CS 611
Advanced Programming Languages

Andrew Myers
Cornell University

Lecture 11
Fixed Points and CPOs
18 Sep 00

Administration

- Homework 2 due in one week (25th)
- Gunter, Mitchell, Stoy placed on reserve in Engineering library

Last time

- Denotational semantics: meaning function $\mathcal{C}[\![e]\!]$ maps language syntax e into domain with 'intrinsic' meaning
- Defined semantics for IMP through meaning functions $\mathcal{A}[\![a]\!], \mathcal{B}[\![a]\!], \mathcal{C}[\![c]\!]$
- Meaning functions defined as *syntax-directed translation*

$$\mathcal{A}[\![a_0+a_1]\!] = \lambda\sigma.\mathcal{A}[\![a_0]\!]\sigma+\mathcal{A}[\![a_1]\!]\sigma \iff \frac{\langle a_0, f_0\rangle \quad \langle a_1, f_1\rangle}{\langle a_0+a_1,\ \lambda\sigma.\ f_0\sigma+f_1\sigma\rangle}$$

Problem: while

$\mathcal{C}[\![\textbf{while } b \textbf{ do } c]\!]\sigma = if\ \neg\mathcal{B}[\![b]\!]\sigma\ then\ \sigma$
else $\mathcal{C}[\![\textbf{while } b \textbf{ do } c]\!]\,(\mathcal{C}[\![c]\!]\,\sigma)$

$$\frac{\langle b, f_b\rangle \quad \langle \textbf{while } b \textbf{ do } c, f_w\rangle \quad \langle c, f_c\rangle}{\langle \textbf{while } b \textbf{ do } c,\ \lambda\sigma\ .\ if\ f_b\sigma\ then\ \sigma\ else\ f_w(f_c\sigma)\rangle}$$

- No finite proof tree for $\langle \textbf{while } b \textbf{ do } c, f\rangle$!
- Definitions of denotations will be total if based on structural induction

Denotation of while

$\mathcal{C}[\![\text{while } b \text{ do } c]\!]$ is solution to

$x = \Gamma(x)$ with

$\Gamma = \lambda f \in \Sigma_\perp \to \Sigma_\perp.\ if\ \neg\mathcal{B}[\![b]\!]\ \sigma\ then\ \sigma\ else\ f\,(\mathcal{C})$

Idea: $\mathcal{C}[\![\text{while } b \text{ do } c]\!]\ \sigma =$

$fix\,(\Gamma)$

$= fix\,(\,\lambda f \in \Sigma_\perp \to \Sigma_\perp.\ if\ \neg\mathcal{B}[\![b]\!]\ \sigma\ then\ \sigma\ else\ f\,(\mathcal{C}[\![c]\!]\sigma))$

- What fixed point do we want (least)?
- How do we define least fixed point operator *fix*?

Approximations

- Consider sequence of approximations to denotation of **while**:

1. $\lambda\sigma.if\ \neg\mathcal{B}[\![b]\!]\ \sigma\ then\ \sigma\ else$ ... (works for 0 iterations)
2. $\lambda\sigma.if\ \neg\mathcal{B}[\![b]\!]\ \sigma\ then\ \sigma\ else$
   $if\ \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\sigma\ then\ \mathcal{C}[\![c]\!]\sigma\ else$ ...
   (0 or 1 iterations)
3. $\lambda\sigma.if\ \neg\mathcal{B}[\![b]\!]\ \sigma\ then\ \sigma\ else$
   $if\ \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\sigma\ then\ \mathcal{C}[\![c]\!]\sigma\ else$
   $if\ \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\mathcal{C}[\![c]\!]\sigma\ then\ \mathcal{C}[\![c]\!]\mathcal{C}[\![c]\!]\sigma\ else$ ...
   (0-2 iterations)

- "limit" of this sequence is denotation of **while**

Orderings

- Fixed points of denotation of **while** differ only in case of non-termination
- We want $\mathcal{C}[\![\textbf{while true do skip}]\!]\,\sigma = \bot$
- *Idea:* define ordering on fixed points of Γ such that *least* fixed point is the one we want
- Compare to inductive definitions
  - ordering was $\subseteq$
  - doesn't work here: how to order elements of $\Sigma_\bot \to \Sigma_\bot$?

Partial orders

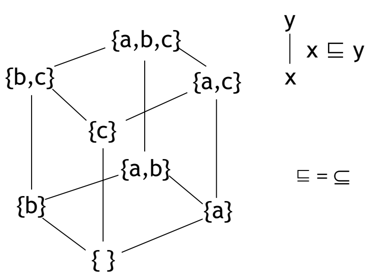
- A *partial-order* is
  - a set of elements S
  - an relation $x \sqsubseteq y$ that is
    - reflexive: $x \sqsubseteq x$
    - transitive: $(x \sqsubseteq y \land y \sqsubseteq z) \Rightarrow x \sqsubseteq z$
    - anti-symmetric: $(x \sqsubseteq y \land y \sqsubseteq x) \Rightarrow x = y$
  - two elements may be incomparable
- Examples $(S, \sqsubseteq)$

  $(Z, \leq)$ $(Z,=)$? $(Z, <)$?

  $(2^S, \subseteq)$ $(2^S, \supseteq)$

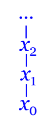
  $(S, \sqsupseteq)$

Hasse diagram: $2^{\{a,b,c\}}, \subseteq$



LUBs and Chains

- Given a subset $B \subseteq S$, y is an *upper bound* of B if $\forall x \in B\,.\, x \sqsubseteq y$
- y is a *least upper bound* ($\sqcup$B) if $y \sqsubseteq z$ for all upper bounds z
- A *chain* is a sequence of elements $x_0, x_1, x_2, \ldots$ such that $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots$
- For any finite chain $x_0,\ldots,x_n$, x_n is LUB
- What about infinite chains?



Complete partial orders

- A *complete partial order* (cpo) is a partial order in which *every* chain has a least upper bound
- Examples $(S, \sqsubseteq)$

  $(2^S, \subseteq)$

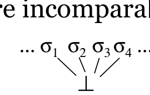
  $(\omega \cup \{\infty\}, \leq)$

  $([0,1], \leq)$

  $(S,=)$? $(S, \sqsupseteq)$?
- *cpo* may have least element $\bot$: *pointed*

Information content

- We consider one domain element to be less than another if it gives less information
- Non-termination gives less information than any store $(\bot \sqsubseteq x)$
- Stores σ are incomparable unless equal

$\Sigma_\bot$:


cpo?

- Recall: trying to find least fixed point in $\Sigma_\bot \to \Sigma_\bot$; how to order *functions*?

Pointwise ordering

- Functions are ordered on their results
- *Given* $f \in D \to E$, $g \in D \to E$, E cpo,

  $f \sqsubseteq_{D \to E} g \triangleq \forall x \in D\,.\, f(x) \sqsubseteq_E g(x)$
- Example $(Z \to Z_\perp)$

  $\lambda x \in Z\,.\, \mathit{if}\ x = 0\ \mathit{then}\ \perp\ \mathit{else}\ x$

  $\sqsubseteq$

  $\lambda x \in Z\,.\, x$
- $D \to E$ is pointed cpo if E is pointed:

  $\perp_{D \to E} = \lambda x \in D\,.\, \perp_E$

Back to while

- $\Gamma = \lambda f \in \Sigma_\perp \to \Sigma_\perp .\ \mathit{if}\ \neg \mathcal{B}[\![b]\!]\ \sigma\ \mathit{then}\ \sigma\ \mathit{else}\ f\,(\mathcal{C}[\![c]\!]\sigma)$
- Approximations to denotation of **while**:

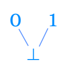
1. $\Gamma(\perp) = \lambda\sigma.\mathit{if}\ \neg\mathcal{B}[\![b]\!]\ \sigma\ \mathit{then}\ \sigma\ \mathit{else}\ \perp\ =$
2. $\Gamma(\Gamma(\perp)) = \lambda\sigma.\mathit{if}\ \neg\mathcal{B}[\![b]\!]\ \sigma\ \mathit{then}\ \sigma\ \mathit{else}$
   $\mathit{if}\ \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\sigma\ \mathit{then}\ \mathcal{C}[\![c]\!]\sigma\ \mathit{else}\ \perp$
3. $\Gamma(\Gamma(\Gamma(\perp))) =$
   $\lambda\sigma.\mathit{if}\ \neg\mathcal{B}[\![b]\!]\ \sigma\ \mathit{then}\ \sigma\ \mathit{else}$
   $\mathit{if}\ \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\sigma\ \mathit{then}\ \mathcal{C}[\![c]\!]\sigma\ \mathit{else}$
   $\mathit{if}\ \neg\mathcal{B}[\![b]\!]\mathcal{C}[\![c]\!]\mathcal{C}[\![c]\!]\sigma\ \mathit{then}\ \mathcal{C}[\![c]\!]\mathcal{C}[\![c]\!]\sigma\ \mathit{else}\ \perp$

- Denotation of while is $\bigsqcup_{n\in\omega}\Gamma^n(\perp)$
- Gives $\mathcal{C}[\![$**while true do skip**$]\!] = \perp_{\Sigma_\perp \to \Sigma_\perp} = \lambda\sigma \in \Sigma_\perp .\ \perp$

Fixed points

- Want to show $\bigsqcup_{n\in\omega}\Gamma^n(\perp)$ is solution to equation $x = \Gamma(x)$
- Not true for arbitrary Γ!
- Consider:

  $\Gamma(x) = $ if x = $\perp$ then 1
  else if x = 1 then $\perp$
  else if x = 0 then 0



  Need: monotonicity ($\Gamma^n(\perp)$ not a chain!)
- Consider a monotonic function:

  $\Gamma(x) = \mathit{if}\ x \le 0\ \mathit{then}\ \tan^{-1}(x)\ \mathit{else}\ 1$ $\quad R \cup \{-\infty, \infty\}$

  Need: continuity ($\Gamma(0) \ne 0$)