LiU-FP2016: Lecture 3

The Untyped λ-Calculus: Recursion and Fixed Points

Henrik Nilsson

University of Nottingham, UK

Fixed Points (1)

Consider a recursive function like factorial:

```
fac(n) = if n == 0 then
             1
         else
             n * fac(n - 1)
```

Fixed Points (1)

Consider a recursive function like factorial:

```
fac(n) = if n == 0 then
             1
         else
             n * fac(n - 1)
```

Attempt to translate into λ-calculus:

$$FAC \equiv \lambda n.IF\ (ISZ\ n)\ 1\ (TIMES\ n\ (FAC\ (PRED\ 1)))$$

Is this OK?

Fixed Points (2)

But consider

$$FAC' \equiv \lambda f.\lambda n.IF\ (ISZ\ n)\ 1\ (TIMES\ n\ (f\ (PRED\ 1)))$$

Fixed Points (2)

But consider

$$FAC' \equiv \lambda f.\lambda n.IF\ (ISZ\ n)\ 1\ (TIMES\ n\ (f\ (PRED\ 1)))$$

Now suppose FAC ***is*** the factorial function.

Fixed Points (2)

But consider

$$FAC' \equiv \lambda f.\lambda n.IF\ (ISZ\ n)\ 1\ (TIMES\ n\ (f\ (PRED\ 1)))$$

Now suppose FAC ***is*** the factorial function.

Then $FAC'\ FAC$ is ***also*** the factorial function.

That is: $FAC = FAC'\ FAC$ (where $=$ here is semantical equality).

Fixed Points (3)

In general, whenever $x = f(x)$ for some function f and value x, x is a ***fixed point*** of f.

Fixed Points (3)

In general, whenever $x = f(x)$ for some function f and value x, x is a ***fixed point*** of f.

Thus, FAC is a fixed point of FAC'.

But

$$FAC \equiv FAC' \; FAC$$

is still useless as a definition.

Fixed Points (3)

In general, whenever $x = f(x)$ for some function f and value x, x is a ***fixed point*** of f.

Thus, FAC is a fixed point of FAC'.

But

$$FAC \equiv FAC'\ FAC$$

is still useless as a definition.

Fixed Points (4)

However, suppose we have a function FIX that when given an arbitrary unary function computes its smallest fixed point; i.e., for any function f:

$$FIX\ f = f\ (FIX\ f)$$

Then

$$FAC \equiv FIX\ FAC'$$

is a valid definition, assuming FIX can be defined.

Fixed Points (4)

However, suppose we have a function FIX that when given an arbitrary unary function computes its smallest fixed point; i.e., for any function f:

$$FIX\ f = f\ (FIX\ f)$$

Then

$$FAC \equiv FIX\ FAC'$$

is a valid definition, assuming FIX can be defined.

Fixed Points (4)

However, suppose we have a function FIX that when given an arbitrary unary function computes its smallest fixed point; i.e., for any function f:

$$FIX\ f = f\ (FIX\ f)$$

Then

$$FAC \equiv FIX\ FAC'$$

is a valid definition, assuming FIX can be defined.

Questions

1. Does a function like FIX exist?
2. Does every function even ***have*** a fixed point?
3. If FIX exists, can it be defined in the λ-calculus?

Answer Q2 (1)

Does every function have a fixed point?

Answer Q2 (1)

Does every function have a fixed point?

If we work with with functions on ordinary sets, clearly not! E.g.

$$x = not\ x$$

does not have a solution in the set $\{False, True\}$.

Answer Q2 (1)

Does every function have a fixed point?

If we work with with functions on ordinary sets, clearly not! E.g.

$$x = \mathit{not}\ x$$

does not have a solution in the set $\{\mathit{False}, \mathit{True}\}$.

Similarly, there is no $n \in \mathbb{N}$ such that

$$n = \mathit{succ}\ n$$

Answer Q2 (2)

But there is a solution of we turn to ***domain theory*** and consider functions over ***pointed domains*** that have a specific bottom element $\perp$ denoting divergence, non-termination:

$$
\begin{aligned}
\perp &= not\ \perp \\
\perp &= succ\ \perp
\end{aligned}
$$

Answer Q2 (2)

But there is a solution of we turn to ***domain theory*** and consider functions over ***pointed domains*** that have a specific bottom element $\perp$ denoting divergence, non-termination:

$$\begin{aligned} \perp &= \mathit{not}\ \perp \\ \perp &= \mathit{succ}\ \perp \end{aligned}$$

In general, domain theory allows for an additional possible result, $\perp$, which is the lest element, meaning all functions have a unique least fixed point in that setting.

Answer Q1 & Q3

Yes and yes: a function for computing fixed points in general exists and it can be defined in the lambda calculus.

Answer Q1 & Q3

Yes and yes: a function for computing fixed points in general exists and it can be defined in the lambda calculus.

Many possibilities. The call-by-name fixed-point combinator Y is probably the most famous and simplest:

$$Y \equiv \lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$$

Answer Q1 & Q3

Yes and yes: a function for computing fixed points in general exists and it can be defined in the lambda calculus.

Many possibilities. The call-by-name fixed-point combinator Y is probably the most famous and simplest:

$$Y \equiv \lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$$

Let's verify $Y\ F = F\ (Y\ F)$ for any F (on the white board).

Back to Factorial

Now we can define:

$$FAC \equiv Y \; FAC'$$

Fixed Point Combinators in Real Life

- In denotational semantics, the meaning of recursion and iteration is given in terms of fixed point constructions.
- In langauges like Haskell, fix can easily be defined (see example).
- Variations of fixed point operators are quite often used in practice; e.g. for monadic fixed points (see later).