Lecture 6: Fixed Points and elements of Domain theory

1 Example

Consider recursive definition of gcd:

$$\mathrm{gcd}(a, b) = \begin{cases} b & \text{if } \mathrm{rem}(a, b) = 0 \\ \mathrm{gcd}(b, \mathrm{rem}(a, b)) & \text{otherwise} \end{cases}$$

where $\mathrm{rem}(a, b)$ is the unique r for which there is q such that $a = qb + r$ and $0 \leq r < b$.

Operational semantics: replace call to a function by its body.

The mathematical contents of the definition: a functional GCD with three arguments h, a and b, where h is a partial function from $\mathcal{N} \times \mathcal{N}$ to $\mathcal{N}$ and $a, b \in \mathcal{N}$. The functional GCD is defined as

$$\mathrm{GCD}(h, a, b) = \begin{cases} b & \text{if } \mathrm{rem}(a, b) = 0 \\ h(b, \mathrm{rem}(a, b)) & \text{otherwise} \end{cases}$$

The definition of GCD is **explicit** and non-recursive!!! it is defined in terms of rem, $\mathrm{if} - \mathrm{then} - \mathrm{else}$, application and abstraction.

The **relationship** between operational semantics and explicit definition: the function computed by the operational semantics is the least fixed point of the functional GCD with respect to the partial order $\subseteq$ between functions. The program computes the least h such that $\mathrm{GCD}(h) = h$.

2 Fixed points in the space of partial functions

We consider conditions which ensure the existence of the least fixed point for functional.

Notations and terminology A, B -sets; $A \to^p B$ the set of partial functions from A into B. Symbols f, g range over $A \to^p B$; F, G range over $(A \to^p B) \to (A \to^p B)$ i.e. over the **total function** from $(A \to^p B)$ into $(A \to^p B)$.

The (partial) functions are identified with their graphs and are partially ordered by the graph inclusion $\subseteq$.

Definition 1 *Let* $F : (A \to^p B) \to (A \to^p B)$ *be a functional*

1. F *is* **monotone** *if* $f \subseteq g$ *implies* $F(f) \subseteq F(g)$.
2. F *is* **continuous** *if* F *is monotone and for every* f, a*:* $F(f)(a)$ *depends only on the finite part of* f*, i.e., there is* $g \subseteq f$ *such that the domain of* g *is finite and* $F(g)(a) = F(f)(a)$.

3. f *is a fixed point of* F *if* $F(f) = f$.

4. f *is the least fixed point if it is a fixed point and it is included in any other fixed point of* F.

Theorem 1 (Fixedpoint theorem) *Each continuous functional* $F : (A \to^p B) \to (A \to^p B)$ *has a (unique) least fixed point.*

<u>Proof</u> Let f_0 be the everywhere undefined function. Let f_{n+1} be defined as $F(f_n)$.

$f_0 \subseteq f_1$; $f_1 = F(f_0) \subseteq f_2 = F(f_1)$, by monotonicity.

$$f_0 \subseteq f_1 \subseteq f_2 \ldots \subseteq f_n \subseteq \ldots$$

We claim that the function f defined as $\cup_{n=0}^{\infty} f_n$ is the least fixed point of F.

$f_n \subseteq F(f)$ by monotonicity. Hence

$$f \subseteq F(f)$$

$$f \supseteq F(f) \text{ by continuity of } F$$

Hence f is a fixed point.

If h is a fixed point of F then $f_n \subseteq h$ and hence $f \subseteq h$. Therefore f us the least fixed point of F.

end proof.

Note that the functional GCD is continuous.

3 ω-CPO

The only properties of partial orders used in the proof of the fixed point theorem were (1) $\emptyset$ is the least element and (2) an ω-chain $f_0 \subseteq f_1 \subseteq f_2 \ldots \subseteq f_n \subseteq \ldots$ has the least upper bound.

Definition 2 (ω Complete Partial Orders) *A structure* $\langle D, \sqsubseteq, \bot \rangle$ *is an* ω*-CPO if* $\sqsubseteq$ *is a partial order on* D*,* $\bot$ *is its least element and every* ω*-chain* $a_0 \sqsubseteq a_1 \ldots \sqsubseteq a_n \sqsubseteq \ldots$ *has the least upper bound.*

Notations: $\sqcup A$ - the least upper bound of a set A; alternative notations $lub\ A$, or $\lim_{n\to\infty} a_n$ or $\lim A$.

Definition 3 *Let* $\langle D, \sqsubseteq_D, \bot_D \rangle$ *and* $\langle E, \sqsubseteq_E, \bot_E \rangle$ *be* ω*-CPO and let* $F : D \to E$.

1. F *is monotone if* $a \sqsubseteq_D b$ *implies* $F(a) \sqsubseteq_E F(b)$.

2. F *is* ω*-continuous if it is monotone and for every* ω*-chain* $a_0 \sqsubseteq_D a_1 \ldots \sqsubseteq_D a_n \sqsubseteq_D \ldots$

$$F(\sqcup_D\{a_i : i \in \mathcal{N}\} = \sqcup_E\{F(a_i) : i \in \mathcal{N}\}$$

Theorem 2 (Examples of continuous functions) *1. Constant functions are* ω*-continuous.*

2. *The identity functions are* ω*-continuous.*

3. *The* ω*-continuous functions are closed under the composition, i.e., if* f *and* g *are* ω*-continuous then* $\lambda x.\ f(g(x)$ *is* ω*-continuous.*

Theorem 3 (Fixed point Theorem) *If* $\langle D, \sqsubseteq_D, \perp_D \rangle$ *is an* ω*-CPO and* $F : D \to D$ *is* ω*-continuous, then* F *has the least fixed point.*

4 Construction of ω-CPO

For a set A the partial order $A_\perp = \langle A \cup \{\perp_A\}, \sqsubseteq_A, \perp_A \rangle$, where $\perp_A$ is a new element and $a \sqsubseteq_A b$ if either $a =\perp_A$ or $a = b$, is an ω-CPO. $A_\perp$ is called a **flat** CPO.

1. What are continuous extensions of succ?

2. What are continuous extensions of iszero?

Cartesian Product - order element $\langle a, b \rangle \sqsubseteq \langle a', b' \rangle$ if $a \sqsubseteq_A a'$ and $b \sqsubseteq_B b'$.

The projection functions are ω-continuous.

Lemma 4 $F : A \times B \to C$ *is* ω*-continuous if it is continuous in each of its arguments, i,e.,* $\lambda x.F(a, x)$ *is* ω*-continuous (for each* $a \in A$*) and* $\lambda y.F(y, b)$ *is* ω*-continuous (for each* $b \in B$*).*

Examples of continuous functions.

1. **strict** extension of $f : A \to B$ to $A_\perp \to B_\perp$. If one of the arguments is $\perp$ then the result is $\perp$.

2. What are continuous extensions of plus?

3. What are continuous extension of equality = :int$\times$ int$\to$ $\mathtt{bool}$?

4. What are continuous extensions of the disjunction?

5. What should be an interpretation of **if-then-else**:bool$\times$int$\times$int$\to$int?

Functional Space

Definition 4 *Let* $\langle D, \sqsubseteq_D, \perp_D\rangle$ *and* $\langle E, \sqsubseteq_E, \perp_E\rangle$ *be* ω*-CPO. Define the pointwise partial order on* $D \to E$ *as:* $f \sqsubseteq g$ *if* $f(d) \sqsubseteq_E g(d)$ *for every* $d \in D$*. Define the partial order* $D \to_c E$ *as*

The universe: *the set of* ω*-continuous functions from* D *to* E*.*

Partial order: *pointwise.*

Lemma 5 *Let* $\langle D, \sqsubseteq_D, \perp_D\rangle$ *and* $\langle E, \sqsubseteq_E, \perp_E\rangle$ *be* ω*-CPO.* $D \to_c E$ *is an* ω*-CPO with the minimal element* $\lambda x.\ \perp_E$.

Lemma 6 (Application) *The application function* $\mathsf{App} : ((D \to_c E) \times D) \to E$ *is* ω*-continuous.*

Lemma 7 (Abstraction) *Let* $F : (D \times E) \to_c A$*. There is a unique function* G *from* D *to* $E \to_c A$ *such that* $G d e = F(d, e)$*. Moreover,* G *is* ω*-continuous.*

Lemma 8 *The least fixed point operator is continuous, i,e., is in* $(D \to_c D) \to_c D$

5 Continuity vs Monotonicity

Consider the following functional $F : (\mathsf{Nat}_\perp \to \mathsf{Nat}_\perp) \to \mathsf{Bool}_\perp$.

$$F(g) = \begin{cases} \mathsf{true} & \text{if } g(\perp) = \perp \text{ and for all } n\ (g(n) \neg \equiv \perp \\ \perp & \text{otherwise} \end{cases}$$

F is monotone.

Consider chain $g_1 \sqsubseteq g_2 \sqsubseteq \ldots g_n \ldots$, where

$$g_k(n) = \begin{cases} 1 & \text{for } 1 \leq n \leq k \\ \perp & \text{otherwise} \end{cases}$$

Let g be lub of g_k. Clearly $g(\perp) = \perp$ and $g(n) = 1$ for all $n \not\equiv \perp$.

$F(g_i) = \perp$ and $F(g) = \mathsf{true}$. Hence lub of $F(g_i) = \perp \neq$true=F(lub g_i). Hence F is not continuous.

6 Two Examples of Fixed Points

6.1 Factorial Function

letrec $x(t)$ be (if $zero(t)$ then 1 else $t \times x(t-1)$ in $x(3)$.

This is shorthand for letrec x be (λt. if $(zero\ t)1(mult\ t\ (x(pred\ t))$ in $x3$.

With the primitives $zero$: int→bool, 1:int, mult : int → int→int, pred:int→int, if: bool→ int → int → int. This primitives are appropriately interpreted.

The meaning will be the value of $x3$ in the environment which assigns to x the least fixed point of $F = \lambda x \lambda t$. if $(zero\ t)1(mult\ t\ (x(pred\ t))$.

Compute the approximations $F_n(\perp) = F(\dots F(\perp))$ to the least fixed point of F.

The least fixed point of F is the factorial function.

6.2 Minimization

letrec $x(t)$ be (if $answer(t)$ then true else $x(t+1)$ in x(0).

Here $answer : int \to bool$.

Unlike the factorial function, here $x(t)$ is reduced to $x(t+1)$ - bigger argument.

Compute δ - the lfp of $F = \lambda x \lambda t$ if $answer(t)$ then t else $x(t+1)$; $F : (int \to int) \to (int \to int)$.

Compute the approximations to δ.

Check that δ is the function which for arbitrary argument n returns the least $m \geq n$ such that $answer(n) = true$ assuming that for k $n \leq k < m$ the $answer(k)$ is false.

7 Continuous Frame

Continuous Frame

1. For primitive α - D^α is chosen as a CPO.
2. $D^{\alpha\to\beta}$ is defined as the set of all continuous functions from D^α to D^β.
3. $D^{\alpha\to\beta}$ is ordered pointwise, i.e,

$$f \leq_{\alpha\to\beta} g \text{ if } \forall d \in D^\alpha.\ fd \leq_\beta gd$$

Inductive definition of semantics $[\![t^\alpha]\!]\rho \in D^\alpha$, here t is a term, ρ is an environment.

Definitions by Structural induction on terms:

Atoms
1. $[\![c]\!]\rho$ - meaning of constants is provided by interpretation and is independent of environment (should belong to an appropriate domain).
2. $[\![x]\!]\rho$ is defined as $\rho(x)$.

Application $[\![M^{\alpha\to\beta}N^\alpha]\!]$?

By induction $[\![M^{\alpha\to\beta}]\!]\rho$ is already defined and is a function from D^α into D^β. Also $[\![N^\alpha]\!]\rho$ is already defined and is an element of D^α. Hence $[\![MN]\!]\rho$ is defined as $[\![M]\!]\rho[\![N]\!]\rho$. (Note that in MN we have syntactical application vs semantical application in $[\![M]\!]\rho[\![N]\!]\rho$.)

Abstraction $[\![\lambda x^\alpha.\ M^\beta]\!]$ is the function f such that for every $d \in D^\alpha$

$$fd = [\![M]\!] \quad \underbrace{\rho[d/x]}_{\text{updated env}}$$

Note that we have to ensure that this is an element of $D^{\alpha\to\beta}$. Here, where it is important that $D^{\alpha\to\beta}$ was chosen as the set of all continuous function from D^α to D^β and that the following lemma holds:

Lemma 9 $[\![t]\!]\rho$ *is a continuous function and depends continuously on* ρ.

In the above lemma the set of environments is ordered pointwise: $\rho_1 \sqsubseteq \rho_2$ if $\rho_1(x) \sqsubseteq_\sigma \rho_2(x)$ for every variable x of type σ

8 Programming Language for Computable Functions

8.1 Language of PCF

Primitive types: num and bool.

Constants and their types:

1. 0: num
2. true, false: bool
3. succ,pred: num→num
4. zero?: num→bool

5. For every type σ if-then- else$_\sigma$: bool$\times\sigma\times\sigma\to\sigma$

6. For every type σ the fixed point operator $Y_\sigma\ :\ (\sigma\to\sigma)\to\sigma$

PCF= typed lambda calculus with the above constants.

Example: minus is defined by expressions $Y\lambda m : num \to num \to num\lambda x : num\lambda y : num \mathit{if}\ zero?(y)\ then\ x\ else\ m(pred(x))(pred(y))$

Numerals: $0, succ(0),\ succ(succ(\ldots succ(0)\ldots))$. We use $\bar{1}$ for $succ(0)$, $\bar{2}$ for $succ(succ(0))$ etc.

Reduction δ rules:

1. $pred(0)\to 0$.

2. $pred(succ(\bar{n}))\to\bar{n}$

3. $zero?(0)\to true$

4. $zero?(succ(\bar{n}))\to false$

8.2 Call by name evaluation Semantics

$$\frac{M\to N}{pred(M)\to pred(N)}$$
$$\frac{M\to N}{zero?(M)\to zero?(N)}$$
$$\frac{M\to N}{succ(M)\to succ(N)}$$
$$\frac{M\to N}{ML\to NL}$$
$$\lambda x:\sigma M)N\to M[N/x]$$
$$\frac{M\to N}{\text{if } M \text{ then } L_1 \text{ else } L_2\ \to \text{if } N \text{ then } L_1 \text{ else } L_2}$$
$$\text{if true then } L_1 \text{ else } L_2\ \to L_1$$
$$\text{if false then } L_1 \text{ else } L_2\ \to L_2$$
$$YM\to M(YM)$$

Lemma 10 *If* $M\to N$ *and* $M\to N'$ *then* $N\equiv N'$.

Many step reduction: $\to^*$. Normal form.

What are closed normal forms? Numerals and $(\lambda x.M)$

$$V ::= 0\mid true\mid false\mid succ(V)\mid(\lambda x.M)$$

8.3 ISWIM

ISWIM = typed λ-calculus + let + letrec and constants.

Syntax:

1. let x be M in N.
2. letrec x be M in N

Binding:

1. for let x be M in N: x binds free occurrences in N; free occurrences in M remains free.
2. for letrec x be M in N: x binds free occurrences in both N and M.

Semantics:

1. By translation to PCF.
   (a) let x be M in N is sugaring for $(\lambda x\ N)M$;
   (b) letrec x be M in N is sugaring for let x be $Y(\lambda x\ M)$ in N which is sugaring for $(\lambda x\ N)(Y(\lambda x\, M)$.
2. Direct operational semantics
   (a) let x be M in $N \to N[M/x]$
   (b) letrec x be M in $N \to$ letrec x be M in $N[M/x]$.

Further sugaring mutual declaration and product type.

let (x be M_1 and y be M_2) in N (y differs from x). Here x binds free N and y binds free y in N; free x and y in M_1 and M_2 remain free.

letrec (x be M_1 and y be M_2) in N (y differs from x). Here x binds free x in M_1, M_2 and N and y binds free y in M_1, M_2 and N.

Reductions

1. let (x be M_1 and y be M_2) in $N \to N[M_1/x, M_2/y]$
2. letrec (x be M_1 and y be M_2) in $N \to$ letrec (x be M_1 and y be M_2) in $N[M_1/x, M_2/y]$.

9 Denotational Semantics for PCF

Interpretation of types bool - flat bool, int - flat int.

Interpretation of functions

Strict interpretation of succ, pred, zero.

If-then-else non strict interpretation.

Y - the least fixed point operators.