Exercise 1

Def-clear path (another example)

Q1: Find def-clear paths for defs and uses of $x$ and $z$.
Q2: Which definitions are live at node 4?
Def-clear Path

Variable \( x \):

\[
d_1(x) \text{ and } u_2(x): \quad 1\rightarrow 2 \\
d_4(x) \text{ and } u_2(x): \quad 1\rightarrow 2\rightarrow 5 \quad \text{or} \quad 1\rightarrow 2\rightarrow 5\rightarrow 6\rightarrow 2\rightarrow 5 \\
d_6(x) \text{ and } u_2(x): \quad 1\rightarrow 2\rightarrow 3\rightarrow 6 \\
\]

\[
1\rightarrow 2\rightarrow 3\rightarrow 4\rightarrow 6 \\
1\rightarrow 2\rightarrow 5\rightarrow 6
\]

Variable \( z \):

\[
d_1(z) \text{ and } u_4(z): \quad 1\rightarrow 2\rightarrow 3\rightarrow 4 \\
d_1(z) \text{ and } u_6(z): \quad 1\rightarrow 2\rightarrow 3\rightarrow 6 \\
d_4(z) \text{ and } u_6(z): \quad 4\rightarrow 6 \\
d_4(z) \text{ and } u_6(z): \quad 5\rightarrow 6 \\
d_4(z) \text{ and } u_6(z): \quad 4\rightarrow 6\rightarrow 2\rightarrow 3\rightarrow 4 \\
d_5(z) \text{ and } u_6(z): \quad 5\rightarrow 6\rightarrow 2\rightarrow 3\rightarrow 4 \\
d_4(z) \text{ and } u_6(z): \quad 4\rightarrow 6\rightarrow 7 \quad \text{or} \quad 4\rightarrow 6\rightarrow 2\rightarrow 3\rightarrow 6\rightarrow 7 \\
d_6(z) \text{ and } u_6(z): \quad 5\rightarrow 6\rightarrow 7 \\
d_5(z) \text{ and } u_6(z): \quad 1\rightarrow 2\rightarrow 3\rightarrow 6\rightarrow 7
\]

If loops are considered, there can be more def-clear paths.
begin
float x, y, z=0.0;
int count;
input(x, y, count);
begin
if (x<10) {
    if (y>0) {
        x=x+1;
    }
} else {
    z=1;
}
y=x*y+z
count=count-1
while (count>0) 
    output(x);
end

d1(x, y, count)
(c, does not go through the loop)

Node Lines
1  1, 2, 3, 4
2  5, 6
3  7
4  8, 9, 10
5  11, 12, 13
6  14, 15, 16
7  17, 18

d1(x, y, count)
(going through the loop once)

Node Lines
1  1, 2, 3, 4
2  5, 6
3  7
4  8, 9, 10
5  11, 12, 13
6  14, 15, 16
7  17, 18
begin
float x, y, z=0.0;
int count;
input(x, y, count);
do{
  if (x<0)
    if (y>0)
      z=y++;
  }
else{
  z=x++;
}
count=count-1;
while (count>0)
output(z);
end

begin
float x, y, z=0.0;
int count;
input(x, y, count);
do{
  if (x<0)
    if (y>0)
      z=x++;
  }
else{
  z=1/x;
}
count=count-1;
while (count>0)
output(z);
end
begin
float x, y, z = 0.0;
int count;
input (x, y, count);
do
if (x<0) {
if (y>0) {
    y=y*z+1;
} else{
    z=1/z;
}
}
while (count>0)
output (z);
end

begin
float x, y, z = 0.0;
int count;
input (x, y, count);
do
if (x<0) {
if (y>0) {
    y=y*z+1;
} else{
    z=1/z;
}
}
while (count>0)
output (z);
end
begin
float x, y, z=0.0;
int count;
input [x, y, count];
do {
  if (x>0) {
    if (y>0) {
      x=x+1;
    }
  }
  else{
    x=1/x;
  }
  y=x*y-z
  count=count-1
} while (count>0)
output (z);
end

begin
float x, y, z=0.0;
int count;
input [x, y, count];
do {
  if (x>0) {
    if (y>0) {
      x=x+1;
    }
  }
  else{
    x=1/x;
  }
  y=x*y-z
  count=count-1
  count=0
} while (count>0)
output (z);
end
begin
float x, y, z=0.0;
int count;
input (x, y, count);
do {
    if (x<=0) {
        if (y>=0) {
            y=y+z;
        } else {
            y=1/x;
        }
    } else {
        y=x^2+1;
    }
} while (count>0)
output (z);
end
begin
    float x, y, z=0.0;
    int count;
    input (x, y, count);
    do
        if (x<0) {
            y=y+1;
        }
        else {
            y=1/x;
        }
        count=count-1;
        while (count>0)
    output (z);
Which definition are live at node 4?
- Variable y: $d_4(y)$ and $d_5(y)$
- Variable z: $d_1(z)$ and $d_4(z)$
Exercise 2

Def-use pairs (example)

Variable \( z \): (5, 4)
Variable count: (1, (6,2)), and (1, (6,7))
are infeasible.

<table>
<thead>
<tr>
<th>Variable (v)</th>
<th>Defined in node (n)</th>
<th>defu (v, n)</th>
<th>dpui (v, n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
<td>{5, 6}</td>
<td>{(2, 3), (2, 5)}</td>
</tr>
<tr>
<td>y</td>
<td>1</td>
<td>{4, 6}</td>
<td>{(3, 4), (3, 6)}</td>
</tr>
<tr>
<td>z</td>
<td>5</td>
<td>{4, 6, 7}</td>
<td>{}</td>
</tr>
<tr>
<td>count</td>
<td>6</td>
<td>{6}</td>
<td>{(8, 2), (6, 7)}</td>
</tr>
</tbody>
</table>
### Code 1

```c
begin
float x, y, z=0.0;
int count;
input (x, y, count);
do {
    if (x<0) {
        if (y>0) {
            z=x*y+1;
        }
    } else{
        z=1/x;
    }
y=xy;z+
count=count-1;
while (count>0)
output (z);
}
def=(x,y,z,count)
c-use[z];
```

### Node Lines

- Line 1: 1, 2, 3, 4
- Line 2: 5, 6
- Line 3: 7
- Line 4: 8, 9, 10
- Line 5: 11, 12, 13
- Line 6: 14, 15, 16
- Line 7: 17, 18

### Code 2

```c
begin
float x, y, z=0.0;
int count;
input (x, y, count);
do {
    if (x<0) {
        if (y>0) {
            z=x*y+1;
        }
    } else{
        z=1/x;
    }
y=xy;z+
count=count-1;
while (count>0)
output (z);
}
def=(x,y,z,count)
c-use[z];
```

### Node Lines

- Line 1: 1, 2, 3, 4
- Line 2: 5, 6
- Line 3: 7
- Line 4: 8, 9, 10
- Line 5: 11, 12, 13
- Line 6: 14, 15, 16
- Line 7: 17, 18
1: begin
2: float x, y, z=0.0;
3: int count;
4: input (x, y, count);
5: do {
6:   if (x<=0) {
7:     if (y>=0) {
8:       z = y^2 + 1;
9:     }
10:   }
11: else {
12:     x = 1/x;
13:   }
14:   y = y + z;
15:   count = count - 1;
16:   while (count > 0)
17:   output (z);
18: }

19: begin
20: float x, y, z=0.0;
21: int count;
22: input (x, y, count);
23: do {
24:   if (x<=0) {
25:     if (y>=0) {
26:       z = y^2 + 1;
27:     }
28:   }
29: else {
30:     x = 1/x;
31:   }
32:   y = y + z;
33:   count = count - 1;
34:   while (count > 0)
35:   output (z);
36: }

37: Write the definition of y at ③
38: Write the definition of x + ①
begin
float x, y; x=0.0;
int count;
input (x, y, count);
do {
if (x<0) {
if (y>0) {
    z=y=x+1;
}
}
else {
    x=1/x;
}
y=x+y;
count=count-1;
while (count>0)
output (z);
end
begin
float x, y, z=0.0;
int count;
input (x, y, count);
do {
    if (x<=0) {
        x=x+y+1;
    } else{
        x=1/x;
    }
} count=count-1
while (count>0)
output (z);
Exercise 3

Def-use pairs: Minimal set (2)

What will be also covered if we have a test case which covers 
\((d_i(z), u_i(z))\)?

How about \((d_i(z), u_i(z))\)?
$$\mathbf{d}_1(z), \mathbf{u}_4(z)$$

begin
  float x, y, z=0.0;
  int count;
  input (x, y, count);
  do {
    if (x<0) {
      if (y>0) {
        z=y*x+z;
      }
      count=count-1;
      while (count>0)
        output (x);
    } else{
      x=x/z;
    }
    y=x*y+z;
  }
end
Select an execution path that covers the specific def-use pair
Identify all the def-use pairs in the selected path
For each recognized def-use pair \((d_i(x), u_j(x))\)
  - Try to find a path from node \(I\) to node \(j\) that is not def-clear path for variable \(x\)
  - If so, remove the def-use pair \((d_i(x), u_j(x))\)
  - If not, keep it

Consider execution path 1->2->3->4->6->7, its covered def-use pairs are
  - Variable \(x\): \((1, (2, 3))\) \((1, 6)\)
  - Variable \(y\): \((1, 4)\) \((1, 6)\) \((1, (3, 4))\)
  - Variable \(z\): \((1, 4)\) \((4, 6)\) \((4, 7)\)
  - Variable \(count\): \((1, 6)\) \((6, (6, 7))\)
  - Def-use pairs for variable \(y\), \((1, 4)\) and \((1, (3, 4))\) can be removed because variable \(y\) is re-defined at node 6 in an alternative execution path 1->2->3->6->2->3->4->6->7
Exercise 4

C-use coverage

C-use coverage:
The c-use coverage of $T$ with respect to $(P, R)$ is computed as

$$\frac{CU}{CU - CU_f}$$

where $CU_c$ is the number of c-uses covered and $CU_f$ the number of infeasible c-uses. $T$ is considered adequate with respect to the c-use coverage criterion if its c-use coverage is 1.
C-use coverage: path traversed

Path $(\text{Start, } q, k, \ldots, z, \ldots, \text{End})$ covers the c-use at node $z$ of $x$ defined at node $q$ given that $(k, \ldots, z)$ is def clear with respect to $x$.

**Exercise:** Find the c-use coverage when program P14.16 (refer to slide 101) is executed against the following test:

$t_i$: <$x=-5, y=-1, count=1$>
covered def-c-use pairs with respect to $t_1$

- Variable $x$: (1, 5) (1, 6)
- Variable $y$: (1, 6)
- Variable $z$: (5, 6) (5, 7)
- Variable $count$: (1, 6)

Exercise 5
**p-use coverage**

**P-use coverage:**

The p-use coverage of $T$ with respect to $(P, R)$ is computed as

$$\frac{PU_c}{(PU - PU_f)}$$

where $PU_c$ is the number of p-uses covered and $PU_f$ the number of infeasible p-uses. $T$ is considered adequate with respect to the p-use coverage criterion if its p-use coverage is 1.

---

**Exercise:** Find the p-use coverage when program P14.16 (refer to slide 101) is executed against test $t_2$: \(<x=-2, y=-1, \text{count}=3>\)
```plaintext
begin
float x, y, z=0;
int count;
input (x, y, count);
do {
  if (x<0) {
    y=x+y;
  } else {
    x=x+y;
  }
  count=1
} while (count>0)
output (x);
end

t_1: <x=-2, y=1, count=3>
```
covered def-p-use pairs with respect to $t_2$

- Variable $x$: (1, (2, 3))
- Variable $y$: (1, (3, 6)) (6, (3, 4)) (6, (3, 6))
- Variable $\text{count}$: (6, (6, 2)) (6, (6, 7))
Exercise 6

All-uses coverage

All-uses coverage:

The all-uses coverage of $T$ with respect to $(P, R)$ is computed as

$$\frac{(CU_c + PU_c)}{((CU + PU) - (CU_f + PU_f))}$$

where $CU$ is the total c-uses, $CU_c$ is the number of c-uses covered, $PU_c$ is the number of p-uses covered, $CU_f$ the number of infeasible c-uses and $PU_f$ the number of infeasible p-uses. $T$ is considered adequate with respect to the all-uses coverage criterion if its c-use coverage is 1.

Exercise: Is $T = \{t_1, t_2\}$ adequate w.r.t. to all-uses coverage for P14.16?
covered def-use pairs with respect to $t_1$

- Variable $x$: (1, 5) (1, 6) (1, (2, 5))
- Variable $y$: (1, 6)
- Variable $z$: (5, 6) (5, 7)
- Variable $count$: (1, 6) (6, (6, 7))

covered def-use pairs with respect to $t_2$

- Variable $x$: (1, 6)
  (1, (2, 3))
- Variable $y$: (1, 6) (6, 4) (6, 6)
  (1, (3, 6)) (6, (3, 4)) (6, (3, 6))
- Variable $z$: (1, 4) (1, 6) (4, 6) (4, 7)
- Variable $count$: (1, 6) (6, 6)
  (6, (6, 2)) (6, (6, 7))
No, four def-use pairs are not covered with respect to $t_1$ and $t_2$, i.e.,
- Variable $y$: (1, 4) and (1, (3, 4))
- Variable $z$: (1, 7) and (4, 4)

In addition, there are three infeasible def-use pairs. That is,
- Variable $z$: (5, 4)
- Variable $count$: (1, (6, 2)) (1, (6, 7))

All use coverage is $\frac{20}{27 - 3} = 0.833$