Recursion / Fixpoint combinators

"combinator" = a lambda calc term with
no free variables
$\lambda x. x$ (identity combinator)

U combinator

$U = \lambda h.\ h\ h$

Python

def fact(x)
  if $x = 0$ then return 1
  else $x * \text{fact}(x-1)$

we can't do this

λ calc

$\text{fact} = \lambda x.\ \text{if}\ (\text{iszero}\ x)\ \text{then}\ 1\ \text{else}\ x * \text{fact}(x-1)$

$\text{fact} = \lambda h.\lambda x.\ \text{if}\ (\text{isZero}\ x)\ \text{then}\ 1\ \text{else}\ x * h(x-1)$

$U = \lambda f.\ f\ f$

$$fact_U = U\left(\lambda h.\lambda x.\ \text{if}\ (\text{isZero}\ x)\ \text{then}\ 1\ \text{else}\ x * (h\ h\ (x-1))\right)$$

Fixpoint (Fixed point)

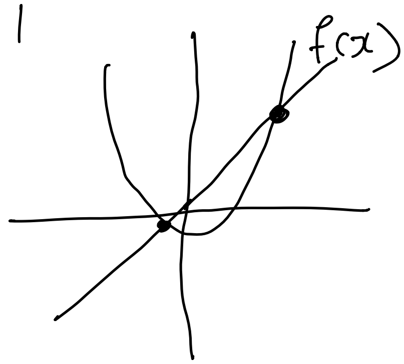
$f : \mathbb{R} \to \mathbb{R}$

if $x \in \mathbb{R}$ s.t. $x = f(x)$

then x is a fixpoint of f

E.g. $f(x) = x^2 - 1$

$x = x^2 - 1$



$fact = \lambda x. \ldots$

$fact = F(fact)$

$$F(f) = \boxed{\lambda x.\ \text{if}\ x \leq 0\ \text{then}\ 1\ \text{else}\ x * f(x-1)}$$

goal: find a solution of f s.t.

$$F(f) = f$$

Deriving the Y combinator

$x = f(x)$

$$F = \lambda x. \ldots\ldots \overset{\text{hole}}{F} \ldots\ldots$$

- pulling out the "F"

$$F = \underbrace{(\lambda f.\, \lambda x. \ldots\, f \ldots)}_{h}\, F$$

$$(*)\quad F = h\,F$$

$$Y\,h \longrightarrow^{*} f$$

s.t. $f = h\,f$

$\rightarrow$ let $F = F'F'$

$$F'F' = h\,(F'F')$$

By $(*)$

$$= (\lambda x.\, h\,(x\,x))\, F'$$

$$x \equiv (\lambda y. y)\, x$$

$$\therefore \quad F' = \lambda x.\, h\,(x\ x)$$

$$F = F'\,F' = (\lambda x.\, h\,(x\ x))\,(\lambda x.\, h\,(x\ x))$$

$$= \underbrace{[\lambda h'.\,(\lambda x.\, h'(x\ x))\,(\lambda x.\, h'(x\ x))]}_{\text{Y combinator}}\ h$$

$$F = (\ldots\ldots)\ h$$

$$F = Y\,h$$

so Y is a fixpoint combinator

$$\underline{F} = \underline{h}\ \underline{F}$$

---

applying to factorial

$$\underbrace{Y\,(\lambda f.\,\lambda x.\,\text{if}\,(\text{isZero}\ x)\ \text{then}\ 1\ \text{else}\ x * f\,(x-1))}\ 10$$

$$F = \lambda x . \ldots F \ldots F \ldots F$$

Theorem: Every λ calculus equation of the above form has a solution