

## 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
```

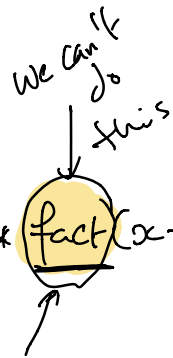
```
    else x * fact(x-1)
```

$\lambda$  calc

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

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

we can't  
do this



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

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

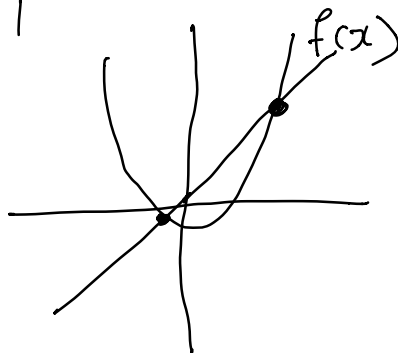
---

Fixpoint (Fixed point)

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

if  $x \in \mathbb{R}$  s.t.  $x = f(x)$   
 then  $x$  is a fixpoint of  $f$

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



---


$$fact = \lambda x. \dots$$

$$fact = F(fact)$$

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

goal: find a solution of  $f$  s.t.  
 $F(f) = f$

Deriving the  $Y$  combinator

$$\textcircled{F} = \lambda x. \dots \overset{\downarrow \text{hole}}{\textcircled{F}} \dots$$

- pulling out the "F"

$$F = \underbrace{(\lambda f. \lambda x. \dots f \dots)}_h F$$

$$(*) \bullet F = \underbrace{h F}$$

$$\gamma h \rightarrow_* f$$
  
s.t.  $f = hf$

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

$$\begin{aligned} \underline{F' F'} &= \underline{h (F' F')} \\ &= (\lambda \textcircled{x}. h (x x)) \textcircled{F'} \\ &\quad \underbrace{\hspace{10em}}_{\textcircled{F'}} \end{aligned}$$

By (\*)

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

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

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

$$= \left[ \lambda h'. (\lambda x. h'(x x)) (\lambda x. h'(x x)) \right] h$$

$$F = (\dots) h$$

↑ combinator

$$F = Y h$$

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

so  $Y$  is a fixpoint combinator

applying to factorial

$$Y (\lambda f. \lambda x. \text{if (is zero } x) \text{ then } 1 \text{ else } x * \hat{f}(x-1))$$

10

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

Theorem: Every  $\lambda$  calculus equation of the above form has a solution