Exercise 1: Hoare logic

In this exercise we consider very simple Hoare triples over Boolean variables where

- the precondition \( precond(X_1, \ldots, X_n) \) is a Boolean expression over the Boolean variables \( X_1, \ldots, X_n \) and does not contain the Boolean variable \( Y \),
- the program consists of the single line
  \[ \begin{align*}
  & \quad Y := expr(X_1, \ldots, X_n),
  
  \end{align*} \]
  where \( Y \) is a Boolean variable and \( expr(X_1, \ldots, X_n) \) is a Boolean expression over the Boolean variables \( X_1, \ldots, X_n \) that does not contain \( Y \), and
- the postcondition \( postcond(X_1, \ldots, X_n) \) is a Boolean expression over the variables \( Y, X_1, \ldots, X_n \).

(a) State a propositional logical formula

\[ vc(Y, X_1, \ldots, X_n) \]

that is valid if and only if a Hoare triple that has the following form is valid.

\[ \{ \begin{align*}
  & \quad precond(X_1, \ldots, X_n) \} \ Y := expr(X_1, \ldots, X_n) \ \{ \begin{align*}
  & \quad postcond(Y, X_1, \ldots, X_n) \} 
  
  \end{align*} \]

(b) Compute your propositional logical formula \( vc(Z, U, V) \) for the following concrete program.

\[ \{ \begin{align*}
  & \quad U \leftrightarrow V \} Z := U \land V \ \{ \begin{align*}
  & \quad Z \leftrightarrow U \} 
  
  \end{align*} \]

Is your formula valid?

(c) Now we drop the restriction that \( precond(X_1, \ldots, X_n) \) does not contain the Boolean variable \( Y \). Find a Hoare triple that is not valid but where your formula \( vc(U, V, Z) \) is valid.
Exercise 2: Hoare logic derivation 2 Points
(a) Write down a partial correctness specification (i.e., precondition and postcondition) for a program C that computes the maximum of x and y and stores the result in z.

(b) Write down the program C. Use the syntax for while programs introduced in the lecture.

(c) Construct a Hoare logic derivation that proves that your program C fulfills your correctness specification.

Exercise 3: Hoare triples 2 Points
Consider the following Hoare triples. Which of them are valid for any program C and any state assertion \( \phi \)?

(a) \{ true \} C \{ \phi \}

(b) \{ false \} C \{ \phi \}

(c) \{ \phi \} C \{ true \}

(d) \{ \phi \} C \{ false \}

If a Hoare triple is valid for any program C and any state assertion \( \phi \), then explain why. If a Hoare triple is not valid for some program C and some state assertion \( \phi \), then give a counterexample.

Exercise 4: Loop Invariant, Invariant, Inductive Invariant 2 Points
(a) Consider the following while command

\[ C \equiv \text{while } x < 42 \text{ do } x := x + y \]

and precondition \( \phi \equiv x = 1 \land y = 1 \).

(i) Find a state assertion \( \theta_1 \) that implies \( x \geq 0 \) and is a loop invariant but not an invariant. (Be careful! It is not the one given for the program in the lecture, even though the program may look similar.)

(ii) Find a state assertion \( \theta_2 \) that implies \( x \geq 0 \) and is an invariant but not an inductive invariant.

(iii) Find a state assertion \( \theta_3 \) that implies \( x \geq 0 \) and is an inductive invariant.

(b) Consider the following scheme of a while command

\[ C \equiv \text{while } b \text{ do } x := x + y \]

and precondition \( \phi \equiv x = 1 \land y = 1 \). Furthermore, let \( \theta \equiv x \geq 0 \).

(i) Find an expression b such that \( \theta \) is a loop invariant but not an invariant.

(ii) Find an expression b such that \( \theta \) is an invariant but not an inductive invariant.

(iii) Find an expression b such that \( \theta \) is an inductive invariant.