Dr. Matthias Heizmann
Tanja Schindler
Dominik Klumpp

Hand in until 10:00 on May 27, 2019
Discussion: May 27, 2019

Tutorial for Program Verification
Exercise Sheet 8

Exercise 1: Program Semantics 3 Points

In the lecture we defined the semantics of Boo programs by assigning a relation to each statement. Compute this relation for the Boo program $P_{\mathsf{pow}} = (V, \mu, \mathcal{T})$ with $V = \{e, x, y, z\}$, $\mu(e) = \mu(x) = \mu(y) = \mu(z) = \mathbb{Z}$, and $\mathcal{T}$ a derivation tree for the program code shown below. List all intermediate steps, i.e., state the relation for each sub-statement.

```
e := 1;
z := 0;

while (z < y) {
  e := e * x;
  z := z + 1;
}
```

Exercise 2: Precondition - Postcondition 2 Points

Consider the following precondition-postcondition pairs. Which of them are satisfied by all program statements st and all formulas φ?

(a) $\{\textbf{true}\}\ st\ \{\varphi\}$

(b) $\{\textbf{false}\}\ st\ \{\varphi\}$

(c) $\{\varphi\}\ st\ \{\textbf{true}\}$

(d) $\{\varphi\}\ st\ \{\textbf{false}\}$

If a precondition-postcondition is satisfied by all program statements st and all formulas φ, then explain why. If a precondition-postcondition is not satisfied by some program statement st and some formulas φ, then give a counterexample.

Exercise 3: Assignment Axiom 1 Point

Find some program P whose code is a single assignment statement of the form `x := expr;` and some formula φ such that P does not satisfy the precondition-postcondition pair $(\{\varphi\}, \{\varphi \wedge x = \texttt{expr}\})$.
The motivation of this exercise is the following. In the lecture we have seen the assignment axiom of the Hoare proof system.

$$(assig)\frac{}{\{\varphi[x \mapsto \texttt{expr}]\}\ \texttt{x := expr;}\ \{\varphi\}}$$

This rule is not very intuitive because the precondition is obtained as a modification of the postcondition. One may wonder if the following proof rule could be an alternative.

$$(BadAss)\frac{}{\{\varphi\}\ \texttt{x := expr;}\ \{\varphi \wedge x = \texttt{expr}\}}$$

The result of this exercise should hint that the (BadAss) proof rule cannot be used as an axiom in a proof system whose goal is the derivation of valid Hoare triples.

Exercise 4: Hoare Proof System 3 Points

Is there a program that can swap the values of two variables without using a temporary variable? In this exercise we will consider such a program and prove that the program indeed has this property.

At the beginning of this course, we used the $\mathcal{N}_{\mathsf{PL}}$ proof system to derive valid implications of the form $\Gamma \vDash F$. Analogously, we will use the Hoare proof system to derive valid Hoare triples of the form $\{\varphi\}st\{\psi\}$.
Analogously to $\mathcal{N}_{\mathsf{PL}}$ we define a *derivation* as a tree whose nodes are labelled by Hoare triples such that the following holds.
If a node that is labelled by a Hoare triple $\{\varphi_{n+1}\}st_{n+1}\{\psi_{n+1}\}$ has children that are labelled by Hoare triples $\{\varphi_1\}st_1\{\psi_1\}$ $\quad\ldots\quad$ $\{\varphi_n\}st_n\{\psi_n\}$ then

$$\frac{\{\varphi_1\}st_1\{\psi_1\} \quad \ldots \quad \{\varphi_n\}st_n\{\psi_n\}}{\{\varphi_{n+1}\}st_{n+1}\{\psi_{n+1}\}}$$

must be an instance of some rule. This means in particular that leafs of the tree may only be labelled by the assignment axiom.
Analogously to $\mathcal{N}_{\mathsf{PL}}$ we can prove that a Hoare triple $\{\varphi\}st\{\psi\}$ is valid by stating a derivation whose root is labelled by $\{\varphi\}st\{\psi\}$.

Consider the Boo program $P_{\mathsf{swap}} = (V, \mu, \mathcal{T})$ with $V = \{a, b, x, y\}$, $\mu(a) = \mu(b) = \mu(x) = \mu(y) = \mathbb{Z}$, and $\mathcal{T}$ a derivation tree for the program code shown below.

```
x := x + y;
y := x - y;
x := x - y;
```

Use the Hoare proof system to show that P satisfies the precondition-postcondition pair $(\{x = a \wedge y = b\}, \{x = b \wedge y = a\})$.
In the lecture we have already seen the following two rules of the Hoare proof system.

$$(assig)\frac{}{\{\varphi[x \mapsto \texttt{expr}]\}\texttt{x := expr;}\{\varphi\}} \qquad (compo)\frac{\{\varphi_1\}st_1\{\varphi_2\} \quad \{\varphi_2\}st_2\{\varphi_3\}}{\{\varphi_1\}st_1st_2\{\varphi_3\}}$$

The lecturer forgot to mention/emphasize that for the rule "compo" the set of states $\{\varphi_2\}$ on the upper left hand side and the set of states $\{\varphi_2\}$ on the upper right hand side are defined by the very same formula φ_2. (It is not sufficient to use two different logically equivalent formulas!)

For solving this exercise you also need one of the following two rules which have not been mentioned in the lecture. Both rules have the validity of an implication between two formulas as a side-condition.

$$(strepre)\frac{\{\varphi\}st\{\psi\}}{\{\varphi'\}st\{\psi\}} \text{ if } \varphi' \vDash \varphi \qquad\qquad (weakpos)\frac{\{\varphi\}st\{\psi\}}{\{\varphi\}st\{\psi'\}} \text{ if } \psi \vDash \psi'$$