Exercise Sheet 4

General remarks:

- **Due date:** November 16th (before the exercise class).
- You can hand in your solutions at the start of the exercise class or via L2P. Please remember to provide your matriculation number. We kindly ask you to hand in your solutions in groups of **three**.
- Solutions must be written in English.
- While we will publish sketches of exercise solutions, we do *not* guarantee that these sketches contain all details that are necessary to properly solve an exercise. Hence, it is recommended to attend the exercise classes.
- If you have any questions regarding the lecture or the exercise, feel free to write us an email or visit us at the chair.

Exercise 1 (Program Verification with Weakest Pre-expectations) $\boxed{\textbf{25\%}}$

For each of the programs P and expectations $f \in \mathbb{E}$ below, compute $wp(P, f)$.

(a) [10%] $P\colon \texttt{if}(x = 0)\,\{\,\texttt{skip}\,\}\,\texttt{else}\,\{\,x := 1 + x\ [1/3]\ x := 6\,\}, \quad f = x.$

(b) [15%] $P\colon \{\,\texttt{while}(\texttt{true})\,\{\,x := 1\,\}\,\}\,[1/2]\,\{\,\texttt{diverge}\,\}, \quad f = 1.$

Exercise 2 (Conservative Pre-expectations) $\boxed{\textbf{30\%}}$

Prove that the weakest pre-expectation semantics is a conservative extension of Dijkstra's weakest pre-condition semantics (see lecture 7, slide 26). That is, show that for every GCL program P and every predicate $F \in \mathbb{P}$, we have

$$\underbrace{wp(P, [F])}_{\text{weakest pre-expectation}} \quad = \quad \underbrace{wp(P, F)}_{\text{weakest pre-condition}} \quad .$$

Exercise 3 (Liberal Invariants) $\boxed{\textbf{15\%}}$

Let $P_{loop} = \texttt{while}(G)\{P\}$ be a pGCL program. We call $I \in \mathbb{E}_{\leq 1}$ a *liberal invariant* of P_{loop} if $[G] \cdot I \sqsubseteq wlp(P, I)$. Show that for every liberal invariant I of P_{loop}, we have

$$I \quad \sqsubseteq \quad wlp(\,P_{loop},\,[\neg G] \cdot I\,)\ .$$

Exercise 4 (Non-deterministic Choice) $\boxed{\textbf{30\%}}$

Recall from lecture 6, slide 7 that Disjkstra's original guarded command language contained a *non-deterministic* choice statement $P_1 [] P_2$, which we largely ignored so far. Intuitively, this statement executes either program P_1 or program P_2, but there is *no* probability distribution underlying this choice. Two common models to deal with non-determinism are known as *demonic non-determinism* and *angelic non-determinism*, i.e. we either choose the "worst" (demonic) or "best" (angelic) option. In terms of weakest (liberal) pre-expectations, this translates to minimizing (demonic) or maximizing (angelic) an expected value. We call the corresponding transformers *dwp*, *awp*, *dwlp*, and *awlp*, respectively. More formally, for $f \in \mathbb{E}_{\leq 1}$, we define

$$\begin{aligned} dwp(P_1 [] P_2, f) &= \min\{dwp(P_1, f), dwp(P_2, f)\} && \text{(demonic wp)} \\ dwlp(P_1 [] P_2, f) &= \min\{dwlp(P_1, f), dwlp(P_2, f)\} && \text{(demonic wlp)} \\ awp(P_1 [] P_2, f) &= \max\{awp(P_1, f), awp(P_2, f)\} && \text{(angelic wp)} \\ awlp(P_1 [] P_2, f) &= \max\{awlp(P_1, f), awlp(P_2, f)\} && \text{(angelic wlp)} \end{aligned}$$

For all other pGCL statements, both *dwp* and *awp* coincide with *wp*. Analogously, *dwlp* and *awlp* coincide with *wlp*.

Show that for all pGCL programs P (including non-deterministic choice) and $f \in \mathbb{E}_{\leq 1}$,

$$dwp(P, f) \quad = \quad 1 - awlp(P, 1 - f)$$