Software Analysis & Verification	**Due: Jun 3, 2020**

Homework 6

Instructor: Fei He
TA: Jianhui Chen, Fengmin Zhu

YOUR NAME (YOUR TSINGHUA ID)

Read the instructions below carefully before you start working on the assignment:

- Please typeset your answers in the attached LaTeX source file, compile it to a PDF, and finally hand the PDF to Tsinghua Web Learning *before the due date.*
- Make sure you fill in your *name* and *Tsinghua ID*, and replace all "`TODO`"s with your solutions.
- Any kind of dishonesty is *strictly prohibited* in the full semester. If you refer to any material that is not provided by us, you *must cite* it.

Problem 1: Multiple Choice

For each of questions below, four choices marked (A), (B), (C) and (D) are provided. ONLY ONE of them is correct. Read the questions carefully and choose the correct answers.

1-1 Which of the following is *not* a loop invariant for the following IMP loop?

$$\textsf{while } Y > 0 \textsf{ do } Y := Y - 1; X := X + 1 \textsf{ end}$$

(A) $X > 10$.

(B) $Y > 10$.

(C) $X + Y = Z$.

(D) $Z + Y < X$.

Solution B ■

1-2 Which of the following is *false* about the IMP program shown below?

$$
\begin{array}{l}
X := 1; \\
\textsf{while } X > 0 \textsf{ do} \\
\quad \textsf{if } N \le 100 \textsf{ then} \\
\quad\quad N := N + 11; \\
\quad\quad X := X + 1 \\
\quad \textsf{else} \\
\quad\quad N := N - 10; \\
\quad\quad X := X - 1 \\
\quad \textsf{fi} \\
\textsf{end}
\end{array}
$$

(A) $X = 0$ is a post condition.

(B) $X \geq 0$ is a loop invariant (for the while-loop).

(C) $N \leq 111$ is a loop invariant (for the while-loop).

(D) The program may not terminate.

Solution D ■

1-3 Let $[X = 0]$ while b do c end $[X = 1]$ be a Hoare triple. Which of the following is *true*?

(A) If c is $X := 1$, then the Hoare triple is valid for some b.

(B) If b is true, then the Hoare triple is valid for some c.

(C) If b is $X \neq 1$, then the Hoare triple is valid no matter what c is.

(D) The Hoare triple is always invalid no matter what b and c are.

Solution A ■

1-4 Recall that two IMP programs (with havoc) c_1 and c_2 are *behaviorally equivalent*, if for every states σ and σ', their big-step operational semantic evaluation relations satisfy $\langle \sigma, c_1 \rangle \Downarrow \sigma' \iff \langle \sigma, c_2 \rangle \Downarrow \sigma'$. In which of the following are c_1 and c_2 behaviorally equivalent?

(A) $c_1 : X := Y; Y := X$ $\quad c_2 : Y := X; X := Y$

(B) $c_1 :$ skip $\quad c_2 :$ if $X > 10$ then $X := 0$ else skip fi

(C) $c_1 :$ havoc $X; X := 10$ $\quad c_2 : X := 10$

(D) $c_1 :$ havoc $X;$ havoc Y $\quad c_2 :$ havoc Y

Solution C ■

1-5 Let F be a CNF with four variables x_1, x_2, x_3, x_4. We apply the DPLL algorithm (without backjump) on F and the following operations are done: decide x_1, propagate x_2, propagate x_3. Which of the following operations will be possibly done in the next step?

(A) Decide $\neg x_3$.

(B) Backtrack and decide $\neg x_1$.

(C) Backtrack and decide $\neg x_2$.

(D) Backtrack and decide $\neg x_3$.

Solution B ■

Problem 2: Assumptions & Assertions

We consider two kinds of commands which indicate a certain statement should hold any time this part of the program is reached – the assumption statement "$\textsf{assume}\ b$", and the assertion statement "$\textsf{assert}\ b$":

- If an assertion statement fails, it causes the program to go into an *error state* and exit (or abort).
- If an assumption statement fails, the program fails to evaluate at all. In other words, the program gets *stuck* and has no final state.

To formally express the program may go into an error state, we have to change the evaluation relation (of big-step operational semantics) from "$\langle \sigma, c\rangle \Downarrow \sigma'$" into "$\langle \sigma, c\rangle \Downarrow r$", where the evaluation *result*

$$r ::= \mathsf{norm}(\sigma) \mid \mathsf{err}$$

can state two possible cases: $\mathsf{norm}(\sigma)$ for normally execution with ending state σ, or err for reaching the error state. The inference rules for the original IMP commands need be modified and we should handle errors carefully (read and think about the differences):

$$\text{(Skip)}\ \frac{}{\langle \sigma, \textsf{skip}\rangle \Downarrow \mathsf{norm}(\sigma)} \qquad \text{(Ass)}\ \frac{\mathcal{A}[\![a]\!]_\sigma = n}{\langle \sigma, x := a\rangle \Downarrow \mathsf{norm}(\sigma[x \mapsto n])}$$

$$\text{(Seq)}\ \frac{\langle \sigma, c_1\rangle \Downarrow \mathsf{norm}(\sigma') \qquad \langle \sigma', c_2\rangle \Downarrow r}{\langle \sigma, c_1; c_2\rangle \Downarrow r} \qquad \text{(SeqErr)}\ \frac{\langle \sigma, c_1\rangle \Downarrow \mathsf{err}}{\langle \sigma, c_1; c_2\rangle \Downarrow \mathsf{err}}$$

$$\text{(IfTrue)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \top \qquad \langle \sigma, c_1\rangle \Downarrow r}{\langle \sigma, \textsf{if}\ b\ \textsf{then}\ c_1\ \textsf{else}\ c_2\ \textsf{fi}\rangle \Downarrow r} \qquad \text{(IfFalse)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \bot \qquad \langle \sigma, c_2\rangle \Downarrow r}{\langle \sigma, \textsf{if}\ b\ \textsf{then}\ c_1\ \textsf{else}\ c_2\ \textsf{fi}\rangle \Downarrow r}$$

$$\text{(WhileFalse)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \bot}{\langle \sigma, \textsf{while}\ b\ \textsf{do}\ c\ \textsf{end}\rangle \Downarrow \mathsf{norm}(\sigma)} \qquad \text{(WhileTrue)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \top \qquad \langle \sigma, c\rangle \Downarrow \mathsf{norm}(\sigma') \qquad \langle \sigma', \textsf{while}\ b\ \textsf{do}\ c\ \textsf{end}\rangle \Downarrow r}{\langle \sigma, \textsf{while}\ b\ \textsf{do}\ c\ \textsf{end}\rangle \Downarrow r}$$

$$\text{(WhileTrueErr)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \top \qquad \langle \sigma, c\rangle \Downarrow \mathsf{err}}{\langle \sigma, \textsf{while}\ b\ \textsf{do}\ c\ \textsf{end}\rangle \Downarrow \mathsf{err}}$$

We redefine Hoare triples "$\{P\}\ c\ \{Q\}$" to mean that, whenever c is started in a state satisfying P, and terminates with result r, then $r = \mathsf{norm}(\sigma)$ (and hence $r \neq \mathsf{err}$) where the state σ satisfies Q.

2-1 Give the evaluation rules for assumption and assertion statements.

Solution

$$\text{(AssertTrue)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \top}{\langle \sigma, \textsf{assert}\ b\rangle \Downarrow \mathsf{norm}(\sigma)} \qquad \text{(AssertFalse)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \bot}{\langle \sigma, \textsf{assert}\ b\rangle \Downarrow \mathsf{err}}$$

$$\text{(AssumeTrue)}\ \frac{\mathcal{B}[\![b]\!]_\sigma = \top}{\langle \sigma, \textsf{assume}\ b\rangle \Downarrow \mathsf{norm}(\sigma)}$$

■

2-2 Design Hoare rules for assumption and assertion statements.

Solution

$$\text{(AssertTrue)}\ \frac{P \Rightarrow b}{\{P\}\ \textsf{assert}\ b\ \{P\}} \qquad \text{(AssertFalse)}\ \frac{P \Rightarrow \neg b}{\{P\}\ \textsf{assume}\ b\ \{\top\}}$$

$$\text{(AssumeTrue)}\ \frac{P \Rightarrow b}{\{P\}\ \textsf{assume}\ b\ \{P\}} \qquad \text{(AssumeFalse)}\ \frac{P \Rightarrow \neg b}{\{P\}\ \textsf{assume}\ b\ \{\bot\}}$$

■

2-3 Compute $\mathsf{wlp}(X := X + 1; \textcolor{blue}{\mathsf{assume}}\ X > 0; Y := Y + X, X + Y + Y \geq 3)$.

Solution

$$
\begin{aligned}
&\mathsf{wlp}(X := X + 1; \mathsf{assume}\ X > 0; Y := Y + X, X + Y + Y \geq 3) \\
&= \mathsf{wlp}(X := X + 1; \mathsf{assume}\ X > 0, \mathsf{wlp}(Y := Y + X, X + Y + Y \geq 3)) \\
&= \mathsf{wlp}(X := X + 1; \mathsf{assume}\ X > 0, X + Y + X + Y + X \geq 3) \\
&= \mathsf{wlp}(X := X + 1, \mathsf{wlp}(\mathsf{assume}\ X > 0, X + Y + X + Y + X \geq 3)) \\
&= \mathsf{wlp}(X := X + 1, (X > 0) \to (X + Y + X + Y + X \geq 3)) \\
&= (X + 1 > 0) \to (X + 1 + Y + X + 1 + Y + X + 1 \geq 3) \\
&= Y \geq 0
\end{aligned}
$$

■