H250: Honors Colloquium – Introduction to Computation

Fixpoints and Applications

Marius Minea
marius@cs.umass.edu

Fixpoint Definition

Given $f : X \to X$, a *fixpoint* (fixed point) of f is a value c with $f(c) = c$.

A function may have 0, 1, or many fixpoints.
Examples:

fixpoint for a real-valued function: intersection with $y = x$
fixpoint for a permutation of $1 \, .. \, n$

Think of the function as a *transformation* (which may be repeated)
Fixpoint: *no change*

Intuitive Examples

All-pairs shortest paths in graph

Simpler: Path relation in graph

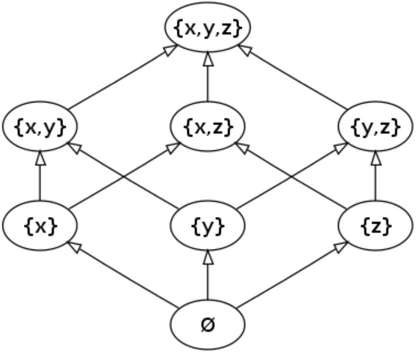
Questions:
When do such computations terminate?

How to express this with fixpoints?

Partially Ordered Sets

Recall: partial order on a set:

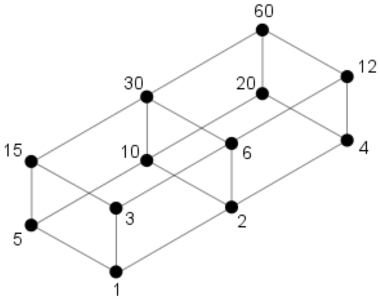
reflexive
antisymmetric
transitive



A set A together with a partial order $\sqsubseteq$ on A is called a *partially ordered set* (*poset*) $\langle A, \sqsubseteq \rangle$

Lattices

Many familiar partial orders have additional properties:



- any two elements have a unique *least upper bound* (*join* $\sqcup$)

least x with $a \sqsubseteq x$ and $b \sqsubseteq x$

- any two elements have a unique *greatest lower bound* (*meet* $\sqcap$)

A poset with these properties is called a *lattice*.

Inductively: any *finite* set has a least upper / greatest lower bound.

Complete lattice: *any* subset has least upper/greatest lower bound.

$\Longrightarrow$ lattice has a top $\top$ and bottom $\bot$ element.

Iterating a function

Define $f^n(x) = \underbrace{f \circ f \circ \ldots \circ f}_{n \text{ times}}(x)$.

On a finite set, iteration will

- close a cycle, or
- reach a fixpoint (particular case)

For an infinite set, iteration may be infinite (none of the above)

Monotonic Functions

Given a poset $\langle S, \sqsubseteq \rangle$, a function $f : S \to S$ is monotonic if it is either

- increasing, $\forall x \forall y : x \sqsubseteq y \to f(x) \sqsubseteq f(y)$
- decreasing, $\forall x \forall y : x \sqsubseteq y \to f(x) \sqsupseteq f(y)$

Knaster-Tarski Fixpoint Theorem

A *monotonic* function on a *complete* lattice has a *least* fixpoint and a *greatest* fixpoint.
More generally:
The set of fixpoints of a *monotonic* function on a *complete* lattice is also a complete lattice.

Proof (for lfp/gfp): Assume for simplicity f is increasing.
Consider the sequence $\perp, f(\perp), f^2(\perp), \ldots$.

Then $\perp \sqsubseteq x$ for any x, by definition, so $\perp \sqsubseteq f(\perp)$.

By induction, $f^n(\perp) \sqsubseteq f^{n+1}(\perp)$.

Take the set $P = \{x : x \sqsubseteq f(x)\}$. Clearly $\perp \in P$ as seen above.
Also, P contains all fixpoints.

Knaster-Tarski Proof (cont'd.)

P has a least upper bound $u = \sqcup P$.
Then for all $x \in P$, we have $x \sqsubseteq u$, so $x \sqsubseteq f(x) \sqsubseteq f(u)$.
So $f(u)$ is also an upper bound, but $u \sqsubseteq f(u)$ (being the least), so then $u \in P$.

Then $f(u) \sqsubseteq f(f(u))$ (by monotonicity), so $f(u) \in P$ and then $f(u) \sqsubseteq u$ (as upper bound), so $f(u) = u$.
So u is a fixpoint, and since all fixpoints are in P, it is the greatest fixpoint.

Symmetric argument for greatest lower bound.

If the lattice has finite height (in particular, finite), can find fixpoints by repeated iteration: $f^n(\bot)$ (least) and $f^n(\top)$ (greatest).

Example: Transitive Closure

Transitive closure of a relation R: least relation that includes R and is transitive:
$R^+ = R \cup R^2 \cup \ldots \cup R^n \cup \ldots$.

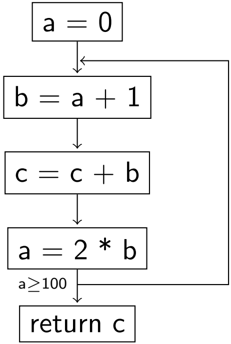
How to write as fixpoint? Think: path = edge or path + edge

Express this recursion as function on relations: $f(X) = R \cup X \circ R$.
Then $f(R) = R \cup R^2$, $f^2(R) = R \cup R^2 \cup R^3$, etc.

$R^+ = \mathit{lfp}\ X\ .\ R \cup X \circ R$ (least fixpoint wrt X)

Application: Program Analysis

```
int a = 0, b, c = 0;
do {
  b = a + 1;
  c = c + b;
  a = 2 * b;
} while (a < 100);
return c;
```



Reaching Definitions: What are all assignments that may reach the current point without being overwritten by other assignments?

Live Variables: For every program point, which variables will have their values used on at least one path from that point?

Solved by a fixpoint iteration on control flow graph.

Application: Temporal Logic Model Checking

Temporal Logic: statements about what may/must happen on an some/all execution paths of a system.

$\mathbf{EX}p$: p holds in *some* next state
$\mathbf{AX}p$: p holds in *all* next states
$\mathbf{AF}p$: p holds sometime in the future on *all* paths
$\mathbf{EG}p$: p holds forever on *some* path etc.

All of these have fixpoint characterizations $\Longrightarrow$ can use to compute state sets

$\mathbf{EG}p = p \land \mathbf{EXEG}p$
As fixpoint: $\mathbf{EG}p = \text{gfp } f \ . \ p \land \mathbf{EX}f$

Programming Languages: Defining Recursion

Lambda Calculus (Alonzo Church, 1930s)

$e ::= x$ — variable

$\mid \lambda x.e$ — function abstraction (definition), think: `fun x -> e`

$\mid e_1\ e_2$ — function application

Think: simplest possible functional language

Recursion in Lambda-Calculus

Usually, recursion requires *naming* the recursive object.
But λ-calculus does not let us introduce names...

Start from the diverging self-application $(\lambda x \,.\, x\ x)(\lambda x \,.\, x\ x)$

Define a closed term that applies a function to an argument
$\mathbf{Y} = \lambda f \,.\, (\lambda x \,.\, f(x\ x))(\lambda x \,.\, f(x\ x))$

$\mathbf{Y}$ is called *fixpoint combinator*, because $\mathbf{Y}\ f = f(\mathbf{Y}\ f)$
Can use $\mathbf{Y}$ to define recursive functions!

Take $fact(n) = \text{if } n = 0 \text{ then } 1 \text{ else } n \cdot fact(n-1)$
Rewrite as $\mathsf{F} = \lambda f \,.\, \lambda n \,.\, \text{if } n = 0 \text{ then } 1 \text{ else } n \cdot f(n-1)$

Then define $fact = \mathbf{Y}\ \mathsf{F}$. Expanding $\mathsf{F} = \mathbf{Y}\ \mathsf{F}$ stops at $n = 0$ which does not re-evaluate argument f (here, $\mathbf{Y}\ \mathsf{F}$).