Series 1

Exercise 1

We consider the following 3-address code sequence:

1. \( a := 1 \)
2. \( b := 2 \)
3. \( e := a+b \)
4. \( d := b\times d \)
5. if \( a+b>0 \) goto 11
6. \( d := c-a \)
7. goto 8
8. \( d := a+b \)
9. \( e := e+1 \)
10. goto 3
11. \( b := a+b \)
12. \( e := c-a \)
13. if \( c > 3 \) goto 3
14. \( c := a+b \)
15. \( b := a-d \)
end

1. Split this sequence into basic blocks, and draw the resulting control flow graph.
2. Give the set of data-flow equations for computing available expressions.
3. Solve these equations.
4. Suppress redundant computations.

Exercise 2

We consider the following program:

\[ \begin{align*}
    b & := 0 \\
\text{while } & d>0 \text{ do } \{ \\
    & a := b+c \\
    & d := d-b \\
    & e := a+f \\
    & \text{if } e > 0 \quad \{ f := a-d ; b := d+f \} \\
    & \quad \text{else} \\
    & \quad \{ e := a-c \} \\
    & \quad b := a+c \\
\} \]

1. Write the 3-address code sequence corresponding to this program.
2. Split this sequence into basic blocks, and draw the resulting control flow graph.
3. Give the set of data-flow equations for computing active variables.
4. Solve these equations.
5. Suppress useless assignments.

**Exercise 3**

We consider the two following CFGs. Modify these CFGs by performing constant folding.

```
x := 5
y := 0
z := x + y
x := z
z := x - y
x := z
t := z + x
```

```
x := 5
y := 1
z := x + y
x := z
z := x - y
x := z
t := z + x
```

**Exercise 4**

Same questions for the following example: