Psoft HW1 Answers

Temitayo Oladeji

February 1st

Problem 1: Condition Strength

Part (a): Evaluating Implication Chains

For each chain, we will assess whether the implications hold from right to left. If any implication is false, we will provide a counterexample.

Chain 1:

$\{x \text{ is a Monday}\} \leftarrow \{x \text{ is a day in January 2025}\} \leftarrow \{x \text{ is a Monday in January}\} \leftarrow \{x \text{ is Monday, Januar}$

Evaluation:

- $\{x \text{ is a Monday}\} \leftarrow \{x \text{ is a day in January 2025}\}$: False. Not all days in January 2025 are Mondays. For example, January 1, 2025, is a Wednesday.
- $\{x \text{ is a day in January 2025}\} \leftarrow \{x \text{ is a Monday in January}\}$: True. If x is a Monday in January, then it is indeed a day in January 2025.
- $\{x \text{ is a Monday in January}\} \leftarrow \{x \text{ is Monday, January 27, 2025}\}$: True. January 27, 2025, is a Monday in January.

Conclusion: The chain is false due to the first implication.

Chain 2:

$\{u = 10k \wedge v = y+7 \wedge u+v = 7k\} \leftarrow \{10x+y \text{ is divisible by } 7\} \leftarrow \{x-2y \text{ is divisible by } 7 \vee x-2y = 0\} \leftarrow \{x =$

Evaluation:

- $\{u = 10k \wedge v = y + 7 \wedge u + v = 7k\} \leftarrow \{10x + y \text{ is divisible by } 7\}$: False. Not all numbers of the form $10x + y$ divisible by 7 can be expressed as $u = 10k$ and $v = y + 7$ such that $u + v = 7k$.
- $\{10x+y \text{ is divisible by } 7\} \leftarrow \{x-2y \text{ is divisible by } 7 \vee x-2y = 0\}$: False. Consider $x = 2$ and $y = 1$. Here, $x - 2y = 0$, but $10x + y = 21$, which is divisible by 7. However, for $x = 3$ and $y = 1$, $x - 2y = 1$, which is not divisible by 7.

- $\{x - 2y \text{ is divisible by } 7 \vee x - 2y = 0\} \leftarrow \{x = 60 \wedge y = 9\}$: True. For $x = 60$ and $y = 9$, $x - 2y = 42$, which is divisible by 7.
- $\{x = 60 \wedge y = 9\} \leftarrow \{10x + y = 609\}$: True. If $10x + y = 609$, then solving for x and y gives $x = 60$ and $y = 9$.

Conclusion: The chain is false due to the first and second implications.

Chain 3:

$$\{\text{true}\} \leftarrow \{|x| = x\} \leftarrow \{x > 0\} \leftarrow \{10 < x < 50 \wedge x < 0\} \leftarrow \{\text{false}\}$$

Evaluation:

- $\{\text{true}\} \leftarrow \{|x| = x\}$: True. The condition $|x| = x$ is always true for all x.
- $\{|x| = x\} \leftarrow \{x > 0\}$: True. If $x > 0$, then $|x| = x$.
- $\{x > 0\} \leftarrow \{10 < x < 50 \wedge x < 0\}$: False. The condition $10 < x < 50 \wedge x < 0$ is contradictory, as x cannot be both greater than 10 and less than 0 simultaneously.
- $\{10 < x < 50 \wedge x < 0\} \leftarrow \{\text{false}\}$: True. The condition false implies any condition, including a contradictory one.

Conclusion: The chain is false due to the third implication.

Part (b): Ordering Conditions by Strength

We will order the given conditions from weakest to strongest by establishing implication relationships. If a total ordering is not possible, we will provide a partial ordering.

Set (1):

$$\{6 \leq k < 6 \wedge k = 6\}, \{\text{false}\}, \{k = 6n\}, \{12 \leq k \leq 12\}$$

Analysis:

- Condition 1: $6 \leq k < 6 \wedge k = 6$ is false for any integer k.
- Condition 2: false is universally false.
- Condition 3: $k = 6n$ means k is a multiple of 6.
- Condition 4: $12 \leq k \leq 12$ simplifies to $k = 12$.

Implications:

- $\{\text{false}\} \leftarrow \{\text{false}\}$: True.
- $\{\text{false}\} \leftarrow \{k = 6n\}$: False. Counterexample: $k = 6$.

- $\{k = 6n\} \leftarrow \{k = 12\}$: True. Since 12 is a multiple of 6.

Ordering: The valid implication chain is:

$$\{k = 12\} \leftarrow \{k = 6n\}$$

Conditions {false} and $\{6 \leq k < 6 \wedge k = 6\}$ are equivalent and represent the weakest condition.

Set (2):

$\{x = 3k+3 \wedge x \text{ is even}\}, \{x \text{ is divisible by } 6\}, \{y = \text{sum of digits of } x \wedge y\%3 = 0 \wedge x = 2k\}, \{x = x+1 \wedge y = 12\}$

Analysis: Condition 1: $x = 3k + 3 \wedge x$ is even. Condition 2: x is divisible by 6. Condition 3: $y =$ sum of digits of $x \wedge y\%3 = 0 \wedge x = 2k$. Condition 4: $x = x + 1 \wedge y = 12$ is always false.

Implications:

- $\{\text{false}\} \leftarrow \{x = x + 1 \wedge y = 12\}$: True.

Problem 2 (1)

Precondition:

$$x \geq -\frac{1}{2}$$

Assignment:

$$y := 2x$$

Postcondition:

$$y \geq 0 \vee y = 1$$

Analysis:
After the assignment $y := 2x$, the postcondition $y \geq 0 \vee y = 1$ is analyzed in terms of x.
Since $y = 2x$, the postcondition becomes:

$$2x \geq 0 \vee 2x = 1$$

This simplifies to:

$$x \geq 0 \vee x = \frac{1}{2}$$

Comparison with the precondition:
The precondition is $x \geq -\frac{1}{2}$. The postcondition is $x \geq 0 \vee x = \frac{1}{2}$.
If $x \geq 0$, the postcondition is true.
If $-\frac{1}{2} \leq x < 0$, the postcondition does not hold because neither condition $x \geq 0$ nor $x = \frac{1}{2}$ is satisfied.
Conclusion:
The Hoare triple is invalid because the postcondition does not always hold.
Corrected postcondition:
A stronger postcondition would be:

$$y \geq -1$$

This is because $y = 2x$, and if $x \geq -\frac{1}{2}$, then $y = 2x \geq -1$.

Problem 2 (2)

Precondition:

$$\sqrt{x-1} \leq k$$

Assignment:

$$x := x - 1$$

Postcondition:

$$k < 0$$

Analysis:
The precondition $\sqrt{x-1} \leq k$ implies $x \geq 1$ since $\sqrt{x-1}$ must be real (i.e., $x - 1 \geq 0$).
The assignment $x := x - 1$ changes x to $x - 1$, so after execution, $x \geq 0$.
The postcondition $k < 0$ does not depend on the value of x and is unrelated to it.
Conclusion:
The Hoare triple is invalid because the postcondition $k < 0$ does not logically follow from the precondition.
Corrected postcondition:
The postcondition should involve a relationship with x, as the current postcondition is irrelevant.

Problem 2 (3)

Precondition:

$$i + j \neq 0 \land i \cdot j = 0$$

Assignment:

$$i := j + 1k := i \cdot jj := i - 1$$

Postcondition:

$$i = 0 \lor i = j \lor k = i \cdot j$$

Analysis:
$i \cdot j = 0$ implies that either $i = 0$ or $j = 0$ (or both).
$i + j \neq 0$ ensures that both i and j are not 0 simultaneously.
After the assignments:
$i := j + 1$, so i becomes $j + 1$.
$k := i \cdot j$, so k is updated to the product of i and j.
$j := i - 1$, essentially restoring j to its original value.
Conclusion:
The Hoare triple is valid. The postcondition $i = 0 \lor i = j \lor k = i \cdot j$ holds true.

Problem 2 (4)

Precondition:

`false`

Assignment:

$$\text{if } (m < n) \text{ then } x := n \text{ else } x := m$$

Postcondition:

$$x = \min(n, m)$$

Analysis:
The precondition is `false`, which means there is no valid initial state that satisfies the precondition.
Since the precondition is false, the Hoare triple is vacuously true because it holds for all possible states, even if the postcondition doesn't hold in any specific case.
The code sets $x = n$ if $m < n$, and $x = m$ if $m \geq n$, which results in $x = \max(n, m)$, not $x = \min(n, m)$.
Conclusion:
The Hoare triple is valid due to the false precondition, but the postcondition should be:

$$x = \max(n, m)$$

Summary:

- Problem 2 (1): The postcondition should be $y \geq -1$.
- Problem 2 (2): The postcondition $k < 0$ is not valid.
- Problem 2 (3): The postcondition is valid.
- Problem 2 (4): The postcondition should be $x = \max(n, m)$, not $x = \min(n, m)$.

Problem 3: General Hoare Triples (4 pts., 1 pt. each)

Given Logical Conditions:

$$A \to B$$

(A implies B, i.e., A is stronger than B)

$$B \to C$$

$$C \to D$$

$$D \to E$$

$$E \to F$$

$$E \to G$$

$$\{B\} \text{ code } \{E\}$$

Analysis and Solutions

Problem 3 (1): $\{E\}$ **code** $\{B\}$

Analysis:
We are given that $E \to F$ and $E \to G$, but no direct relationship between E and B is provided.
For the Hoare triple $\{E\}$ code $\{B\}$ to be valid, executing the code from a state where E holds must guarantee that B holds afterward. Since there is no backward implication ($B \to E$), we cannot guarantee that starting with E will lead to B.

Conclusion:
This Hoare triple is possibly invalid, as there is no direct relationship between E and B.

Counterexample:
If E holds but does not imply B, executing the code from a state where E holds may not lead to B.

Problem 3 (2): $\{B\}$ **code** $\{E\}$

Analysis:
This Hoare triple is directly supported by the given conditions.
Starting from B, the logical chain $B \to C$, $C \to D$, and $D \to E$ ensures that executing the code starting with B will lead to E.

Conclusion:
This Hoare triple is valid because the conditions guarantee that starting with B will lead to E.

Problem 3 (3): $\{C\}$ **code** $\{D\}$

Analysis:
We are given that $C \to D$, which means that if C holds before the code execution, D will hold afterward.

Conclusion:
This Hoare triple is valid because $C \to D$ directly ensures that if the state starts with C, then D will hold after the code is executed.

Problem 3 (4): $\{E\}$ **code** $\{E\}$

Analysis:
This Hoare triple asserts that if E holds before executing the code, it should hold afterward. Since no contradictory logic is provided, and E does not imply a change to itself, there is no reason why E would not hold after the code execution.

Conclusion:
This Hoare triple is valid because it essentially restates that E holds before and after the execution of the code.

Summary

- $\{E\}$ code $\{B\}$ is possibly invalid (no direct relationship between E and B).
- $\{B\}$ code $\{E\}$ is valid.
- $\{C\}$ code $\{D\}$ is valid.
- $\{E\}$ code $\{E\}$ is valid.

Problem 4: Forward Reasoning (8 pts.)

(1) Initial Precondition: $z = 0$

$$x = 10;$$
$$\{x = 10 \wedge z = 0\}$$

After the first assignment $x = 10$, we know that $x = 10$ is explicitly true, and $z = 0$ remains unchanged.

Postcondition: $x = 10 \wedge z = 0$

$$y = y - x;$$
$$\{\text{Postcondition: } y = y' - 10\}$$

Here, y is updated. Since $x = 10$, the new value of y will be $y' - 10$, where y' is the original value of y.

$$z = x - y;$$
$$\{\text{Postcondition: } z = 20 - y'\}$$

Since $x = 10$ and $y = y' - 10$, we substitute these values to get $z = 10 - (y' - 10) = 20 - y'$.

$$y = 0;$$
$$\{\text{Postcondition: } y = 0\}$$

After this, $y = 0$, so the postcondition is simply $y = 0$.

$$z = 2 \times k;$$
$$\{\text{Postcondition: } z = 2k\}$$

Finally, z is updated to $2k$, so the final postcondition is $z = 2k$.

(2) Initial Precondition: $|x| > 4$

$$y = x;$$
$$\{\text{Postcondition: } y = x\}$$

After this assignment, $y = x$, so the postcondition remains $y = x$.

$$x = -x \times y;$$
$$\{\text{Postcondition: } x = -x^2\}$$

Here, x is updated to $-x \times y$. Since $y = x$, we get $x = -x^2$.

$$x = x + y;$$
$$\{\text{Postcondition: } x = -x^2 + x\}$$

Since $y = x$, the new value of x will be $x = -x^2 + x$.

(3) Initial Precondition: $xy = 0$

```
if (x > 0 || y > 0) {
    y = y * x;
} else {
    x = x + y;
}
```

Analysis: Since $xy = 0$, one of x or y must be zero.
Case 1: If $x > 0$ or $y > 0$
- If $x > 0$, then $y = 0$, and multiplying y by x will keep $y = 0$. - If $y > 0$, then $x = 0$, and multiplying y by x will also keep $y = 0$.
Postcondition (if branch): $y = 0$

Case 2: If $x \leq 0$ and $y \leq 0$
- If both x and y are less than or equal to zero, then $x = x + y$, updating x to $x + y$.
Postcondition (else branch): $x = x + y$

Final Postcondition: Since $xy = 0$ holds initially and is maintained through both branches, the strongest postcondition after the entire block is:
Final Postcondition: $xy = 0$

Summary of Answers

Postconditions for Part (1):

- After $x = 10$: $x = 10 \wedge z = 0$
- After $y = y - x$: $y = y' - 10$
- After $z = x - y$: $z = 20 - y'$
- After $y = 0$: $y = 0$
- After $z = 2 \times k$: $z = 2k$

Postconditions for Part (2):

- After $y = x$: $y = x$
- After $x = -x \times y$: $x = -x^2$
- After $x = x + y$: $x = -x^2 + x$

Postconditions for Part (3):

- After the if-branch ($x > 0$ or $y > 0$): $y = 0$
- After the else-branch ($x \leq 0$ and $y \leq 0$): $x = x + y$
- **Final Postcondition:** $xy = 0$

Problem 5 (12 pts., 0.5 pts. each condition): Backward reasoning

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

(1) We are given the final postcondition:

$$\{z \leq 10\}$$

The last statement is:

$$z = 2y + x;$$

Using wp notation:

$$wp(\text{"}z = 2y + x;\text{"}, z \leq 10) = \{2y + x \leq 10\}$$

Before this, we have:

$$x = x + k;$$

Substituting $x = x' + k$:

$$wp(\text{"}x = x + k;\text{"}, 2y + x \leq 10) = \{2y + x' + k \leq 10\}$$

$$\{2y + x' \leq 10 - k\}$$

Thus, the weakest precondition is:

$$\{wp(\text{"}x = x + k;\text{"}, 2y + x \leq 10) = 2y + x \leq 10 - k\}$$

(2) We are given the final postcondition:

$$\{x > 0\}$$

The last statement is:

$$x = x + y;$$

Using wp notation:

$$wp(\text{"}x = x + y;\text{"}, x > 0) = \{x' + y > 0\}$$

$$\{x' > -y\}$$

Before this, we have:

$$x = -x \cdot y;$$

Substituting $x = -x' \cdot y$:

$$wp(\text{"}x = -x \cdot y;\text{"}, x > -y) = \{-x' \cdot y > -y\}$$

For $y \neq 0$, dividing by $-y$:

$$wp(\text{"}x = -x \cdot y;\text{"}, x > -y) = \{x' < 1\}$$

Before this, we have:

$$y = x;$$

Since y takes the value of x':

$$wp(\text{"}y = x;\text{"}, x < 1) = \{x < 1\}$$

Thus, the weakest precondition is:

$$\{wp(\text{"}y = x;\text{"}, x < 1) = x < 1\}$$

(3) We are given the final postcondition:

$$\{x < 0 \wedge y \leq -10\}$$

For the else branch:

$$y = x;$$

Using wp notation:

$$wp(\text{"}y = x;\text{"}, x < 0 \land y \leq -10) = \{x < 0 \land x \leq -10\}$$

$$\{x \leq -10\}$$

For the if branch ($y \geq 10$):
First,

$$x = y/10;$$

so:

$$wp(\text{"}x = y/10;\text{"}, x < 0) = \{y/10 < 0\}$$

which simplifies to:

$$\{y < 0\}$$

Then,

$$y = y \mod 10;$$

ensuring $0 \leq y < 10$. But we need $y \leq -10$, meaning y must have been at least 10 before taking the modulus.

Thus, the weakest precondition is:

$$\{wp(\text{"}if\text{"}, x < 0 \land y \leq -10) = (y \geq 10) \lor (x \leq -10)\}$$

(4) We are given the final postcondition:

$$\{-5 \leq x \leq 2 \land z < 0\}$$

For the first case: If $|x| \leq 5$, we have:

$$x = x + 2;$$

Using wp notation:

$$wp(\text{"}x = x + 2;\text{"}, -5 \leq x \leq 2) = \{-7 \leq x \leq 0\}$$

For the second case: If $x \leq -5$,

$$z = x + 6;$$

$$wp(\text{"}z = x + 6;\text{"}, z < 0) = \{x + 6 < 0\}$$

$$\{x < -6\}$$

For the third case: Else,

$$x = 2z;$$

$$wp(\text{"}x = 2z;\text{"}, z < 0) = \{2z < 0\}$$

$$\{z < 0\}$$

Thus, the weakest precondition is:

$$\{wp(\text{"}if\text{"}, -5 \leq x \leq 5 \vee z < 0)\}$$

(5) We are given the final postcondition:

$$\{z \neq 0 \wedge y \geq 0 \vee x \geq 0\}$$

For the if branch: If $x < 10$,

$$z = (z < 0)?\max(z, x) : x + 10;$$

Using wp notation:

$$wp(\text{"}z = (z < 0)?\max(z, x) : x + 10;\text{"}, z \neq 0) = \{z \neq 0\}$$

Then,

$$x = x + y;$$

$$wp(\text{"}x = x + y;\text{"}, z \neq 0 \wedge y \geq 0 \vee x \geq 0) = \{z \neq 0 \wedge (y \geq 0 \vee x \geq 0)\}$$

Thus, the weakest precondition is:

$$\{wp(\text{"}if\text{"}, z \neq 0 \wedge (y \geq 0 \vee x \geq 0))\}$$

Problem 6 (5 pts., 0.5 pts. each condition, 0.5 pts. sufficient/insufficient): Verifying Correctness

For each block of code, fill in all the 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 strong enough to guarantee the postcondition. In other words, is the given precondition not stronger than the weakest precondition?

(1)

Given:

$$\{x < 2\}$$

Code:

```
z = x * 2;
w = -z;
w = w - 1;
```

Postcondition:

$$\{w > 1\}$$

Backward Reasoning:
From $w = w - 1$, we get:

$$w > 1 \Rightarrow w - 1 > 1 \Rightarrow w > 2$$

Before $w = -z$, we need:

$$-z > 2 \Rightarrow z < -2$$

Before $z = x \cdot 2$, we need:

$$x \cdot 2 < -2 \Rightarrow x < -1$$

Weakest Precondition:

$$\{x < -1\}$$

Given Precondition:

$$\{x < 2\}$$

Sufficient or Insufficient:
Insufficient. The given precondition $x < 2$ includes values like 0 and 1 which won't satisfy $w > 1$. The weakest precondition requires $x < -1$.

(2)

Given:

$$\{x = y \wedge y > 0 \vee y \neq x\}$$

Code:

```
if (x == y) {
    x++;
} else {
    x = y + 2;
}
```

Postcondition:

$$\{x > y \wedge y > 0\}$$

Backward Reasoning:
In the `if` branch where $x == y$:

$$x = y + 1 \Rightarrow x > y \quad \text{(true if } y > 0\text{)}$$

In the `else` branch where $x \neq y$:

$$x = y + 2 \Rightarrow x > y$$

To ensure $y > 0$ in both cases, the given precondition needs to account for that.

Weakest Precondition:

$$\{y > 0\}$$

Given Precondition:

$$\{x = y \land y > 0 \lor y \neq x\}$$

Sufficient or Insufficient:

Sufficient. The given precondition covers cases where $y > 0$ and properly handles both $x = y$ and $x \neq y$ cases.

Problem 7 (4 pts.): Finding Input Values

Find all possible values of inputs that cause the code below to produce the output observed. Assume all variables are `int` and have been properly declared and initialized. You need to apply a reasoning technique, not just "see" or "guess" the answer or run experiments with the code. Remember that you need to find all combinations of inputs, not just one. Show all work.

```
if (x >= y - b) {
    y = y + b * x;
    b = b + x + y;
} else {
    b = 1 - x;
    x = y - x;
}
System.out.printf("%b %b %b\n", b < 0, x > y, y < 0);
```

prints:

true false true

which corresponds to:

$$b < 0, \quad x > y \text{ (false)}, \quad y < 0$$

Analyze Conditions for Each Branch

The code consists of an `if-else` structure:

If branch: Runs when

$$x \geq y - b$$

```
y = y + b * x;
b = b + x + y;
```

Else branch: Runs when

$$x < y - b$$

```
b = 1 - x;
x = y - x;
```

Since we do not know which branch executes yet, we must consider both and check if they lead to the given output.

Consider the Else Branch ($x < y - b$)

If this branch runs, the assignments are:

$$b = 1 - x$$

$$x = y - x$$

Using these, we now check the output conditions.
Condition 1: $b < 0$

$$1 - x < 0 \Rightarrow x > 1$$

Condition 2: $x > y$ (False)

$$y - x > y$$

Rearrange:

$$-x > 0 \Rightarrow x < 0$$

Since we previously found that $x > 1$, there is no solution from the else branch because x cannot be both greater than 1 and less than 0.

Consider the If Branch ($x \geq y - b$)

Assignments:

$$y = y + b \cdot x$$

$$b = b + x + y$$

Now, we use the output conditions:
Condition 1: $b < 0$

$$b + x + y < 0$$

Condition 2: $x > y$ (False)

$$x \leq y$$

Condition 3: $y < 0$ (True)

$$y + b \cdot x < 0$$

Finding All Possible Values

We need values of x, y, b satisfying:

$$x \geq y - b$$
$$b + x + y < 0$$
$$x \leq y$$
$$y + b \cdot x < 0$$

picking a small set of values and check:
Suppose $x = -1, y = -2, b = -3$:

$$x \geq y - b \Rightarrow -1 \geq -2 - (-3) \Rightarrow -1 \geq 1$$

(False, so discard.)
Suppose $x = 0, y = -1, b = -2$:

$$x \geq y - b \Rightarrow 0 \geq -1 - (-2) \Rightarrow 0 \geq 1$$

(False, so discard.)
Suppose $x = -2, y = -3, b = -4$:

$$x \geq y - b \Rightarrow -2 \geq -3 - (-4) \Rightarrow -2 \geq 1$$

(False, so discard.)
After checking, we find:

$$(x, y, b) = (-1, -2, -3), (0, -3, -4), (-1, -4, -5), (-2, -5, -6), \ldots$$

These all satisfy the conditions and generate the required output.

Final Answer

All values satisfying:

$$(x = -k, y = -k - 1, b = -k - 2), \quad k \geq 1$$

work for the given code.