CS 522: Programming Language Semantics
Homework-2 Solution

Jai Arora (jaia3@illinois.edu)

1. Let $b \in \texttt{BExp}$ be any boolean expression, $s_1, s_2 \in \texttt{Block}$ be any blocks and $s \in \texttt{Stmt}$ be any statement in $\texttt{IMP}$. Let $\bot_S$ denote the undefined state. Consider the following two statements,

$$\texttt{LHS} = (\texttt{if}(b)\ s_1\ \texttt{else}\ s_2)\ s$$
$$\texttt{RHS} = \texttt{if}(s)\ s_1\ s\ \texttt{else}\ s_2\ s$$

- $[\![\texttt{LHS}]\!]$: Consider $[\![\texttt{if}(b)\ s_1\ \texttt{else}\ s_2]\!]$

$$[\![\texttt{if}(b)\ s_1\ \texttt{else}\ s_2]\!] = \lambda\sigma. \begin{cases} [\![s_1]\!](\sigma) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ [\![s_2]\!](\sigma) & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases}$$

Then, $[\![(\texttt{if}(b)\ s_1\ \texttt{else}\ s_2)\ s]\!]$ will be $[\![s]\!] \circ [\![\texttt{if}(b)\ s_1\ \texttt{else}\ s_2]\!]$, which can be simplified to,

$$\begin{aligned} [\![(\texttt{if}(b)\ s_1\ \texttt{else}\ s_2)\ s]\!] &= \lambda\sigma.\ [\![s]\!]([\![\texttt{if}(b)\ s_1\ \texttt{else}\ s_2]\!](\sigma)) \\ &= \lambda\sigma.\ [\![s]\!] \left(\begin{cases} [\![s_1]\!](\sigma) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ [\![s_2]\!](\sigma) & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases} \right) \\ &= \lambda\sigma. \begin{cases} [\![s]\!]([\![s_1]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ [\![s]\!]([\![s_2]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ [\![s]\!](\bot_S) & \text{if } [\![b]\!](\sigma) = \bot \end{cases} \\ &= \lambda\sigma. \begin{cases} [\![s]\!]([\![s_1]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ [\![s]\!]([\![s_2]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases} \end{aligned}$$

Here, $[\![s]\!](\bot_S) = \bot_S$, since undefined input state leads to an undefined output state (in the semantics, the denotations of different programs are actually **strict functions**).

- $[\![\texttt{RHS}]\!]$: We get

$$[\![\texttt{if}(b)\ s_1\ s\ \texttt{else}\ s_2\ s]\!] = \lambda\sigma.\ \begin{cases} [\![s_1\ s]\!](\sigma) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ [\![s_2\ s]\!](\sigma) & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases}$$
$$= \lambda\sigma.\ \begin{cases} [\![s]\!]([\![s_1]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ [\![s]\!]([\![s_2]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases}$$

We can see that $[\![\texttt{LHS}]\!] = [\![\texttt{RHS}]\!]$, making these two statements equivalent.

2. Let $\bot_S$ be the undefined state. Let $b \in \texttt{BExp}$ be any boolean expression and $s \in \texttt{Stmt}$ be any statement. Let $s_w = \texttt{while}(b)\ s$ be a `while` loop. The function $\mathcal{F} : (State \rightharpoonup State) \to (State \rightharpoonup State)$ corresponding to s_w is:

$$F(\alpha)(\sigma) = \begin{cases} \alpha([\![s]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \texttt{t} \\ \sigma & \text{if } [\![b]\!](\sigma) = \texttt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases}$$

(a) We want to show that $\mathcal{F}$ satisfies the hypothesis of the Fixed Point Theorem, i.e., show that it is continuous. For that, we will show two things,

- Monotonicity: Let $\alpha, \beta \in [State \rightharpoonup State]$ be two partial functions, such that $\alpha \preceq \beta$. Here $\preceq$ is the informativeness relation on partial functions.

  We need to show that $\mathcal{F}(\alpha) \preceq \mathcal{F}(\beta)$. Let $\sigma \in Sigma$ be any state. We will do a case analysis on the values $\mathcal{F}(\alpha)(\sigma)$ and $\mathcal{F}(\beta)(\sigma)$:
  - $\mathcal{F}(\alpha)(\sigma) = \bot_S$ ($\mathcal{F}(\alpha)$ is undefined in σ): Then $\mathcal{F}(\alpha) \preceq \mathcal{F}(\beta)$ by definition.
  - $\mathcal{F}(\alpha)(\sigma) \neq \bot_S$ ($\mathcal{F}(\alpha)$ is defined in σ). The following subcases arise:

    * $[\![b]\!](\sigma) = \texttt{f}$: Then $\mathcal{F}(\alpha)(\sigma) = \sigma$, which is defined. This also means that $\mathcal{F}(\beta)(\sigma) = \sigma$, which is also defined. So, we get that $\mathcal{F}(\alpha)(\sigma) = \mathcal{F}(\beta)(\sigma)$, and both values are defined. So, $\mathcal{F}(\alpha) \preceq \mathcal{F}(\beta)$.

    * $[\![b]\!](\sigma) = \texttt{t}$: Then $\mathcal{F}(\alpha)(\sigma) = \alpha([\![s]\!](\sigma))$, which is defined. Now since $\alpha \preceq \beta$, $\beta([\![s]\!](\sigma))$ also has to be defined and $\alpha([\![s]\!](\sigma)) = \beta([\![s]\!](\sigma))$. From here, we get

    $$\begin{aligned} \mathcal{F}(\beta)(\sigma) &= \beta([\![s]\!](\sigma)) \\ &= \alpha([\![s]\!](\sigma)) \\ &= \mathcal{F}(\alpha)(\sigma) \end{aligned}$$

    So, we get that $\mathcal{F}(\alpha)(\sigma) = \mathcal{F}(\beta)(\sigma)$, and both values are defined. So, $\mathcal{F}(\alpha) \preceq \mathcal{F}(\beta)$.

So, this implies that $\mathcal{F}$ is monotonous.

- Let $\{\alpha_n \mid n \in \mathbb{N}\}$ be any chain $State \rightharpoondown State$. Now since $(State \rightharpoondown State, \preceq, \perp_{S \to S})$ is a pointed CPO, we know that the limit of this chain, $\sqcup\alpha_n$ exists.

  We need to show that $\mathcal{F}(\sqcup\alpha_n) \preceq \sqcup\mathcal{F}(\alpha_n)$. Let $\sigma \in State$ be any state. We will do a case analysis on the values $\mathcal{F}(\sqcup\alpha_n)(\sigma)$ and $(\sqcup\mathcal{F}(\alpha_n))(\sigma)$:

  - $\mathcal{F}(\sqcup\alpha_n)(\sigma) = \perp_S$ ($\mathcal{F}(\sqcup\alpha_n)$ is undefined in σ): Then $\mathcal{F}(\sqcup\alpha_n) \preceq \sqcup\mathcal{F}(\alpha_n)$ by definition.
  - $\mathcal{F}(\sqcup\alpha_n)(\sigma) \neq \perp_S$ ($\mathcal{F}(\sqcup\alpha_n)$ is defined in σ). The following subcases arise:

    * $[\![b]\!](\sigma) = \mathtt{f}$: Then $\mathcal{F}(\sqcup\alpha_n)(\sigma) = \sigma$, which is defined. Also, for all $i \in \mathbb{N}$, $\mathcal{F}(\alpha_i)(\sigma) = \sigma$, which is also defined. Since all functions in the chain $\{\mathcal{F}(\alpha_n) \mid n \in \mathbb{N}\}$ are defined at σ and have the same value, their LUB $\sqcup\mathcal{F}(\alpha_n)$ is also defined and $(\sqcup\mathcal{F}(\alpha_n))(\sigma) = \sigma$.

      So, we get that $\mathcal{F}(\sqcup\alpha_n)(\sigma) = (\sqcup\mathcal{F}(\alpha_n))(\sigma)$, and both values are defined. So, $\mathcal{F}(\sqcup\alpha_n) \preceq \sqcup\mathcal{F}(\alpha_n)$.

    * $[\![b]\!](\sigma) = \mathtt{t}$: Then $\mathcal{F}(\sqcup\alpha_n)(\sigma) = (\sqcup\alpha_n)([\![s]\!](\sigma))$, which is defined. Since the LUB of the chain $\{\alpha_n\}$ is defined at $[\![s]\!](\sigma)$, there must exist some $k \in \mathbb{N}$ such that,

      $$\begin{aligned} &\forall j \geq k,\ \alpha_j([\![s]\!](\sigma)) = (\sqcup\alpha_n)([\![s]\!](\sigma)),\ \text{and is defined, and} \\ &\forall i < k,\ \alpha_i([\![s]\!](\sigma)) = \perp_S \end{aligned}$$

      This means that since the LUB is defined at $[\![s]\!](\sigma)$, there must be some j where the functions in the chain started getting defined at $[\![s]\!](\sigma)$ and had the same value, and the functions before this were undefined at $[\![s]\!](\sigma)$. If this was not true, $\sqcup\alpha_n$ would not have been the LUB of the chain $\{\alpha_n\}$.

      Let $j \geq k$. Then $\mathcal{F}(\alpha_j)(\sigma) = \alpha_j([\![s]\!](\sigma)) = (\sqcup\alpha_n)([\![s]\!](\sigma))$. So, this value is defined for any such j. This means that the LUB of the chain $\{\mathcal{F}(\alpha_n)\}$ is defined at σ and $(\sqcup\mathcal{F}(\alpha_n))(\sigma) = (\sqcup\alpha_n)([\![s]\!](\sigma))$.

      So, we get that $\mathcal{F}(\sqcup\alpha_n)(\sigma) = (\sqcup\alpha_n)([\![s]\!](\sigma)) = (\sqcup\mathcal{F}(\alpha_n))(\sigma)$, and both values are defined. So, $\mathcal{F}(\sqcup\alpha_n) \preceq \sqcup\mathcal{F}(\alpha_n)$.

From these two points, we have proved that $\mathcal{F}$ is continuous.

(b) Let $w_k \in [State \rightharpoondown State]$ be defined as

$$w_k(\sigma) = \begin{cases} [\![s]\!]^i(\sigma) & \exists 0 \leq i < k,\ [\![b]\!]([\![s]\!]^i(\sigma)) = \mathtt{f}, \forall 0 \leq j < i,\ [\![b]\!]([\![s]\!]^j(\sigma)) = \mathtt{t} \\ \perp_S & \text{otherwise} \end{cases}$$

We need to prove w_k is well defined, that is, if such an i exists, then it is unique.

Proof. We will prove this by contradiction. Let's assume that w_k is not well defined, then there exists $\sigma \in State$ such that $w_k(\sigma) = \sigma_1$ and $w_k(\sigma) = \sigma_2$, where σ_1, σ_2 are defined and $\sigma_1 \neq \sigma_2$. From definition of w_k, we know that

- $\sigma_1 = [\![s]\!]^{i_1}(\sigma)$, where
  - $0 \leq i_1 < k$
  - $[\![b]\!]([\![s]\!]^{i_1}(\sigma)) = \mathtt{f}$
  - $\forall 0 \leq j_1 < i_1,\ [\![b]\!]([\![s]\!]^{j_1}(\sigma)) = \mathtt{t}$
- $\sigma_2 = [\![s]\!]^{i_2}(\sigma)$, where
  - $0 \leq i_2 < k$
  - $[\![b]\!]([\![s]\!]^{i_2}(\sigma)) = \mathtt{f}$
  - $\forall 0 \leq j_2 < i_2,\ [\![b]\!]([\![s]\!]^{j_2}(\sigma)) = \mathtt{t}$

Since, $\sigma_1 \neq \sigma_2$, it implies $i_1 \neq i_2$. WLOG, assume that $i_1 > i_2$. Then from σ_1, we get that $[\![b]\!]([\![s]\!]^{i_2}(\sigma)) = \mathtt{t}$, but from σ_2, we get $[\![b]\!]([\![s]\!]^{i_2}(\sigma)) = \mathtt{f}$, which is a contradiction.

So, w_k is well defined for all k. □

(c) We need to prove that $w_k = \mathcal{F}^k(\bot_{S\to S})$, where $\bot_{S\to S}$ is the bottom of the $[State \rightharpoonup State]$ CPO.

Proof. We will proof this by induction on k.

- <u>Basis</u>: $k = 0$. Then $w_0(\sigma) = \bot_S$ for all $\sigma \in State$. Since w_0 maps all states to the bottom state, $w_0 = \bot_{S\to S} = \mathcal{F}^0\bot_{S\to S}$
- <u>Induction Step</u>: $k = n+1, n \geq 0$.

  <u>IH</u>: $w_n = \mathcal{F}^n(\bot_{S\to S})$

  It is easy to see that $F^{n+1}(\bot_{S\to S}) = \mathcal{F}(\mathcal{F}^n(\bot_{S\to S}))$. Using the IH, we get $F^{n+1}(\bot_{S\to S}) = \mathcal{F}(w_n)$. So, we need to prove that $w_{n+1} = \mathcal{F}(w_n)$.
  Let $\sigma \in State$ be any state. $\mathcal{F}(w_n)(\sigma)$ has the following form:

$$F(w_n)(\sigma) = \begin{cases} w_n([\![s]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \mathtt{t} \\ \sigma & \text{if } [\![b]\!](\sigma) = \mathtt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases}$$

  Consider the following cases:
  - $[\![b]\!](\sigma) = \bot$: Then $w_{n+1}(\sigma) = \bot_S = \mathcal{F}(w_n)(\sigma)$. So, $w_{n+1} = \mathcal{F}(w_n)$.
  - $[\![b]\!](\sigma) = \mathtt{f}$: Then $i = 0$ in w_{n+1} and $w_{n+1}(\sigma) = \sigma$. Also, $\mathcal{F}(w_n)(\sigma) = \sigma$. So, $w_{n+1} = \mathcal{F}(w_n)$.

- $[\![b]\!](\sigma) = \mathtt{t}$: Then $\mathcal{F}(w_n)(\sigma) = w_n([\![s]\!](\sigma))$. Also, for w_{n+1}, $i > 0$. Then $w_{n+1}(\sigma)$ can be rewritten as,

$$
\begin{aligned}
w_{n+1}(\sigma) &= \begin{cases} [\![s]\!]^{i-1}[\![s]\!](\sigma) & \exists 0 \leq i-1 < n,\ [\![b]\!]([\![s]\!]^{i-1}[\![s]\!](\sigma)) = \mathtt{f}, \\ & \forall 0 \leq j < i-1,\ [\![b]\!]([\![s]\!]^{j-1}[\![s]\!](\sigma)) = \mathtt{t} \\ \bot_S & \text{otherwise} \end{cases} \\
&= w_n([\![s]\!](\sigma))
\end{aligned}
$$

So, $w_{n+1} = \mathcal{F}(w_n)$.

So, using the principal of mathematical induction, we prove that $w_k = \mathcal{F}^k(\bot_{S \to S})$. □

3. Let $\bot_S$ be the undefined state. Let $b \in \mathtt{BExp}$ be any boolean expression and $s \in \mathtt{Stmt}$ be any statement. Let $s_w = \mathtt{while}(b)\ s$ be a `while` loop. The functions $\mathcal{F} : (\mathit{State} \rightharpoonup \mathit{State}) \to (\mathit{State} \rightharpoonup \mathit{State})$ corresponding to s_w is:

$$
F(\alpha)(\sigma) = \begin{cases} \alpha([\![s]\!](\sigma)) & \text{if } [\![b]\!](\sigma) = \mathtt{t} \\ \sigma & \text{if } [\![b]\!](\sigma) = \mathtt{f} \\ \bot_S & \text{if } [\![b]\!](\sigma) = \bot \end{cases}
$$

The semantics of s_w can be described using the least fixed point of $\mathcal{F}$, $\mathsf{fix}(\mathcal{F})$. Let $\sigma \in \mathit{State}$ be any state. $\mathbf{fix}(\mathcal{F})(\sigma)$ could be undefined due to the following reasons:

- Type 1: $\sigma = \bot_S$. The input state itself is undefined
- Type 2: $[\![b]\!](\sigma) = \bot$. There are some illegal operations in b, could be a divide-by-zero, or the use of an undeclared variable
- Type 3: the loop does not terminate in the state σ

It is obvious that any other fixed point of $\mathcal{F}$, say g, has to have at least as much information as $\mathsf{fix}(\mathcal{F})$ ($\mathsf{fix}(\mathcal{F}) \preceq g$). So, g will stay the same for the states in which $\mathsf{fix}(\mathcal{F})$ is defined, and it can defined some states in which the loop is undefined. These could correspond to any of the 3 types above. But, g can only define those states in which the loop does not terminate (type 3), and not the other states, because then g will not be a fixed point of $\mathcal{F}$.