# Assignment 9

**Exercises on lecture 9/chapter 9**

5 November 2024

We will work on the following exercises during the next exercise class. You can choose among the problems of sheet 4 – 9 to work one out and submit it individually for feedback.

**Exercise 9.1** — Complete the proof of theorem 9.5, by proving the remaining case of the induction that shows that

$$(\chi_1[\chi_2] \lhd (f, \sigma)) \Downarrow \sigma' \text{ iff } (\chi_1 \lhd (O\chi_2 f, \sigma)) \Downarrow \sigma'$$

**Exercise 9.2** — Prove lemma 9.7 by proving that the composition of natural transformations satisfies the unit and associativity laws.

**Exercise 9.3** — Let $F, G \colon \mathcal{C} \to \mathcal{D}$ be functors and $\alpha \colon F \to G$ a natural transformation. Show that a family of morphisms $\beta_X \colon GX \to FX$ indexed by objects in $\mathcal{C}$ with $\alpha_X \circ \beta_X = \mathrm{id}_{FX}$ and $\beta_X \circ \alpha_X = \mathrm{id}_{GX}$ for all $X$, gives a natural transformation $\beta \colon G \to F$ that is inverse to $\alpha$.

**Exercise 9.4** — Complete the proof of lemma 9.8 by proving that for all $\chi$, $\sigma$ and $\sigma'$,

$$(\chi \lhd (f, \sigma)) \Downarrow \sigma' \quad \text{implies} \quad (D\chi)(f)(\sigma) = \eta(\sigma')$$

by induction on derivations for the big-step operational semantics.

**Exercise 9.5** — Prove corollary 9.10 (full abstraction of `Imp`) from theorem 9.9 (equivalence of denotational and operational semantics).

**Problem 9.6** — The goal of this problem is to define a syntactic model $O'$ of `Imp` and relate it to the denotational model $D$. This new model is given on objects by $S\bullet^A = \text{Disc AExp}$, $S\bullet^B = \text{Disc BExp}$ and $\bullet^C = \text{Disc Com}$, and on morphisms by $(S\,\chi)u = \chi[u]$ for all contexts $\chi$ and expressions/commands $u$.

**a)** Prove that $S$ is a functor $\mathcal{C}_{\texttt{Imp}} \to \boldsymbol{\omega}\mathbf{CPO}$.

**b)** Show that there is a natural transformation $\gamma \colon S \to D$.

**c)** Analyse what it means to give a natural transformation $D \to S$ and prove that such a natural transformation cannot exist. (Hint: You can use that `Imp` can only implement computable functions.)