**Exercise Sheet 8**

**Due date:** June 28th. Please hand in your solutions at the start of the exercise class.

## Task 1: Hoare Logic for Timed Correctness (40 Points)

Consider the Hoare logic for timed correctness (Lecture 13, Definition 13.7).

(a) Show that the following rule for sequential composition is *not* sound.

$$\frac{\{A\}c_1\{e_1 \Downarrow C\} \quad \{C\}c_2\{e_2 \Downarrow B\}}{\{A\}c_1; c_2\{e_1 + e_2 \Downarrow B\}}$$

That is, provide programs $c_1, c_2$, assertions $A, B$, and arithmetic expressions $e_1, e_2$, which satisfy the premise of the above rule but do not satisfy the conclusion.

(b) Determine an arithmetic expression $e$ such that for your programs $c_1, c_2$ and your assertions $A, B$ from (a) it holds that $\vdash \{A\}c_1; c_2\{e \Downarrow B\}$. Prove this triple in Hoare logic for timed correctness using the sound rule for sequential composition (Definition 11.13)

## Task 2: Operational Semantics of Procedure Calls (30 Points)

A naïve version of the operational semantics of procedure calls might be defined as follows:

$$\frac{(\rho, \pi) \vdash \langle c, \sigma \rangle \to \sigma' \qquad \pi(P) = (c, \rho', \pi')}{(\rho, \pi) \vdash \langle \mathsf{call}\ \mathsf{P}, \sigma \rangle \to \sigma'}$$

Construct a program $c$ with procedures that illustrates the difference between the above rule and the call-rule from the lecture (Definition 14.2).

Validate your claim by constructing two different derivation trees (one using the above rule, one using the rule from the lecture) for $c$ and a suitable initial program state.

## Task 3: Axiomatic Semantics with Local Variables (30 Points)

Assume we extend the WHILE programming language with blocks whose local variables are initialized (procedures are not considered in the extension).

$$\begin{aligned} v &::= \mathsf{Var}\ x := e;\ v \mid \epsilon \qquad (e \text{ ranges over } \mathsf{AExp}) \\ c &::= \ \ldots \mid \mathsf{begin}\ v\ c\ \mathsf{end} \end{aligned}$$

(a) Let $A$ be an assertion with free variables $\mathsf{FV}(A)$. Define an assertion $A'$ in which every $x \in \mathsf{FV}(A)$ is replaced by a fresh existentially quantified variable $x'$ such that $\models (A \Rightarrow A')$ holds.

(b) Extend the rules of axiomatic semantics to capture the local variable declarations and block definitions. You may assume that a sequence $v$ of variable declarations contains no duplicates. For convenience, you may use $\mathsf{FV}(v)$ ( resp. $\mathsf{FV}(A)$) to denote the set of variables occuring in $v$ (resp. $A$) and $Exp(v)$ to denote the corresponding arithmetic expressions.