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 * fact(x-1)
```

λ calc

we can't do this

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

$fact = \lambda h. \lambda x.$ if (isZero x) then 1 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)

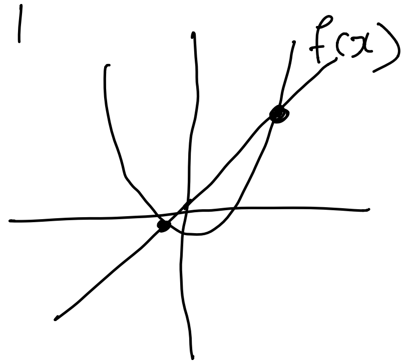
$f : \mathbb{R} \to \mathbb{R}$

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

then x is a fixpoint of f

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

$x = x^2 - 1$



$fact = \lambda x \ldots$

$fact = F(fact)$

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

<u>goal</u>: find a solution of f s.t.

$$F(f) = f$$

Deriving the Y combinator

hole

$$F = \lambda x. \dots\dots F \dots\dots$$

$$x = f(x)$$

- pulling out the "F"

$$F = \underbrace{\left(\lambda f.\ \lambda x. \dots f \dots\right)}_{h} F$$

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

$$Y h \longrightarrow_{*} f$$

<u>s.t</u> $f = h f$

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

$$\underline{F'F'} = h\,\underline{(F'\,F')}$$

By $(*)$

$$= \underbrace{(\lambda x.\ h\ (x\ x))}\ F'$$

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

$$F'$$

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

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

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

$$F = (\,.\;.\;.\;.\;.\;.\,)\, h$$

$$F = Y\,h$$

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

so Y is a fixpoint combinator

---

applying to factorial

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

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

<u>Theorem</u>: Every λ calculus equation of the above form has a solution