Dr. Matthias Heizmann
Dominik Klumpp

Hand in until 10:00 on July 06, 2020
Discussion: July 06, 2020

Tutorial for Program Verification
Exercise Sheet 14

In this exercise sheet we work with the Hoare proof system, control flow graphs, and reachability graphs.

Submit your solution by uploading it as PDF in ILIAS.

Exercise 1: Alternative Assume Axiom — 2 Points

In the lecture, we introduced the following axiom for the `assume` statement in the Hoare proof system:

$$\frac{}{\{\varphi\}\ \texttt{assume expr;}\ \{\varphi \land \texttt{expr}\}}\ (\mathit{assu})$$

Alternatively, we could have introduced the following axiom:

$$\frac{}{\{\texttt{expr} \to \psi\}\ \texttt{assume expr;}\ \{\psi\}}\ (\mathit{assu}')$$

In this exercise we will show that both rules are equivalent.

(a) Give a proof for the Hoare triple $\{\texttt{expr} \to \psi\}\ \texttt{assume expr;}\ \{\psi\}$ (for an arbitrary formula ψ) using the Hoare proof system.

(b) Give a proof of the Hoare triple $\{\varphi\}\ \texttt{assume expr;}\ \{\varphi \land \texttt{expr}\}$ for an arbitrary formula φ. Use a modified variant of the Hoare proof system, where the rule (assu) has been replaced by the rule (assu').

Exercise 2: From Programs to CFGs — 2 Points

For each of the programs given below, draw a control-flow graph.

(a) Code of program P_{pow}:

```
e := 1;
z := 0;
while (z < y) {
  e := e * x;
  z := z + 1;
}
```

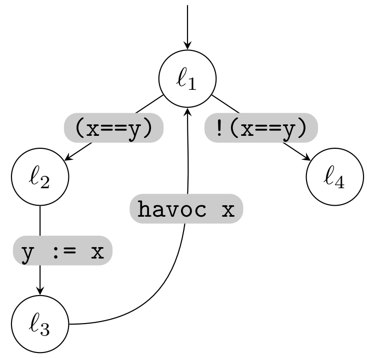
(b) Code of program P_{findmin}:

```
i := lo;
min := a[lo, lo];
while (i <= hi) {
  j := lo;
  while (j <= hi) {
    if (a[i, j] < min) {
      min := a[i, j];
    }
    j := j + 1;
  }
  i := i + 1;
}
```

Exercise 3: Program Configurations 2 Points

Consider the program $P = (V, \mu, \mathcal{T})$ with $V = \{x, y\}$, $\mu(x) = \mu(y) = \{\mathbf{true}, \mathbf{false}\}$ and $\mathcal{T}$ a derivation tree for the statement below on the left. On the right, a CFG for P is shown.

```
while (x == y) {
  y := x;
  havoc x;
}
```



Draw the reachability graph for this control-flow graph and the precondition-postcondition-pair $(x, x \to \neg y)$.

Exercise 4: Existence of Program Executions 2 Points

Prove the following lemma from the lecture slides.

> **Lemma** (RelAndExec.2) Let $G = (Loc, \Delta, \ell_{\mathsf{init}}, \ell_{\mathsf{ex}})$ be a control-flow graph for the sequential composition $st_1 st_2$. There exists a program execution $(\ell_0, s_0), \ldots, (\ell_n, s_n)$ with $\ell_0 = \ell_{\mathsf{init}}$ and $\ell_n = \ell_{\mathsf{ex}}$, iff $(s_0, s_n) \in [\![st_1 st_2]\!]$.