Tutorial for Program Verification
Exercise Sheet 14

Exercise 1: CFG for Conditional Statement 2 Points
In the lecture, we defined the notion of a control-flow graph of a given statement. This
definition is not yet complete, the case of the conditional-statement was left out. Com-
plete the definition:

Let $st_1, st_2$ be two statements. Let $G_1 = (Loc^1, \Delta^1, \ell^1_{init}, \ell^1_{exit})$ be a control flow graph for
$st_1$, and let $G_2 = (Loc^2, \Delta^2, \ell^2_{init}, \ell^2_{exit})$ be a control flow graph for $st_2$ such that $Loc^1$ and
$Loc^2$ are disjoint. Define a control flow graph of if (expr) { $st_1$ } else { $st_2$ }.

Exercise 2: From Programs to CFGs 2 Points
For each of the programs given below, draw a control-flow graph.

(a) Code of program $P_{pow}$:

```
1 e := 1;
2 z := 0;
3 while (z < y) {
4   e := e * x;
5   z := z + 1;
6 }
```

(b) Code of program $P_{findmin}$:

```
1 i := lo;
2 min := a[lo, lo];
3 while (i <= hi) {
4   j := lo;
5   while (j <= hi) {
6     if (a[i, j] < min) {
7       min := a[i, j];
8     }
9   } 
10  j := j + 1;
11 }
12 i := i + 1;
13 }
```
Exercise 3: Program Configurations  

Consider the program \( P = (V, \mu, T) \) with \( V = \{x, y\} \), \( \mu(x) = \mu(y) = \{\text{true}, \text{false}\} \) and \( T \) 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 \((x, x \rightarrow \neg y)\).

Exercise 4: Existence of Program Executions 

Proof the following lemma that has been added to the slides.

**Lemma (RelAndExec.2)** Let \( G = (\text{Loc}, \Delta, \ell_{\text{init}}, \ell_{\text{ex}}) \) be a control-flow graph for the sequential composition \( st_1 st_2 \). 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 \((s_0, s_n) \in [st_1 st_2]\).