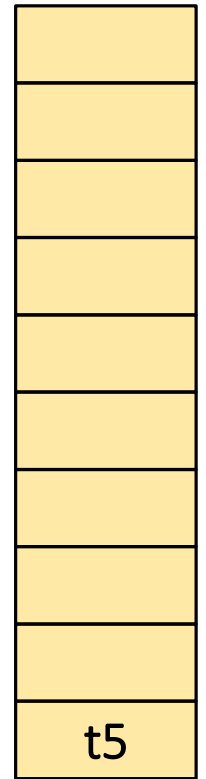
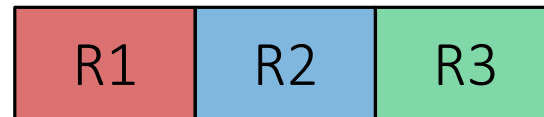
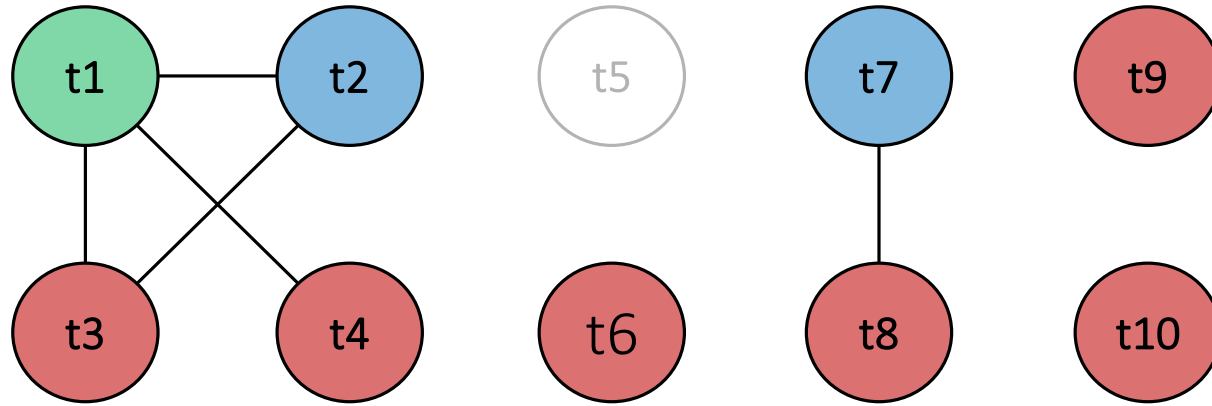
Graph Coloring ($k = 3$)



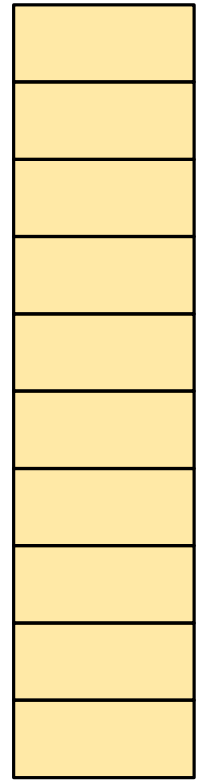
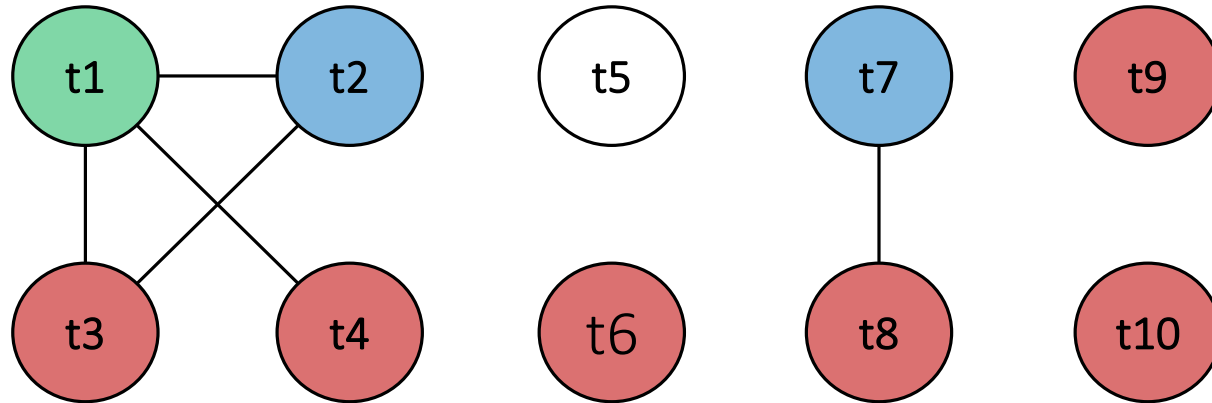
Graph Coloring ($k = 3$)



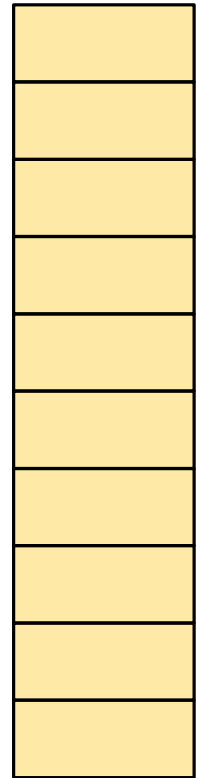
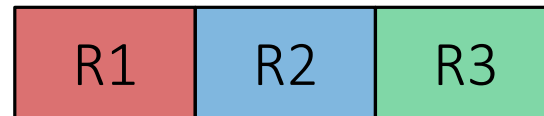
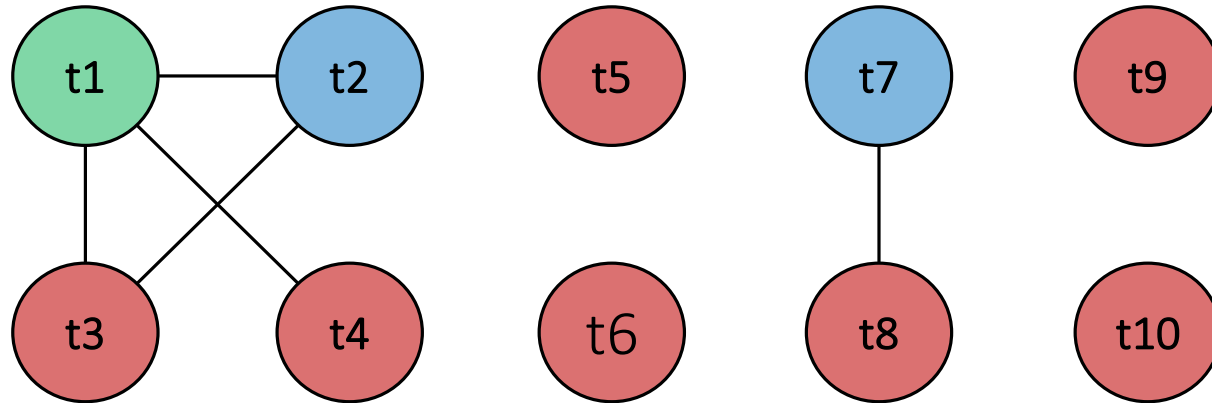
Graph Coloring ($k = 3$)



Graph Coloring ($k = 3$)



Graph Coloring ($k = 3$)



Register allocation

According to the coloring, our register allocation is:

IR Register	Color	MIPS Register
t1	R3	t2
t2	R2	t1
t3	R1	t0
t4	R1	t0
t5	R1	t0
t6	R1	t0
t7	R2	t1
t8	R1	t0
t9	R1	t0
t10	R1	t0

Register allocation

```
t1 = x
t2 = y
t3 = z
t4 = sub t2, t3
t5 = mult, t1, t4
a = t5
t6 = a
bne t6, 1, end
t7 = a
t8 = 1
t9 = add t7, t8
a = t9
end:
t10 = a
b = t10
```

IR Register	MIPS Register
t1	t2
t2	t1
t3	t0
t4	t0
t5	t0
t6	t0
t7	t1
t8	t0
t9	t0
t10	t0

```
lw $t2, 8($fp)
lw $t1, 12($fp)
lw $t0, 16($fp)
sub $t0, $t1, $t0
mul $t0, $t2, $t0
sw $t0, -4($fp)
lw $t0, -4($fp)
bne $t0, 1, end
lw $t1, -4($fp)
li $t0, 1
add $t0, $t1, $t0
sw $t0, -4($fp)
end:
lw $t0, -4($fp)
sw $t0, -8($fp)
```