Least Fixed Points

A relation $\sqsubseteq$ is a *partial order* iff it is reflexive, transitive and anti-symmetric. Paired with a set D it forms a *poset* $(D, \sqsubseteq)$.

The least element of a poset, $\bot$, satisfies $\forall x \in D.\ \bot \sqsubseteq x$ if it exists.

The *least upper bound* of a chain $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$ is written $\bigsqcup_{n\geq 0} d_n$ if it exists, which satisfies $\forall n \in \mathbb{N}.d_n \sqsubseteq \bigsqcup_{n\geq 0} d_n$ and $\forall d \in D.(\forall m \geq 0.d_m \sqsubseteq d) \Rightarrow \bigsqcup_{n\geq 0} d_n \sqsubseteq d$.

A *chain complete poset* or *cpo* is a poset $(D, \sqsubseteq)$ in which all countable increasing chains have lubs. A *domain* is a cpo that further possesses a least element.

Domain Of Partial Functions

$D = \{f | f \text{ is apartial function}, dom(f) \subseteq X, im(f) \subseteq Y\}$

$f \sqsubseteq g \iff dom(f) \subseteq dom(g) \wedge (\forall x \in dom(f).f(x) = g(x))$

The lub is $f(x) = \begin{cases} f_n(x) & \text{if } x \in dom(f_n), \text{ some } n \\ undefined & \text{otherwise} \end{cases}$, and $\bot(x) = undefined$

Poset Mappings

$f : D \to E$ is *monotone* if $\forall d, d' \in D.d \sqsubseteq d' \Rightarrow f(d) \sqsubseteq f(d')$.

$f : D \to E$ is *continuous* if it is monotone and $f(\bigsqcup_{n\geq 0} d_n) = \bigsqcup_{n\geq 0} f(d_n)$.

$f : D \to E$ is *strict* if $f(\bot) = \bot$.

To check that monotone f is continuous it is sufficient to show that for every chain $d_n$in D, $f(\bigsqcup_{n\geq 0} d_n) \sqsubseteq \bigsqcup_{n\geq 0} f(d_n)$ holds in E.

Fixed Points

If D is a poset and $f : D \to D$, $d \in D$ is a *pre-fixed point* of f if $f(d) \sqsubseteq d$. The least such point is $fix(f)$ and satisfies $f(fix(f)) \sqsubseteq fix(f)$ and $\forall d \in D.f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$.

Tarski Fixed Point Theorem: for continuous f, $fix(f) = \bigsqcup_{n\geq 0} f^n(\bot)$ and furthermore $f(fix(f)) = fix(f)$ and hence is the least fixed point of f as well.

To see this is true, first observe that $f^n(\bot)$ forms a chain since $f^0(\bot) = \bot \sqsubseteq f^1(\bot)$ and by monotonicity $\big(f^n(\bot) \sqsubseteq f^{n+1}(\bot)\big) \Rightarrow \big(f^{n+1}(\bot) = f(f^n(\bot)) \sqsubseteq f(f^{n+1}(\bot)) = f^{n+2}(\bot)\big)$, with the rest following by induction. Now:

$$\begin{aligned} f(fix(f)) &= f(\bigsqcup_{n\geq 0} f^n(\bot)) \\ &= \bigsqcup_{n\geq 0} f(f^n(\bot)) \\ &= \bigsqcup_{n\geq 0} f^{n+1}(\bot) \\ &= \bigsqcup_{n\geq 0} f^n(\bot) \\ &= fix(f) \end{aligned}$$

Where the penultimate step depends on the fact that discarding finite elements at the start of a chain doesn't change its lub. It is easy to show that $\forall n \in \mathbb{N}.f^n(\bot) \sqsubseteq d$ for any $d \in D$ such that $f(d) \sqsubseteq d$ by induction on n.

Domain Constructions

Product

$D = D_1 \times D_2$

$(d_1, d_2) \sqsubseteq (d_1', d_2') \iff d_1 \sqsubseteq_1 d_1' \wedge d_2 \sqsubseteq d_2'$

$\bigsqcup_{n\geq 0}(d_{1,n}, d_{2,n}) = (\bigsqcup_{i\geq 0} d_{1,i}, \bigsqcup_{j\geq 0} d_{2,j})$

Continuous Functions Of Two Arguments

$f : D \times E \to F$ for cpos D, E and F is monotone iff it is monotone in each argument separately and continuous iff it preserves lubs of chains in each argument separately.

Diagonalization

If the family of elements $d_{m,n} \in D$ satisfies $m \leq m' \wedge n \leq n' \Rightarrow d_{m,n} \sqsubseteq d_{m',n'}$ then $\bigsqcup_{n\geq 0} d_{0,n} \sqsubseteq \bigsqcup_{n\geq 0} d_{1,n} \sqsubseteq \ldots$ and $\bigsqcup_{m\geq 0} \bigsqcup_{n\geq 0} d_{m,n} = \bigsqcup_{k\geq 0} d_{k,k} = \bigsqcup_{n\geq 0} \bigsqcup_{m\geq 0} d_{m,n}$

Functions

For cpos $(D, \sqsubseteq_D)$ and $(E, \sqsubseteq_E)$, the function cpo $(D \to E, \sqsubseteq)$ has:

$D \to E = \{f | f : D \to E, f \text{ is a continuous function}\}$

$f \sqsubseteq f' \iff \forall d \in D.f(d) \sqsubseteq_E f'(d)$

$(\bigsqcup_{n\geq 0} f_n)(d) = \bigsqcup_{n\geq 0} f_n(d)$

If E is a domain then $\bot_{D\to E}(d) = \bot_E$

The functions $ev(f, d) = f(d)$ and $cur(f)(d') = \lambda d \in D.f(d', d)$ are continuous.

Fixed Point

For a domain D, the function $fix : (D \to D) \to D$ is continuous.

Flat Domains

For any set X, $x \sqsubseteq x' \iff x = x'$ makes $(X, \sqsubseteq)$ into a cpo, called the *discrete* cpo with underlying set X. For $X_\perp = X \cup \{\perp\}$ with $x \sqsubseteq x' \iff x = x' \vee x = \perp$, $(X_\perp, \sqsubseteq)$is a domain called the *flat* domain.

Scott Induction

A subset $S \subseteq D$ is called *chain-closed* iff for all chains $d_n$in D $(\forall n \geq 0. d_n \in S) \Rightarrow (\bigsqcup_{n \geq 0} d_n) \in S$.

If D is a domain, $S$is called *admissible* iff it is a chain-closed subset of D and $\perp \in S$.

A property $\Phi(d)$ is called chain-closed/admissible if its corresponding set, $\{d \in D | \Phi(d)\}$ is.

Scott's Fixed Point Induction Principle: for continuous $f : D \to D$ on domain D, to show that $fix(f) \in S$ for admissible $S \subseteq D$ it suffices to prove that $\forall d \in D(d \in S \Rightarrow f(d) \in S)$.

Basic relations: for cpo D, $\{(x, y) \in D \times D | x \sqsubseteq y\}$ and $\{(x, y) \in D \times D | x = y\}$ are chain closed subsets.

Inverse image: for continuous $f : D \to E$ between cpos D and E, for a chain-closed subset S of E, $f^{-1}S = \{x \in D | f(x) \in S\}$ is a chain-closed subset.

Logical operations: for cpo D with chain-closed subsets S and T, then $S \cup T$ and $S \cap T$ are chain-closed subsets since any chain will have to visit one of the sets infinitely often, so we can use the lub from that set.

Infinite intersections: for cpo D with a family of chain-closed subsets $S_i$indexed by a set I, $\bigcap_{i \in I} S_i$ is a chain-closed subset of D. This doesn't work for unions (consider unioning together an infinite number of singleton sets to build arbitrary chain-closed sets..).

PCF

$\tau ::= nat | bool | \tau \to \tau$

$M ::= 0 | succ(M) | pred(M) | zero(M) | true | false | if\, M\, then\, M\, else\, M | x | fn\, x : \tau.M | M\, M | fix(M)$ with $x \in \mathbb{V}$ and expressions identified up to α-conversion.

Typing relations are of the form $\Gamma \vdash M : \tau$ where Γ is a finite partial function mapping variables to types and obey the standard rules, which have the unique-type property and that if $\Gamma \vdash M : \tau$ and $\Gamma[x \to \tau] \vdash M' : \tau'$ then $\Gamma \vdash M'[M/x] : \tau'$.

The evaluation relation takes the form $M \Downarrow_\tau V$ where V is a value $(V ::= 0 | succ(V) | true | false | fn\, x : \tau.M)$. The evaluation rules are also standard, and have the determinism property. The only interesting one is that:

$$\frac{M\ fix(M) \Downarrow_\tau V}{fix(M) \Downarrow_\tau V}$$

Aims Of Semantics

Types τ map to domains $[\![\tau]\!]$, terms $M : \tau$ map to elements $[\![M]\!] \in [\![\tau]\!]$ and $[\![M]\!] = [\![M']\!] \Rightarrow [\![\mathcal{C}[M]]\!] = [\![\mathcal{C}[M']]\!]$.

Soundness: for any type τ, $M \Downarrow \tau V \Rightarrow [\![M]\!] = [\![V]\!]$.

Adequacy: for $\tau = bool$ or nat, $[\![M]\!] = [\![V]\!] \in [\![\tau]\!] \Rightarrow M \Downarrow_\tau V$.

Contextual equivalence: if any occurrences of the first phrase in a complete program can be replaced by the second phrase without affecting the observable results of executing the program. $\Gamma \vdash M_1 \cong_{ctx} M_2 : \tau$ holds iff the typings hold and for all contexts $\mathcal{C}$ where $\mathcal{C}[M_1]$ and $\mathcal{C}[M_2]$ are closed terms of type γ, $\mathcal{C}[M_1] \Downarrow_\gamma V \iff \mathcal{C}[M_2] \Downarrow_\gamma V$.

For all types $\tau$and closed terms M_1, M_2 of that type, if $[\![M_1]\!] = [\![M_2]\!] \in [\![\tau]\!]$ then $M_1 \cong_{ctx} M_2 : \tau$ by soundness, compositionality and adequacy.

Types

$[\![nat]\!] = \mathbb{N}_\perp$, $[\![bool]\!] = \mathbb{B}_\perp$ (flat domain)

$[\![\tau \to \tau']\!] = [\![\tau]\!] \to [\![\tau']\!]$ (function domain)

Terms

For well typed M, define a function $[\![\Gamma \vdash M]\!] : [\![\Gamma]\!] \to [\![\tau]\!]$ where $[\![\Gamma]\!] = \prod_{x \in dom(\Gamma)} [\![\Gamma(x)]\!] = \{\rho : dom(\Gamma) \to \bigcup_{x \in dom(\Gamma)} [\![\Gamma(x)]\!] | \forall x \in dom(\Gamma). \rho(x) \in [\![\Gamma(x)]\!]\}$whose elements are mapping variables to an element in the domain of the corresponding type.

Semantic rules are defined by structural induction and are both continuous and strict:

$[\![\Gamma \vdash 0]\!](\rho) = 0 \in [\![nat]\!]$

$[\![\Gamma \vdash true]\!](\rho) = true \in [\![bool]\!]$

$[\![\Gamma \vdash false]\!](\rho) = false \in [\![bool]\!]$

$[\![\Gamma \vdash x]\!](\rho) = \rho(x) \in [\![\Gamma(x)]\!]$ where $x \in dom(\Gamma)$

$$[\![\Gamma \vdash succ(M)]\!](\rho) = \begin{cases} [\![\Gamma \vdash M]\!](\rho) + 1 & \text{if } [\![\Gamma \vdash M]\!](\rho) \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

$$[\![\Gamma \vdash pred(M)]\!](\rho) = \begin{cases} [\![\Gamma \vdash M]\!](\rho) - 1 & \text{if } [\![\Gamma \vdash M]\!](\rho) > 0 \\ \perp & \text{otherwise} \end{cases}$$

$$[\![\Gamma \vdash zero(M)]\!](\rho) = \begin{cases} true & \text{if } [\![\Gamma \vdash M]\!](\rho) = 0 \\ false & \text{if } [\![\Gamma \vdash M]\!](\rho) > 0 \\ \bot & \text{otherwise} \end{cases}$$

$$[\![\Gamma \vdash if\, M_1\, then\, M_2\, else\, M_3]\!](\rho) = \begin{cases} [\![\Gamma \vdash M_2]\!](\rho) & \text{if } [\![\Gamma \vdash M_1]\!](\rho) = true \\ [\![\Gamma \vdash M_3]\!](\rho) & \text{if } [\![\Gamma \vdash M_1]\!](\rho) = false \\ \bot & \text{otherwise} \end{cases}$$

$[\![\Gamma \vdash M_1\, M_2]\!](\rho) = ([\![\Gamma \vdash M_1]\!](\rho))([\![\Gamma \vdash M_2]\!](\rho))$

$[\![\Gamma \vdash fn\, x : \tau . M]\!](\rho) = \lambda d \in [\![\tau]\!].[\![\Gamma[x \to \tau] \vdash M]\!](\rho[x \to d])$ (where $x \notin dom(\Gamma)$)

$[\![\Gamma \vdash fix(M)]\!](\rho) = fix([\![\Gamma \vdash M]\!](\rho))$

For closed terms $M \in PCF_\tau$, $\emptyset \vdash$M:τ so we get the only Γ-environment being $\bot$ and since functions $\{\bot\} \to D$ are in bijection with D we can identify the denotation of closed PCF terms with elements of the domain denoting their type: $[\![M]\!] = [\![\emptyset \vdash M]\!](\bot) \in [\![\tau]\!]$.

Continuity Aside

Constant functions are continuous.

Projections in dependent product domains are continuous.

If $f : D \to E$ and $g : E \to F$ are continuous functions between cpos, then their composition $g \circ f : D \to F$ is also continuous.

Function pairing $\langle f, g\rangle (x) = (f(x), g(x))$ is continuous.

Substitution And Soundness

Substitution: if $\Gamma \vdash M : \tau$ and $\Gamma[x \to \tau] \vdash M' : \tau'$ then $[\![\Gamma \vdash M'[M/x]]\!](\rho) = [\![\Gamma[x \to \tau] \vdash M']\!](\rho[x \to [\![\Gamma \vdash M]\!](\rho)])$.

Soundness: for all types $\tau$and closed terms $M, V \in PCF_\tau$, if $M \Downarrow_\tau V$ is derivable then $[\![M]\!]$ and $[\![V]\!]$ are equal elements of the domain $[\![\tau]\!]$.

Adequacy

The family of binary relations $\lhd_\tau \subseteq [\![\tau]\!] \times PCF_\tau$ indexed by types τ relates elements of the domain $[\![\tau]\!]$ to closed terms of type τ.

A Γ-substitution σ is a function mapping each variable $x \in dom(\Gamma)$ to a closed PCF term $\sigma(x)$ of type $\Gamma(x)$.

Say that $\rho \lhd_\Gamma \sigma \iff \forall x \in dom(\Gamma). \rho(x) \lhd_{\Gamma(x)} \sigma(x)$ (i.e. lifting the relation from types and terms to type environments and Γ-substitutions).

$d \lhd_{nat} M \iff (d \in \mathbb{N} \Rightarrow M \Downarrow_{nat} succ^d(0))$

$d \lhd_{bool} M \iff (d = true \Rightarrow M \Downarrow_{bool} true) \wedge (d = false \Rightarrow M \Downarrow_{bool} false)$

$d \lhd_{\tau \to \tau'} M \iff \forall e, N.(e \lhd_\tau N \Rightarrow d(e) \lhd_{\tau'} M\, N)$

Fundamental Property: If $\Gamma \vdash M : \tau$ is a valid typing, then $\rho \lhd_\Gamma \sigma \Rightarrow [\![\Gamma \vdash M]\!](\rho) \lhd_\tau M[\sigma]$ where $M[\sigma]$ is the term resulting from the simultaneous substitution of $\sigma(x)$ for x in M for each $x \in dom(\Gamma)$. This specializes to $[\![M]\!] \lhd_\tau M$.

From this we can prove that $[\![M]\!] = [\![V]\!] \Rightarrow M \Downarrow_\tau V$ as required. For example, if $\tau = nat$, then for some $n \in \mathbb{N}$ $V = succ^n(0)$ and so:

$$\begin{aligned} [\![M]\!] &= [\![succ^n(0)]\!] \\ &\Rightarrow (n = [\![M]\!]) \lhd_\tau M \\ &\Rightarrow M \Downarrow succ^n(0) \end{aligned}$$

Extensionality

$\Gamma \vdash M_1 \leq_{ctx} M_2 : \tau$ holds iff the typings hold and for all contexts $\mathcal{C}$ for which $\mathcal{C}[M_1]$ and $\mathcal{C}[M_2]$ are closed terms of type $\gamma \in \{nat, bool\}$ and for all values $V : \gamma$, it is true that $\mathcal{C}[M_1] \Downarrow_\gamma V \Rightarrow \mathcal{C}[M_2] \Downarrow_\gamma V$.

It can be shown that for all types τ and terms M_1, M_2 it is true that $M_1 \leq_{ctx} M_2 : \tau \iff [\![M_1]\!] \lhd_\tau M_2$ and for $\tau \in \{nat, bool\}$ it is also true that $M_1 \leq_{ctx} M_2 : \tau \iff \forall V : \tau(M_1 \Downarrow_\tau V \Rightarrow M_2 \Downarrow_\tau V)$ i.e. their evaluations are syntactically equal, and for $\tau = \tau_1 \to \tau_2$, $M_1 \leq_{ctx} M_2 : \tau \iff \forall M : \tau_1(M_1\, M \leq_{ctx} M_2\, M : \tau_2)$ i.e. their evaluation at all points obey the equality.

Full Abstraction

There are contextually equivalent PCF terms with unequal denotations. A denotational semantics is fully abstract if this is not the case.

por	$true$	$false$	$\bot$
$true$	$true$	$true$	$true$
$false$	$true$	$false$	$\bot$
$\bot$	$true$	$\bot$	$\bot$

There is no closed PCF term $P : bool \to (bool \to bool)$ satisfying $[\![P]\!] = por$. Now consider:

$$\begin{aligned} T_i \;=\; & fn\, f : bool \to (bool \to bool). \\ & if\, (f\, true\, \Omega)\, then \\ & if\, (f\, true\, \Omega)\, then \\ & if\, (f\, false\, false)\, then\, \Omega\, else\, B_i \\ & else\, \Omega \\ & else\, \Omega \end{aligned}$$

Where $B_1 = true$ and $B_2 = false$. Now $T_1 \cong_{ctx} T_2 : (bool \to (bool \to bool)) \to bool$ but $[\![T_1]\!] \neq [\![T_2]\!]$. Contextual equivalence can be proved by using the extensionality property on T_i

and then just considering all M of the argument type. This requires that M evaluates to $true$, $true$ and $false$ (in order that we get a value) on the arguments $(true, \Omega)$, $(\Omega, true)$ and $(false, false)$ respectively. This means that $[\![M]\!]$ coincides with por, which is impossible and hence the terms are trivially contextually equivalent because they can never return a value in any PCF context.

Extending PCF with por in the obvious way and our semantics with the additional clause:

$$[\![\Gamma \vdash por(M_1, M_2)]\!](\rho) = por([\![\Gamma \vdash M_1]\!](\rho))([\![\Gamma \vdash M_2]\!](\rho))$$

Yields a language which is fully abstract with respect to our semantics i.e. $\Gamma \vdash M_1 \cong_{ctx} M_2 : \tau \iff [\![\Gamma \vdash M_1]\!] = [\![\Gamma \vdash M_2]\!]$