CSCI 2600 Principles of Software

Homework 1: Reasoning About Code

Due: February 19 @ 11:59pm EST

Submission Instructions

- This homework is different from the previous homework. It does not involve any Java coding. Rather, you will answer some questions involving reasoning about code.
- Follow the directions in the version control handout for cloning your hw01 git repo.
- Submit your answers in a single .PDF file named `hw1_answers.pdf` in the `answers/` directory of your repository.
  You MUST type up your answers. Handwritten solutions will not be accepted or graded, even if they are scanned into a PDF file.
  We recommend using LaTeX. If you have never used LaTeX, take a look at this tutorial.
- Be sure to commit and push the file to Submitty. Follow the directions in the version control handout for adding and committing files.
- You must click the **Grade My Repository** button for your answers to be graded. If you do not, they will not be graded and you will receive a zero for this homework.

Directions:

- Unless otherwise indicated, assume that all numerical variables are integers, and that integer overflow or underflow will not occur.
- In your answers, we strongly recommend that you use standard logical symbols ∧ and ∨ for "and" and "or". We recommend you use = for equivalence of logical statements.
- If no precondition is required for a code sequence, simply write {true} to denote the trivial precondition.

Problems

Problem 1 (6 pts, 1 pt each): Condition Strength

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 ∧ y = x + 1 }
{ x is even ∧ y is odd }

2. { 0 ≤ x ≤ 5 }
{ 0 ≤ x ≤ 3 }
{1 ≤ x ≤ 3 }
```

3. { x > 0 ∧ y > 0 }
{ x > 0 ∨ y > 0 }

4. {x is divisible by 50 }
   {x is divisible by 5 }

5. { -5 < x < 40 }
   { 10 < x < 50 }

6. abs(result*result - x) ≤ 0.0001 assume result is a double
abs(result*result - x) ≤ 0.000001

Problem 2 (8 pts, 2 pts each): Hoare Triples

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.  { x > 0 }
    y = -2 * x;
    { y ≤ 0 }

2.  { x + 1 ≤ N }
    x = x + 1
    { x ≤ N }

3. { i + j ≠ 0 }
    i = i + 1;
    j = j - 1;
    { i + j = 0 }

4. { true }
    if (x > y)
        m = x;
    else
        m = y;
    { ( m = x ∧ x > y ) ∨ ( m = y ∧ x < y ) }
```

Problem 3 (2 pts, 1 pt each): General Hoare Triples

```
   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}
```

2. {C} code {D}

Problem 4 (11 pts, 1 pt. for each condition): Forward reasoning

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;
             {_____________________________________}
       z = y + 4;
             {_____________________________________}
       y = 0;
             {_____________________________________}

2.
3. { |x| > 11 }
       x = -x;
             {_____________________________________}
       x = x * x;
             {_____________________________________}
       x = x + 1;
             {_____________________________________}


3. { |x| < 5 }
       if (x > 0) {
                  {_____________________________________}
            y = x + 2;
                  {_____________________________________}
        } else {
                  {_____________________________________}
            y = x - 1;
                  {_____________________________________}
        }
             {_____________________________________}
```

Problem 5 (14 pts, 1 pt each condition): Backward reasoning

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.

Copy all code to your answer file and fill in the blanks. Answers must be expressed in the format `wp(expression, condition) = precondition`.
Show all work.

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

2.  {_____________________________}
  if (x > 0) {
       {_____________________________}
     x = x + 6;
  else {
       {_____________________________}
     x = 4 - x;
  }
       { x > 7 }

3.    {_____________________________}
  if (x > 4) {
              {_____________________________}
            x = x -3;
  } else {
                        {_____________________________}
            if (x < - 4) {
                        {_____________________________}
                    x = x + 3;
            } else {
                        {_____________________________}
                    x = x + 1;
            }
  }
             { x > 0 }

 4.  {_____________________________}
    x = y + 2;
      {_____________________________}
    z = x + 1;
      { z > 2y }

 5.
  {_____________________________}
  if (x >= 0)
    {_____________________________}
     z = x;
  else
    {_____________________________}
     z = x + 1;
  { z ≠ 0 }
```

Problem 6 (8 pts, 1 pt each condition, 1 pt sufficient/insufficient): Verifying Correctness

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?

Remember that if a variable is assigned to somewhere in the code, you might need to use some notation to distinguish between the values before and after assignment, e.g., y_{pre} and y_{post} (or y_0, y_1, …). If a variable is never assigned to, its "pre" and "post" values are guaranteed to be the same, i.e., for y it would be $y = y_{pre} = y_{post}$. If subscripts are not given, any variable in the precondition assumes the value "pre", any variable in the postcodition assumes the value "post".

Copy all code to your answer file and fill in the blanks. Answers must be expressed in the format `wp(expression, condition) = precondition`. Show all work. Assume all variables are integers.

```
1.     { x_pre < y_pre }
       {_____________________________}
   w = y;
       {_____________________________}
   z = x + y - w;
       {_____________________________}
   x = w;
       { x_post = y_pre ∧ y_post = x_pre }
Sufficient or Insufficient: _____________________________

2. { (x = y) ∨ (x ≠ y ∧ y > 0) }
       {_____________________________}
   if (x == y)
       {_____________________________}
       x = 0;
   else
       {_____________________________}
       x = x * y;
   { x ≤ y }
Sufficient or Insufficient: _____________________________
```

Collaboration (0.5 pts)

Please answer the following questions in a file named `collaboration.pdf` in your `hw1/answers/` directory.

The standard academic integrity policy applies to this homework.

State whether or not you collaborated with other students. If you did collaborate with other students, state their names and a brief description of how you collaborated.

Reflection (0.5 pts)

Please answer the following questions in a file named `reflection.pdf` in your `hw1/answers/` directory. Answer briefly, but in enough detail to help you improve your own practice via introspection and to enable us to improve Principles of Software in the future.

1. In retrospect, what could you have done better to reduce the time you spent solving this homework?
2. What could we (the instructors, TAs, and mentors) have done better to improve your learning experience in this homework?
3. What do you know now that you did not know before beginning the homework?

Errata

Check the Submitty Forum for possible errata.