CS611 Lecture 12: Fixed Points and Complete Partial Orders September 24, 1999
Nate Nystrom

---

1 Correspondence between operational and denotational semantics for IMP

Recall the following definitions for IMP:

$$\begin{array}{rcl}\mathcal{A} & : & \mathrm{Aexp} \to \Sigma \to \mathbb{N} \\ \mathcal{B} & : & \mathrm{Bexp} \to \Sigma \to \mathbb{T} \\ \mathcal{C} & : & \mathrm{Com} \to \Sigma \rightharpoonup \Sigma\end{array}$$

There is a correspondence between denotational semantics and operational semantics. First note that commands in IMP are deterministic; that is, for a given σ, there is only one σ' for the transition $\langle c, \sigma\rangle \to \sigma'$. We can thus say that commands are functions from states to states. Similarly, expressions are functions from states to integers or truth values.

We would like to show the following correspondence between denotational semantics and operation semantics:

$$\begin{array}{rcl}\mathcal{A}[\![a]\!]\sigma = n & \iff & \langle a, \sigma\rangle \to n \\ \mathcal{B}[\![b]\!]\sigma = t & \iff & \langle b, \sigma\rangle \to t \\ \mathcal{C}[\![c]\!]\sigma = \sigma' & \iff & \langle c, \sigma\rangle \to \sigma'\end{array}$$

There is an inductive argument for this in the book, but, as an example, we can translate the rule

$$\frac{\langle a_0, \sigma\rangle \to n_0 \quad \langle a_1, \sigma\rangle \to n_1}{\langle a_0 + a_1, \sigma\rangle \to n_0 + n_1}$$

into

$$\frac{\mathcal{A}[\![a_0]\!]\sigma = n_0 \quad \mathcal{A}[\![a_1]\!]\sigma = n_1}{\mathcal{A}[\![a_0 + a_1]\!]\sigma = n_0 + n_1}$$

which is just another way to write

$$\mathcal{A}[\![a_0 + a_1]\!]\sigma = \mathcal{A}[\![a_0]\!]\sigma + \mathcal{A}[\![a_1]\!]\sigma$$

Another example:

$$\frac{\langle c_0, \sigma\rangle \to \sigma'' \quad \langle c_1, \sigma''\rangle \to \sigma'}{\langle c_0; c_1, \sigma\rangle \to \sigma'}$$

translates to

$$\frac{\mathcal{C}[\![c_0]\!]\sigma = \sigma'' \quad \mathcal{C}[\![c_1]\!]\sigma = \sigma'}{\mathcal{C}[\![c_0; c_1]\!]\sigma = \sigma'}$$

which is a longer way to write

$$\mathcal{C}[\![c_0; c_1]\!]\sigma = \mathcal{C}[\![c_1]\!](\mathcal{C}[\![c_0]\!]\sigma)$$

using σ'' as temporary variable for the subexpression $\mathcal{C}[\![c_0]\!]\sigma$. We can do similar translations for almost every rule in the operational semantics for IMP to a corresponding denotational rule. The exception is the rule for **while**.

2 Partial functions and denotational semantics

The advantages of denotational semantics is that

- It is more compact and more convienient because we can treat relations as functions.

- It corresponds more directly to what a compiler is doing.

Unfortunately, we are stuck with the partial function $\mathcal{C}$

$$\mathcal{C} : \mathrm{Com} \to \Sigma \rightharpoonup \Sigma$$

Partial functions are more difficult to reason about, but we need partial functions to solve equations for **while** and for recursive equations in general. We would much rather have total functions.

3 $\mathcal{C}$ as a total function

Recall that **while** is defined using a least fixed point. We will generalize how to find least fixed points so that we get total functions rather than partial functions. The trick is to introduce a new type of store, called bottom and denoted $\perp$. $\perp$ is the result of all commands which diverge. Using $\perp$, we can now define $\mathcal{C}$ as a total function

$$\mathcal{C} : \mathrm{Com} \to \Sigma_\perp \to \Sigma_\perp$$

For example,

$$\mathcal{C}[\![\textbf{while true do skip}]\!]\sigma = \perp.$$

A more complicated example is the IMP program **while** $a = 0$ **do skip**. Without $\perp$, we have the rule:

$$\mathcal{C}[\![\textbf{while } a = 0 \textbf{ do skip}]\!]\sigma = \{(\sigma, \sigma) \mid \sigma(a) \neq 0\}$$

which is undefined for $\sigma(a) = 0$. With $\perp$, we can change the rule to

$$\mathcal{C}[\![\textbf{while } a = 0 \textbf{ do skip}]\!]\sigma = \{(\sigma, \sigma) \mid \sigma(a) \neq 0\} \cup \{(\sigma, \perp) \mid \sigma(a) = 0\}.$$

In general, the old rule for **while** was

$$\mathcal{C}[\![\textbf{while } b \textbf{ do } c]\!]\sigma = \textbf{fix}\, \Gamma$$

where

$$\Gamma = \lambda f : \Sigma \to \Sigma \,.\, \lambda\sigma : \Sigma \,.\, \textbf{if } \mathcal{B}[\![b]\!]\sigma \textbf{ then } f(\mathcal{C}[\![c]\!]\sigma) \textbf{ else } \sigma$$

Since we have a construction of least fixed points of sets, we can evaluate the fixed point of Γ if we consider functions as sets. However, this construction relies on rule induction from $\emptyset$:

$$\emptyset \to R(\emptyset) \to R(R(\emptyset)) \to \cdots \to R^n(\emptyset)$$

If functions are total, the function that corresponds to $\perp$ is

$$\{(\sigma, \perp) \mid \sigma \in \Sigma\}$$

There is no $\emptyset$.

4 Fixing fix

4.1 Review of rule induction

To fix **fix**, recall rule induction. Let A be a set of transitions. We define an operator R that, when applied to A, returns a new set of transitions A'.

$$R(A) = A'$$

Rules *instances* can be expressed in the form

$$\frac{x_1 \quad x_2 \quad \cdots \quad x_n}{x}.$$

where $x_1, \ldots, x_n$, are potential members of the set being constructed, and x is a member of the set if all of the x_i are. Note that rule instances are obtained for a given rule by substituting into the rule all possible set members that satisfy the constraints associated with the rule.

When we apply R to a set A, we get

$$R(A) = \{x \mid \exists \frac{x_1 x_2 \cdots x_n}{x} \,.\, \{x_1, x_2, \ldots, x_n\} \in A\}$$

$R(\emptyset)$ is the set of all the transitions that we can conclude from the axioms, i.e., rules of the form $\overline{\;x\;}$. $R(R(\emptyset))$ is the set of all the transitions that we can conclude from the axioms and one application of the rules.

We can argue inductively that

$$\emptyset \subseteq R(\emptyset) \subseteq R(R(\emptyset)) \subseteq \cdots \subseteq R^n(\emptyset) \subseteq \cdots$$

The function R is monotonic in the sense that if $A \subseteq B$, then $R(A) \subseteq R(B)$. Then we can see inductively that $R^n(\emptyset) \subseteq R^{n+1}(\emptyset)$. Clearly $R^0(\emptyset) \subseteq R^1(\emptyset)$ because $R^0(\emptyset) = \emptyset$. Then assuming $R^{n-1}(\emptyset) \subseteq R^n(\emptyset)$, we apply R to both sides of the equation to obtain $R^n(\emptyset) \subseteq R^{n+1}(\emptyset)$. By induction and transitivity, the sets $R^n(\emptyset)$ have the property stated. Finally, we can take $\bigcup_{n \in N} R^n(\emptyset)$ as the least fixed point of R. It was showed in Lecture 7 that this set is a fixed point of R; the book proves that it is the least fixed point. We will also show that it is the least fixed point in the next lecture.

4.2 Complete partial orders

The generalization of sets that we will use to develop a better denotational semantics is that of a *complete partial order*, which is a partial order with one additional property. First let us start with the definition of a partial order.

Definition 1 A partial order is a set P and an ordering relation $\sqsubseteq$ such that for all $a, b, c \in P$,

- $a \sqsubseteq a$ (reflexive)
- $a \sqsubseteq b \wedge b \sqsubseteq c \Rightarrow a \sqsubseteq c$ (transitive)
- $a \sqsubseteq b \wedge b \sqsubseteq a \Rightarrow a = b$ (antisymmetric)

Some examples of partial orders are:

- $(\mathbb{Z}, \leq)$
- $(\mathbb{Z}, =)$ is a partial order.
- $(2^S, \subseteq)$ is a partial order.

$(\mathbb{Z}, <)$ is not a partial order since it is not reflexive.

Definition 2 A *chain* in a partial order $(P, \sqsubseteq)$ is a sequence, $a_1, a_2, a_3, \ldots$, where each $a_i \in P$, such that

$$a_1 \sqsubseteq a_2 \sqsubseteq a_3 \sqsubseteq \cdots$$

Definition 3 For a partial order $(P, \sqsubseteq)$, an upper bound of a set $S \subseteq P$ is $b \in P$ such that for all $a \in S$, $a \sqsubseteq b$.

Definition 4 For a partial order $(P, \sqsubseteq)$, the least upper bound of a set $S \subseteq P$ is the upper bound of S, b', such that

$$\forall b \,.\, (\forall a \,.\, a \in S \wedge a \sqsubseteq b) \Rightarrow b' \sqsubseteq b.$$

For a set $\{a_1, a_2, \ldots, a_n, \ldots\}$, we write

$$\bigsqcup_i a_i = b'$$

Definition 5 A lattice is a partial order $(P, \sqsubseteq)$ such that for all finite subsets $S \subseteq P$, a least upper bound is defined for S.

Definition 6 A complete lattice is a partial order $(P, \sqsubseteq)$ such that for all infinite subsets $S \subseteq P$, a least upper bound is defined for S.

Definition 7 A complete partial order is a partial order $(P, \sqsubseteq)$ such that for all chains $S \subseteq P$, a least upper bound is defined for S.

Note that all partial orders defined over a finite set P are complete, because any chain must necessarily be finite, and the top element of the chain is the least upper bound for that chain. Examples of partial orders that are not complete are necessarily infinite in size. Here are some examples of partial orders that are *not* complete:

- The set of all natural numbers, with the ordering relation $\leq$. The chain $0, 1, 2, 3, \ldots$ has no least upper bound. However, if we add an extra element ∞ that is greater than every natural number, the partial order is then complete.
- The open interval (0,1), with the ordering relation $\leq$. The chain $\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots$ has no least upper bound because 1 is not in the open interval. The closed interval [0,1] is a complete partial order.