Dr. Matthias Heizmann
Tanja Schindler
Dominik Klumpp

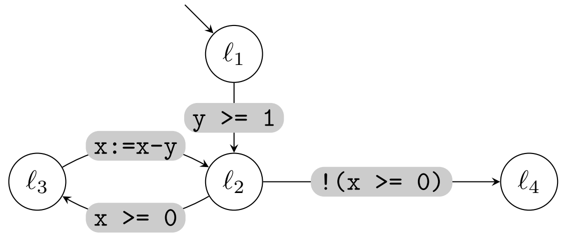
Hand in until 16:15 on July 24, 2019
Discussion: July 24, 2019

# Tutorial for Program Verification

## Exercise Sheet 23

Let us consider the program whose code and control-flow graph are given below.

```
assume(y >= 1);
while (x >= 0) {
  x := x - y;
}
```



The program is terminating, however there is no ranking function for the while loop.

**Definition (Loop Entry)** Given a while loop `while(expr){st}` and a control-flow graph $G = (Loc, \Delta, \ell_{\mathsf{init}}, \ell_{\mathsf{ex}})$ for this while loop, we call $\ell_{\mathsf{init}}$ the *entry location* of the while loop.

**Definition (Ranking Function)** Given a program $P = (V, \mu, st)$, a Floyd-Hoare annotation $\beta$ for $P$, a while loop `while(expr){st}` whose loop entry is the location $\ell$, and a set $W$ together with a well-founded relation $R \subseteq W \times W$, we call a function $f : S_{V,\mu} \to W$ a *ranking function* for `while(expr){st}` and $\beta$ if for each pairs of states where $s \in \{\beta(\ell)\}$ and $(s, s') \in [\![$`assume expr; st`$]\!]$, the relation $(f(s), f(s')) \in R$ holds.

### Exercise 1: Ranking Function — 2 Points

Give a Floyd-Hoare annotation $\beta$ and a function $f$ such that $f$ is a ranking function for $\beta$ and the while loop.

### Exercise 2: Easter Bunny — 2 Bonus Points

Let `k` be an integer variable and `a` be an array that has integer indices and integer values. Is the following program terminating?

```
while (a[a[k]] >= 0) {
  a[a[k]] := a[a[k]] - 1;
}
```

A story that motivates this program:
The array `a` models some street with infinitely many farmhouses. Each farmhouse has a finite number of eggs and because of the zombie apocalypse no new eggs can be produced. The easter bunny is stealing eggs in this street according to a strategy given below and we will ask ourselves if the easter bunny will eventually fail to find new eggs.

The array `a` maps house numbers to the number of eggs that are available in the house. The easter bunny picks initially an integer `k`, and steals eggs iteratively (one at a time) according to the following strategy. In each iteration, the easter bunny first visits farm number `k` and checks the amount of eggs that are available at that farm. Then he visits another farm, namely the farm whose house number coincides with this amount. If no egg is available at this farm, the easter bunny gives up. Otherwise he steals one egg and continues with the next iteration.

Is this a bad strategy because the easter bunny will eventually visit always the same farm? Or is there some street and some choice of k in which the easter bunny can continue forever?

Please note that the easter bunny does not neccesarily steal eggs from the same farm. If he takes the egg from farm number $k$ he will alter the farm that he visits next.