What to turn in

Turn in the assignment during class on the due date.

Problem 1: Partial Orders

In this problem to define a partial ordering over graphs. A graph is a pair (N, E) consisting of:

- A set of nodes N;
- A map $E : (N \times N) \rightarrow \{0, 1\}$, representing the edges. If $E(n, m) = 1$ there is an edge between nodes n and m; otherwise, there is no edge.

We define a binary relation $\sqsubseteq$ over the set of all graphs, such that $G_1 \sqsubseteq G_2$ iff:

- G_2 contains all of the nodes in G_1, and may contain additional nodes;
- Each edge in G_2 between the nodes of G_1 must also be present in G_1;
- There is no edge in G_2 from inside the nodes of G_1 to outside the nodes of G_1.

1a) Formally write this binary relation.

1b) Show that it is a partial order.

1c) Formally write what is the least upper bound of two graphs $\bigsqcup\{G_1, G_2\}$.

Problem 2: Continuous Functions

2a) Cpo's in which the partial order is the identity relation are said to be discrete. Precisely what kinds of functions are continuous from a cpo with bottom to a discrete cpo?

2b) Let $(D, \sqsubseteq)$ be a cpo and $g : D \rightarrow D$ a continuous function. Consider the set F of all continuous functions $f : D \rightarrow D$ with the property that:

$$f \circ g \circ f = g \circ f \circ g$$

Show that the set F with the pointwise ordering of functions forms a cpo.

2c) By the Fixed Point Theorem, each continuous function $f : D \rightarrow D$ over a cpo domain with bottom $(D, \sqsubseteq, \perp)$ has a least fixed point. Therefore, we can define a fixed point operator:

$$\begin{array}{rcl} \mathit{fix} & : & [D \rightarrow D] \rightarrow D \\ \mathit{fix}(f) & = & \text{least fixed point of } f \end{array}$$

where $[D \rightarrow D]$ is the domain of all continuous functions from D to D. Show that *fix* is continuous.

Note: Make sure you explain each step in your proofs of continuity.

Problem 3: Do-Until Loops

We extend IMP with **do-until** loops: $c ::= \ldots \;\mid\; \textbf{do}\ c\ \textbf{until}\ b$. Informally, this command first executes the loop body c, then tests the condition b. If that condition is true, then the loop terminates; otherwise, the execution proceeds to the next iteration.

3a) The denotational semantics of **do** c **until** b is the least fixed point of a higher order function Γ_d. Write the function Γ_d.

3b) Consider the following program:

```
do { X := X + 1;
     Y := Y + 2*X - 1;
} until (X = M)
```

Write the first three approximations of the loop semantics for this example: $\Gamma_d(\perp_{\Sigma\to\Sigma_\perp})$, $\Gamma_d^2(\perp_{\Sigma\to\Sigma_\perp})$, and $\Gamma_d^3(\perp_{\Sigma\to\Sigma_\perp})$. Make sure you instantiate the condition b and the command c with the particular condition and loop body in this example. You must simplify your results as much as possible.

3c) Generalize the results from the previous part by deriving $\Gamma_d^n(\perp_{\Sigma\to\Sigma_\perp})$ for the example in part (b). You must prove your result (if necessary, use induction on n).

Problem 4: Probabilistic Choice

We extend IMP by adding a probabilistic choice command $c_0\ []_p\ c_1$:

$$c \quad ::= \quad \textbf{skip} \;\mid\; X := a \;\mid\; c_0\,;c_1 \;\mid\; \textbf{if}\ b\ \textbf{then}\ c_0\ \textbf{else}\ c_1 \;\mid\; \textbf{while}\ b\ \textbf{do}\ c \;\mid\; c_0\ []_p\ c_1$$

The new command $c_0\ []_p\ c_1$ non-deterministically executes one of its component commands, with a certain probability: it executes c_0 with probability p, and c_1 with probability $1-p$. The execution of all the other commands in a given state is the same as in IMP.

Because of the nondeterminism in the program, the execution of a command may yield multiple final states. However, since the choice operator indicates the probability of its choices, we can determine the probability for each of the final states. In other words, we can compute a probability distribution for the final states. In this problem, we want to define a program semantics which describes distributions of probabilities over states. Let Δ and $\Delta_\perp$ be domains of state distributions over Σ and $\Sigma_\perp$:

$$\begin{aligned} \Delta &= \Sigma \to \mathbb{R} \\ \Delta_\perp &= \Sigma_\perp \to \mathbb{R} \end{aligned}$$

For a particular state distribution $\delta \in \Delta$, $\delta(\sigma)$ is the probability that the execution is in state σ. A value of 1 for $\delta(\sigma)$ means that state σ has 100% probability of occurring. If $\delta \in \Delta_\perp$, then $\delta(\perp)$ is the probability of non-termination. We want to define the denotational semantics of each command as a functions that maps an initial probability distribution, before the command, to a resulting probability distribution, after the execution of the command:

$$\mathcal{C}[\![c]\!] : \Delta \to \Delta_\perp$$

Expressions are the same as in IMP, hence we use the same semantics: $\mathcal{A}[\![a]\!] : \Sigma \to \mathbb{Z}$ and $\mathcal{B}[\![b]\!] : \Sigma \to \mathbb{T}$.

4a) Write the probabilistic denotational semantics $\mathcal{C}[\![c]\!] : \Delta \to \Delta_\perp$ for all the regular IMP commands and for the choice command $c_0\ []_p\ c_1$.

4b) Determine the state distribution after the execution of the following program under the initial distribution $\delta_0 = \{(\sigma_0, 1)\}$. You must prove your result using the semantics from part (a).

```
X := 0;
while (X < 3) do
  (X := X + 1  []0.5  X := X + 2)
```