Scribe: Ross Fleming, Frances Spalding Lecturer: Radu Rugina

1 Review

Recall our while loop from last time, $w \equiv \textbf{while}\ b\ \textbf{do}\ c$. We were trying to prove that $w \sim \textbf{if}\ b\ \textbf{then}\ (c; w)\ \textbf{else skip}$. We defined $\mathcal{C}[\![w]\!]$ as follows:

$$\begin{aligned} \mathcal{C}[\![w]\!] \quad = \quad & \{(\sigma, \sigma) \mid \neg\mathcal{B}[\![b]\!]\sigma\} \cup \\ & \{(\sigma, \perp) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma = \perp\} \cup \\ & \{(\sigma, \mathcal{C}[\![w]\!](\mathcal{C}[\![c]\!]\sigma)) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma \neq \perp\} \end{aligned}$$

This is not a definition because $\mathcal{C}[\![w]\!]$ occurs on both sides of the equation. Instead, this is a recursive equation. We would like to know if there is a solution, and if so, if it is unique.

2 Using Induction Rules

We can rewrite $\mathcal{C}[\![w]\!]$ using inductive rules.

$$\frac{}{(\sigma, \sigma)}\ \neg\mathcal{B}[\![b]\!]\sigma$$

$$\frac{}{(\sigma, \perp)}\ \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!] = \perp$$

$$\frac{(\mathcal{C}[\![c]\!]\sigma, \sigma')}{(\sigma, \sigma')}\ \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma \neq\perp$$

Recall the rule operator from a couple of lectures ago.

$$\hat{R}(X) = \{y \mid \exists X' \subseteq X\ .\ \frac{X'}{y}\}$$

$$A = \bigcup_{n \in \mathbb{N}} \hat{R}^n(\emptyset)$$

We can apply the rule operator to our inductive definition.

$$\begin{aligned} \hat{R}(\emptyset) \quad = \quad & \{(\sigma, \sigma) \mid \neg\mathcal{B}[\![b]\!]\sigma\} \cup \\ & \{(\sigma, \perp) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma = \perp\} \end{aligned}$$

This is a partial function, because if the loop executes at least once we don't have a mapping. We can extend the partial function to a total function by mapping σ to $\perp$ if the function is not defined for σ.

$$\begin{aligned} \hat{R}^2(\emptyset) = \hat{R}(\hat{R}(\emptyset)) \quad = \quad & \{(\sigma, \sigma) \mid \neg\mathcal{B}[\![b]\!]\sigma\} \cup & (1) \\ & \{(\sigma, \perp) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma = \perp\} \cup & (2) \\ & \{(\sigma, \mathcal{C}[\![c]\!]\sigma) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma \neq \perp \wedge \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\sigma\} \cup & (3) \\ & \{(\sigma, \perp) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma \neq \perp \wedge \mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\sigma \wedge \mathcal{C}[\![c]\!]\mathcal{C}[\![c]\!]\sigma = \perp\} & (4) \end{aligned}$$

In the above equation, line 1 represents the case where the condition is initially false and so we have the identity function. Line 2 is the case where the condition is initially true, but the command inside the while loop never terminates. Line 3 models exactly one iteration of the loop. Line 4 is the case where the command inside the loop does not terminate on the second iteration.

[Note that $\emptyset$ is a partial function that is undefined for all values in the domain. Extending it to a total function means that it always returns $\perp : \emptyset \equiv \lambda x.\perp$]

Looking at this a different way, we can rewrite $\hat{R}$ as the function $\Gamma : (\Sigma \to \Sigma_\perp) \to (\Sigma \to \Sigma_\perp)$.

$$\begin{array}{rcl}\Gamma(\gamma) & = & \{(\sigma, \sigma) \mid \neg\mathcal{B}[\![b]\!]\sigma\} \cup \\ & & \{(\sigma, \perp) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma = \perp\} \cup \\ & & \{(\sigma, \gamma(\mathcal{C}[\![c]\!]\sigma)) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma \neq \perp\}\end{array}$$

The application of Γ (a higher-order function) is the same as applying the rule operator (which inductively defines sets).

$$\begin{array}{rcl}\perp_{\Sigma\to\Sigma_\perp} & = & \lambda x.\perp \\ \Gamma(\perp_{\Sigma\to\Sigma_\perp}) & = & \{(\sigma, \sigma) \mid \neg\mathcal{B}[\![b]\!]\sigma\} \cup \\ & & \{(\sigma, \perp) \mid \mathcal{B}[\![b]\!]\sigma\} \\ \Gamma^2(\perp_{\Sigma\to\Sigma_\perp}) & = & \{(\sigma, \sigma) \mid \neg\mathcal{B}[\![b]\!]\sigma\} \cup \\ & & \{(\sigma, \mathcal{C}[\![c]\!]\sigma) \mid \mathcal{B}[\![b]\!]\sigma \wedge \mathcal{C}[\![c]\!]\sigma \neq \perp \wedge \neg\mathcal{B}[\![b]\!](\mathcal{C}[\![c]\!]\sigma)\} \cup \\ & & \{(\sigma, \perp) \mid otherwise\}\end{array}$$

At step n, Γ^n models the execution of at most n iterations of the loop and returns $\perp$ if the loop executes at least n iterations. Note that in this setting, you should think of $\perp$ as "undefined", rather than "nontermination."

Recall the two definitions for inductively defined sets:

1. $A = \bigcup_{n\in\mathbb{N}} \hat{R}^n(\emptyset)$
   $\emptyset \subseteq \hat{R}(\emptyset) \subseteq \hat{R}^2(\emptyset) \subseteq \ldots$
2. A is the least fixed point ($\hat{R}(A) = A$) of $\hat{R}$
   $A = fix(\hat{R})$
   We define the function fix as follows:
   fix: $(D \to D) \to D$
   fix(f) = least fixed point of f

We can use similar definitions to define $\mathcal{C}[\![w]\!]$:

1. $\mathcal{C}[\![w]\!] = \bigsqcup_{n\in\mathbb{N}} \Gamma^n(\perp_{\Sigma\to\Sigma_\perp})$
   $\perp_{\Sigma\to\Sigma_\perp} \sqsubseteq \Gamma(\perp_{\Sigma\to\Sigma_\perp}) \sqsubseteq \Gamma^2(\perp_{\Sigma\to\Sigma_\perp}) \sqsubseteq \ldots$
2. $\mathcal{C}[\![w]\!] = fix(\Gamma)$

But we don't know how to take the "union" of functions ($\bigsqcup$) or what "$\sqsubseteq$" means in this case. Such operators and relations can be generally defined in the context of partially ordered sets, which we will discuss next.

3 Partial Orders

A partially ordered set (poset) is a pair (S,$\sqsubseteq$) consisting of a set S and a binary relation $\sqsubseteq$ on that set , such that the order relation $\sqsubseteq$ is:

- reflexive: $\forall x \in S \,.\, x \sqsubseteq x$
- antisymmetric: $\forall x, y \in S \,.\, (x \sqsubseteq y \wedge y \sqsubseteq x) \Rightarrow x = y$
- transitive: $\forall x, y, z \in S \,.\, (x \sqsubseteq y \wedge y \sqsubseteq z) \Rightarrow x \sqsubseteq z$

[Note: $\sqsubseteq$ is read as "less-than"]
Partial orders are called "partial" because there may be pairs of elements that are not comparable. Totally ordered sets have the same three properties as partially ordered sets, plus the property that $(\forall x, y \,.\, x \sqsubseteq y \vee y \sqsubseteq x)$. We will get to total orders later in the class.

Examples of partial orders:

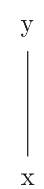
- $(\mathbb{N}, \leq)$, $(\mathbb{Z}, \leq)$ are posets
- $(\mathbb{N}, =)$ is a poset (also called a flat order)
- $(\mathcal{P}(x), \subseteq)$ is a poset (where $\mathcal{P}(x)$ is the powerset of x)
- If $(S, \sqsubseteq)$ is a poset, then $(S, \sqsupseteq)$ is also a poset where $x \sqsupseteq y \iff y \sqsubseteq x$
- ranges: $R = \{(a, b) \mid a \in \mathbb{Z}, b \in \mathbb{Z}, a \leq b\}$
  $[(a_0, b_0) \sqsubseteq (a_1, b_1) \iff a_1 \leq a_0 \wedge b_0 \leq b_1] \iff (R, \sqsubseteq)$ is a poset
- If $(A, \sqsubseteq_A)$, $(B, \sqsubseteq_B)$ are partial orders, then we can define partial order over the cartesian product $(A \times B, \sqsubseteq)$ as follows: $(a_0, b_0) \sqsubseteq (a_1, b_1) \iff (a_0 \sqsubseteq_A a_1) \wedge (b_0 \sqsubseteq_B b_1)$
  In the above example with ranges, $(A, \sqsubseteq_A) \equiv (\mathbb{Z}, \geq)$ and $(B, \sqsubseteq_B) \equiv (Z, \leq)$.
- $(\mathbb{N}, |)$ is a poset, where $a \mid b \iff \exists k \in \mathbb{N} \,.\, b = ka$
- functions: if $(B, \sqsubseteq_B)$ is a poset, then $(A \rightarrow B, \sqsubseteq)$ is also a poset where
  $(f_1 \sqsubseteq f_2 \iff \forall x \in A \,.\, f_1(x) \sqsubseteq_B f_2(x))$

Examples that are not partial orders:

- $(\mathbb{N}, <)$ (because it is not reflexive)
- $(\mathbb{Z}, |)$ (because it is not antisymmetric)

4 Hasse Diagrams

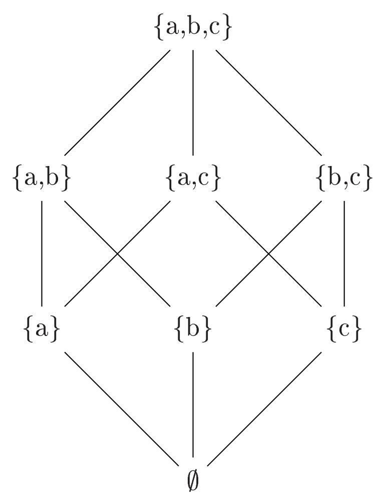
Partial orders can be represented by Hasse Diagrams. Each element in the set is a node in the diagram. If $x \sqsubseteq y$, then x appears below y in the diagram.



There is an edge between x and y if $x \sqsubseteq y$ and $\nexists z$ s.t. $x \sqsubseteq z \wedge z \sqsubseteq y$ and $z \neq x \wedge z \neq y$. In other words, there are no edges in the graph for relations that can be inferred through transitivity.

Examples of Hasse Diagrams:

- $(\mathcal{P}(\{a, b, c\}), \subseteq)$



  Note: $\{b, c\}$ and $\{a\}$ are not comparable, so there is no path between them in the diagram.

- $(\mathbb{Z}, \leq)$

$$\begin{array}{c} \vdots \\ | \\ 2 \\ | \\ 1 \\ | \\ 0 \\ | \\ -1 \\ | \\ -2 \\ | \\ \vdots \end{array}$$

  (This is also a total order.)

- $(\mathbb{Z}, =)$

$$\cdots \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad \cdots$$

  There are no edges between the elements because there are no relations between the elements. This is called a flat order.