Principles of Software: Homework #1

Due on February 19 2021

Prof. Valera

Jared Gridley

Problem 1

`Indicate the strongest condition in each set. Write "None" if the conditions are unrelated by implication. Unless otherwise stated, assume all variables are ints.`

1. $\{x \; is \; even \bigwedge y = x + 1\}$
2. $\{1 \leq x \leq 3\}$
3. $\{x > 0 \bigwedge y > 0\}$
4. {x is divisible by 50 }
5. None
6. abs(result*result - x) $\leq$ 0.000001

Problem 2

`State whether each Hoare triple is valid. If it is invalid, explain why and show how you would modify the postcondition to make it valid.`
`Unless otherwise stated, assume all variables are ints.`

1. Valid
2. Valid
3. Invalid - The precondition and post condition contradict each other. The operations inside are preserving the sum (subtracting 1 for j and adding 1 to i). So to make this valid, we would modify the postcondition to be i + j $\neq$ 0.
4. Invalid - This does not account for the case when x = y. It would hit the else statement but then fail the postcondition. To fix this we just alter the second part of the postcondition to be:
   $(m = x \bigwedge x > y) \bigvee (m = y \bigwedge x \leq y)$

Problem 3

A, B, C, D, E, F are logical conditions (logical formulas).
The following are true:
B => D (B implies D, i.e., B is stronger than D)
C => D
D => F
A => E
{F} code {B}
Indicate whether the following are valid or possibly invalid.

1. {A} code {E}

This is valid. Since A implies E from the statements above, it means that A is stronger than E. That is, the conditions for E are included in A. Since A contains at least the conditions in E, then this statement will be valid

2. {C} code {D}

This is also valid. Since it is given above that C => D, it means that C is stronger than D, and that the conditions for C at least include those in D. Thus, the statement must be valid as any code that works for C will also work for D.

Problem 4

Find the strongest postcondition of each code sequence by inserting the appropriate condition in each blank. The first condition in part (a) is supplied as an example. Please simplify your answers as much as possible. Assume all variables are ints. Copy all code to your answer file and fill in the blanks. Carry all variables forward. Show all work.

```
1. { x > 0 }
      x = 10;
{ x = 10 }
      y = 20 - x;
            {y = 10}
      z = y + 4;
            {z = 10}
      y = 0;
            {y = 0}
2. { |x| > 11 }
      x = -x;
            {-11 < x  x > 11}
      x = x * x;
            {x > 121}
      x = x + 1;
            {x > 144}
3. { |x| < 5 }
      if (x > 0) {
                {0 < x < 5}
           y = x + 2;
                {2 < y < 7}
       } else {
                {-4 <= x <= 0}
           y = x - 1;
                {-6 < y < 0}
            {|x| < 5}
```

Problem 5

Find the weakest precondition of each code sequence by inserting the appropriate condition in each blank. Simplify your answers as much as possible. Assume all variables are ints. The first condition in part (1) is supplied as an example.

1.
```
{y > -2x}
x = -5;
        { wp(z = 2 * x + y, z > 0) = (2x + y > 0) = (y > -2x) }
z = 2 * x + y;
 {z>0}
```

2.
```
{x >= 2}
if (x > 0) {
    {wp(x = x + 6, x > 7) = x >= 2}
   x = x + 6;
else {
    {wp(x = 4 - x, x > 7) = x <= -4}
   x = 4 - x;
 }
    {x>7}
```

3.
```
{x >= 0}
if (x > 4) {
         {wp(x = x - 3, x > 4}
    x = x - 3;
 } else {
         {wp(x <= 4, x > 0) = (x >= 0)}
    if (x < - 4) {
        {wp(x = x + 3, x > 0) = x >= -2}
       x = x + 3;
    } else {
        {wp( x = x + 1, x > 0 ) = x >= 0}
       x = x + 1;
    }
 }
    {x>0}
```

4.
```
{ wp( x = y + 2, z > 2y ) = y <= 2 }
  x = y + 2;
   {wp(z = x + 1, z > 2y) = x <= 4 }
  z = x + 1;
     { z > 2y }
```

```
5.
 {x != 1  x != 0}
 if (x >= 0)
   {wp( z = x, z != 0) = x != 0 }
    z = x;
 else
   {wp( z = x + 1, z != 0) = x != 1}
    z = x + 1;
    {z != 0}
```

Problem 6

For each block of code, fill in the intermediate conditions, then use them to state whether the precondition is sufficient to guarantee the postcondition. If the precondition is insufficient, explain why.
Hint: Use backward reasoning to find the weakest precondition that guarantees the postcondition and see if the given precondition is too weak to guarantee the postcondition. In other words, is the given precondition weaker than the weakest precondition?

1.

```
    {xpre < ypre }
       {wp( w = y, xpost = ypre  ypost = xpre) = xpre = ypre}
    w = y;
       {wp( z = x + y - w, xpost = ypre  ypost = xpre) = z=x}
    z = x + y - w;
       {wp( x = w, xpost = ypre  ypost = xpre) = w=y}
    x = w;
        {xpost = ypre  ypost = xpre}
```

Sufficient or Insufficient: Insufficient. The post condition will not hold when xpre < ypre because we do not alter xpre or ypost so if they are different values then the condition will fail. It should be xpre = ypre.

2.

```
{(x = y)  (x != y  y > 0)}
        {wp(x = y  x != y, x <= y) = ((x = y)  (x <= y/x  y > 0))}
    if (x == y)
        {wp(x = 0, x <= y) = (y = 0)}
        x = 0;
    else
        {wp(x = x * y, x <= y) = (x <= y/x)}
        x = x * y;
    {x <= y}
```

Sufficient or Insufficient: Insufficient. The precondition is insufficient because it fails to account for the case where x > y. It x > y then it will hit the else statement and then post condition will fail. To fix this, you would need to add x < y or to be more specific x <= y/x, so that when it comes to the else statements, the post condition will still hold.