# Fixed Points

The denotational semantics of Postfix is based on well-known semantic objects like numbers, boolean, and sequences.

To give the denotational semantics of a programming language, even a simple functional one like PCF, we need more complex mathematical objects.

The key difficulty in PCF is the existance of the fix point operator – what does it mean?

The problem reduces to that of defining recursive functions. This is also needed as the starting point for any other programming language, such as ML, Algol, or C++.

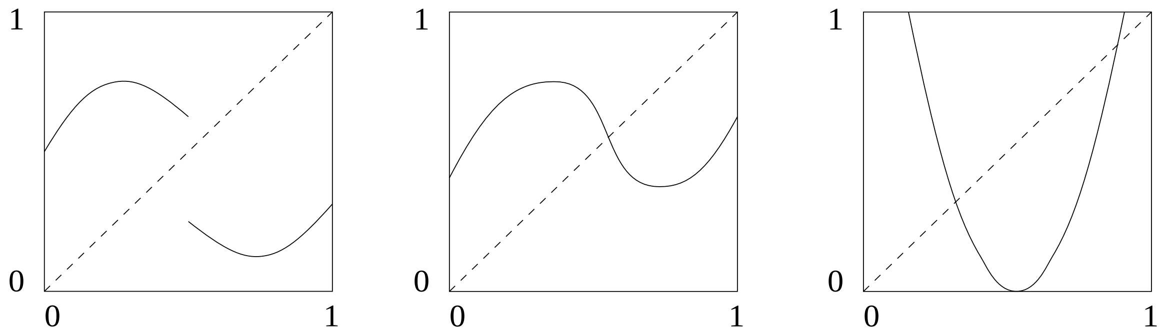
# Fixed Points in General

Recall the factorial example from PCF, which desugared to:

$$(\lambda f : nat \to nat.\ f\ 1)\ (fix_{nat\to nat}\ \lambda g : nat \to nat\ .\lambda y : nat\ .$$
$$\texttt{if}\ \ Eq?\ y\ 0\ \texttt{then}\ 1$$
$$\texttt{else}\ \ y * g(y-1))$$

We saw a denotational semantics for the conditional statement, but what is the meaning of $fix$? Intuitively, we want it to find the fixed point of the given function.

Not all functions have fixed points, and some have multiple fixed points:



# Iterative Techniques for Finding Fixed Points

- The fixed point of "rewriting" is the normal form (assuming it exists).
- The square root of a non-negavive number $n$ is given by:

$$x = \frac{x + n/x}{2}$$

  Iterate on values of $x$ starting with $n$.

- Definition of the set of even numbers:

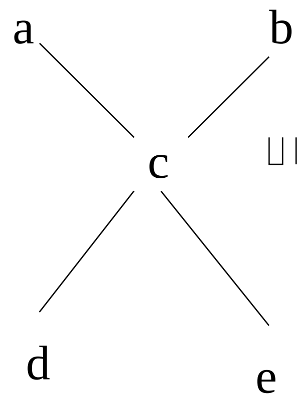
$$evens = \{0\} \cup \{x + 2 | x \in evens\}$$

In each case, we start with an estimate of the fixed point and iterate to improve the estimate. Each step is better than the previous.
In the limit, each of these iterative processes converge, but this isn't necessarily the case for all functions, all domains, or all starting points.

# Partial Orders

We need some semantic machinery for defining domains on which fixed points of functions will exist.
A *partial order* is a pair $\langle D, \sqsubseteq \rangle$, where $\sqsubseteq$ is a binary relation on $D$ that is reflexive, transitive, and anti-symmetric.

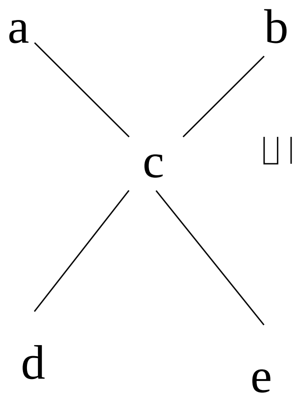
$a \subseteq b$ means that $a$ is weaker than $b$ or $a$ is an approximation for $b$.




Since these are *partial orders* and not *total orders*, there may be elements that are incomparable, i.e., neither $a \sqsubset b$ nor $b \sqsubseteq a$.

# Upper Bounds

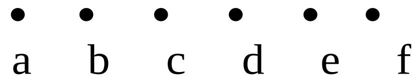
An *upper bound* of $X \sqsubseteq D$ is an element $u \in D$ such that for all $x \in X$, $x \sqsubseteq u$.




In this example, $c$ is an upper bound of $\{d, e\}$ as are $a$ and $b$. $b$ is the only upper bound of $\{c, b\}$, while the set $\{a, b\}$ has no upper bounds.

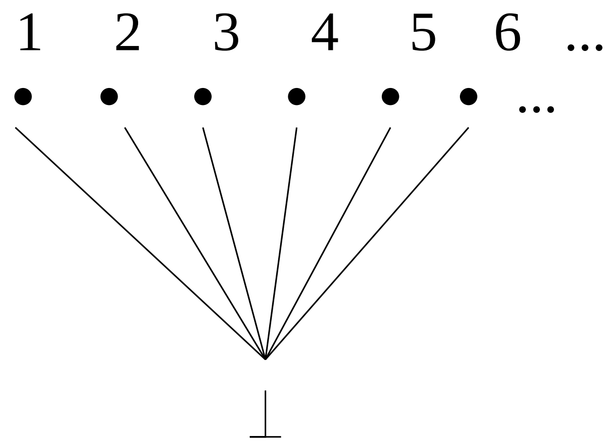
The *least upper bound* of a set $X$ is the smallest upper bound, assuming such a unique element exists. It is denoted $\sqcup_D X$.
For example, $\sqcup_D \{d, e\}$ is $c$.

A *discrete* partial order is one in which every pair of elements is incomparable.



# Lifted domains

The smallest or weakest element in a domain is usually called *bottom*, denoted $\perp$.
Any domain can be *lifted* to a domain with a bottom element by adding that element.




A *flat* partial order is a lifted discrete partial order.

You should think of this relation as the "definedness" relation, so all integers are equally defined. It is not the usual less-than-or-equal ordering on integers.

# Partial Orders on Semantic Domains

Our semantics domains for the denotation semantics of PCF were constructed out of basic elements like integers, which will be flat partial orders, and out of functions, products, and sums.

Given domains $D$ and $E$ with partial orders $\sqsubseteq_D$ and $\sqsubseteq_E$, we define the following:

**Product Domains:**

$\sqsubseteq_{D\times E}$ is a partial order on $D \times E$ defined as:

$$\langle d_1, e_1\rangle \sqsubseteq_{D\times E} \langle d_2, e_2\rangle \text{ if } d_1 \sqsubseteq_D d_2 \text{ and } e_1 \sqsubseteq_E e_2$$

**Sum Domains:**

$\sqsubseteq_{D+E}$ is a partial order on $D + E$ defined as:

$$(inject_{D,E} d_1) \sqsubseteq_{D+E} (inject_{D,E} d_2) \text{ if } d_1 \sqsubseteq_D d_2$$

$$(inject_{D,E} e_1) \sqsubseteq_{D+E} (inject_{D,E} e_2) \text{ if } e_1 \sqsubseteq_E e_2$$

**Function Domains:**
$\sqsubseteq_{D+E}$ is a partial order on $D \rightarrow E$ defined as:

$$f_1 \sqsubseteq_{D \rightarrow E} f_2 \text{ if, for all } d \text{ in } D, f(d) \sqsubseteq_E f(e)$$

Need to confirm that each of these are reflexive, transitive and anti-symmetric.

There are also constructions for sequence domains.

# CPOs

A partial order $D$ is *complete* if every chain in $D$ has a least upper bound in $D$.

- Any partial order with a finite number of elements is a CPO.
- Any flat partal order is a CPO.
- The interval [0, 2] ordered by $leq$ on reals is a CPO. Note that an infinite chain (say $[0, \sqrt{(2)})$) is does not necessarily contain its own lub.
- The partial functions with signatures $Nat \rightarrow Nat$ ordered by "definedness" form a CPOs.

# Pointed CPOs

A CPO is *pointed* if it has a bottom element.
We can view bottom as a bad approximation to any element in the domain. I.e., $\perp_{Nat}$ is an approximation for any natural number, while the empty function is an approximation to a partial function (i.e., undefined everywhere).

Pointed CPOs often used to encode partial functions at total functions.

If $f : D \to E_{\perp}$ is partial, then $f' : D \to E$ can be defined the total function that is equal to $f$ wherever it is defined and $\perp$ otherwise.

# Monotonicty

Supposed the $f : D \rightarrow E$, where $D$ and $E$ are (not necessarily pointed) CPOs. Then

- $f$ is *monotonic* if $d_1 \sqsubseteq d_2$ implies $f(d_1) \sqsubseteq_E f(d_2)$
- $f$ is *continuous* if, for all chains $C$ in $D$, $f(\sqcup_D C) = \sqcup_E \{f(c) | c \in C\}$

A monotonic function preserves order between CPOs.
A continuous function preserves limits of CPOs.

# Continuous and Monotonic Functions

On finite CPOs and infinite ones with only finite changes, monotonicity implies continuity.
On any CPO, continuity implies monotonicity.

# Least Fixed Point Theorem

Theorem:
If $D$ is a pointed CPO, then a continuous function $f : D \rightarrow D$ has a least fixed point given by $\sqcup_D\{(f^n \perp_D) | n \geq 0\}$.

# Pointed CPOs

Domain D with $\sqsubseteq$ is a lattice iff for all $a, b$ in $D$ $a \sqcup b$ and $a \sqcap b$ exist.

meet

For D with $\sqsubseteq$ , X subset of D, $\bigsqcup X$ (least upper bound) denotes the element,

if it exists, such that

for all x in X, $x \sqsubseteq \bigsqcup X$

for all d in D, if for all x in X, $x \sqsubseteq d$, then $\bigsqcup X \sqsubseteq d$.

greatest lower bounds are analogous

D is a complete lattice if l.u.b.s and g.l.b.s exist for all subsets

# Pointed CPOs

A subset X of D is a chain iff X is non-empty and for all $a, b$ in X

$$a \sqsubseteq b \quad \text{or} \quad b \sqsubseteq a$$

D is a complete partial order (c.p.o.) iff every chain in D has and l.u.b. in D

D is a pointed c.p.o. if it also has a least element.

# Pointed CPOs

In semantic domains, read $\sqsubseteq$ as "less defined than" for functions

A function f:A->B is monotonic if for all x, y in A,

$$x \sqsubseteq_A y \quad \Rightarrow f(x) \sqsubseteq_B f(y)$$

E.g., program-halts(p) = true if p != $\perp$
false if p = $\perp$
is not monotonic

Given c.p.o.'s A and B, a monotonic function f: A-> B is continuous iff

$$f(\sqcup X) = \sqcup \{ f(x) \mid x \text{ in } X \} \quad \text{for all chains } X$$

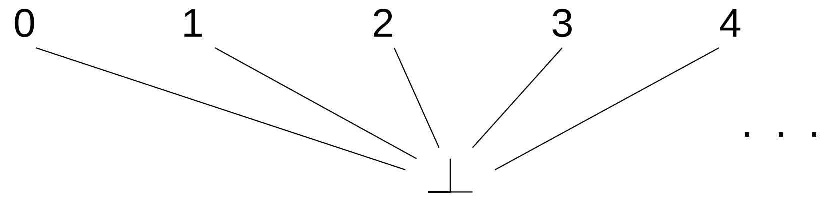
# Pointed CPOs

Given a continuous function f: D-> D, d in D is a fixed point of f iff f(d) = d. It is the least fixed point if there is none small in the $\sqsubseteq$ ordering.

Theorem: If D is a pointed cpo, then the least fixed point of a continuous function f: D -> D is defined to be:

$$\textbf{fix}\ \ f = \bigsqcup \{ (f \perp) \mid i >= 0\}$$

# Pointed CPOs




If A and B are pointed cpo's, then so are

A x B

A + B

A->

and all operation builders (projection, function expressions, etc.) are continuous