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

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

\[ \{\varphi\} \text{assume expr}; \{\varphi \land \text{expr}\} \]

(\text{assu})

Alternatively, we could have introduced the following axiom:

\[ \{\text{expr} \rightarrow \psi\} \text{assume expr}; \{\psi\} \]

(\text{assu}')

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

(a) Give a proof for the Hoare triple \(\{\text{expr} \rightarrow \psi\} \text{assume expr}; \{\psi\}\) (for an arbitrary formula \(\psi\)) using the Hoare proof system.

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 Use a modified variant of the Hoare proof system, where the rule (\text{assu}) has been replaced by the rule (\text{assu}').

Exercise 2: From Programs to CFGs

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

(a) Code of program \(P_{\text{pow}}\):

```
1 e := 1;
2 z := 0;
3 while (z < y) {
4   e := e * x;
5   z := z + 1;
6 }
```
(b) Code of program $P_{\text{findmin}}$:

```plaintext
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  

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

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

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

Exercise 4: Existence of Program Executions  

Prove the following lemma from the lecture slides.

**Lemma** (RelAndExec.2) Let $G = (\text{Loc}, \Delta, \ell_{\text{init}}, \ell_{\text{ex}})$ be a control-flow graph for the sequential composition $st_1st_2$. There exists a program execution $(\ell_0, s_0), \ldots, (\ell_n, s_n)$ with $\ell_0 = \ell_{\text{init}}$ and $\ell_n = \ell_{\text{ex}}$, iff $(s_0, s_n) \in [st_1st_2]$. 
