COL 100 - Lec 9

- Linear recursion
- Nested recursion
- Mutual recursion

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ f(n-1) + g(n-1) & \text{otherwise} \end{cases}$$

$$g(n) = \begin{cases} 0 & \text{if } n = 0 \\ g(n-1) + f(n) & \text{otherwise} \end{cases}$$

$g(g(2))$

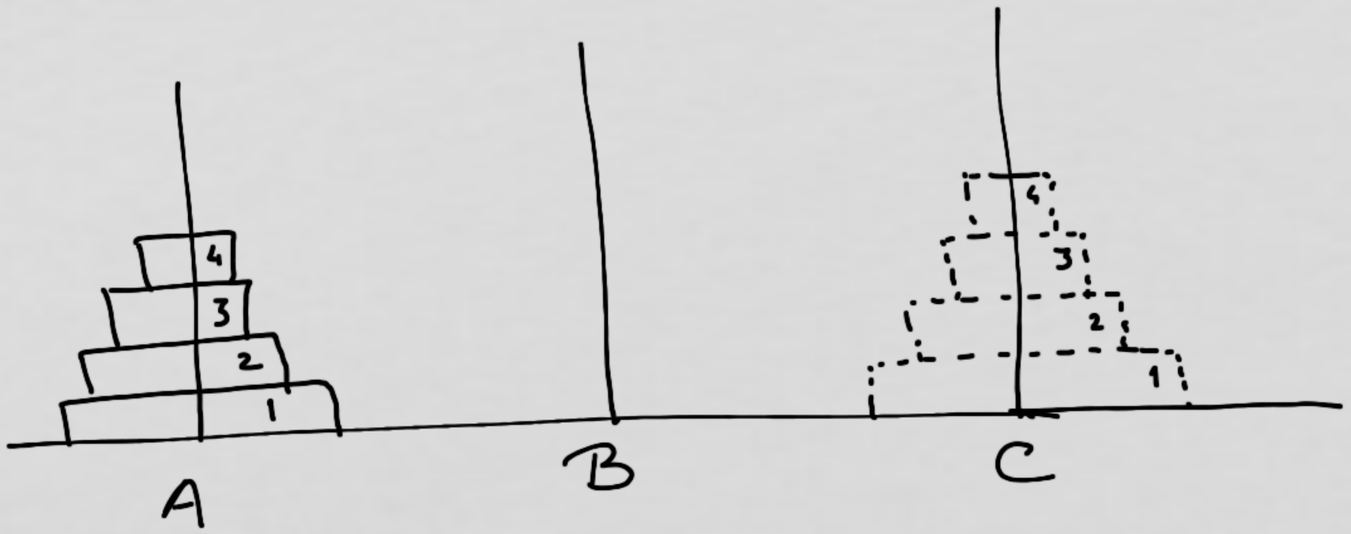
$= g(g(1) + f(2))$

$g(0) + f(1)$ (0, 0)

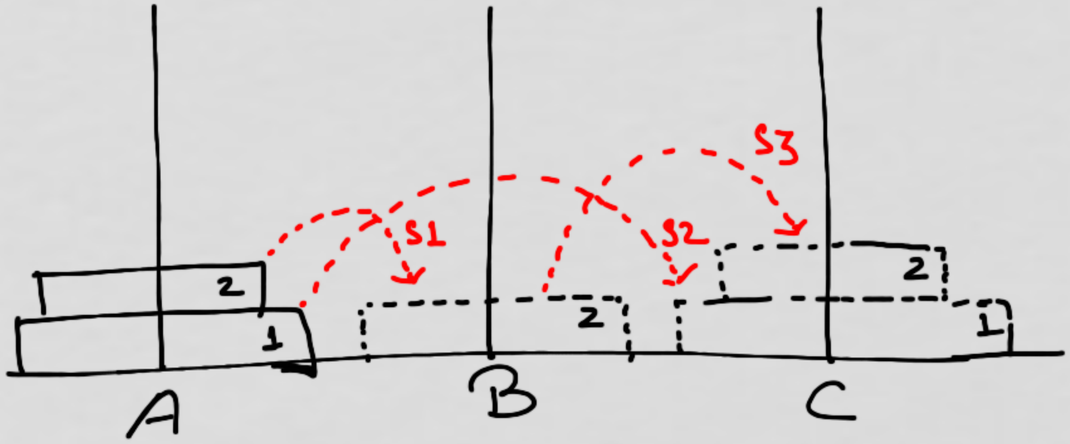
$f(0) + g(0)$

$f(1) + g(1)$

Tower of Hanoi [Tower of Brahma]



Constraint : At no point a smaller disk is beneath a larger disk !



Example with 2 disks

If I have n disks, what observation can I make?

- Move top (n-1) disks from A to B
- Move the n^{th} disk [bottom-most] from A to C
- Move (n-1) disks from C to A

MoveDisks (n, src, aux, dst)

$$= \begin{cases} \text{Move disk from src to dst} & \text{if } n = 1 \\ \text{MoveDisks}(n-1, src, dst, aux) & \text{otherwise} \\ \text{Move disk from src to dst} & \\ \text{MoveDisks}(n-1, aux, src, dst) & \end{cases}$$

Proof of Correctness

Base case: $n = 1$, true

I.H. : $\exists\, n \geq 1$: MoveDisks (n, A, B, C) is true

I.S. : MoveDisks (n+1, A, B, C) = by I.H.
- MoveDisks (n, A, C, B)
- Move n^{th} disk from A to C
- MoveDisks (n, B, A, C)

Recurrence Relation

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 1 + 2T(n-1) & \text{otherwise} \end{cases}$$

$T(n) = 2 * T(n-1) + 1$

$= 2^2 . T(n-2) + 2 + 1$

$= 2^3 . T(n-3) + 2^2 + 2 + 1$

$\vdots$

$= 2^{n-1} T(1) + 2^{n-2} + \cdots + 2^2 + 2 + 1$

$= 2^n - 1$

Technical Completeness

```
fun pow(x, n) = ....
```

by default it is $\mathbb{N} \longrightarrow \mathbb{N}$

- If I want to specify pow for all reals

then

```
fun pow (x:real, n:int) =
        if n=0 then 1.0
        else pow (x, n-1) * x
```

Is it techically complete?

- Does your function guarantee, for instance,
  - $x \neq 0$ ✗
  - x is real ✓
  - n is int ✓
  - n is nonnegative ✗

```
fun pow(x: real, n) =
    if n<0 then 1.0 / pow(x, ~n)
    else if n=0 then 1.0
         else x * pow(x, n-1)
```

Is it technically complete? $0.0^0 = 1.0$ whereas $0.0^n = 0 \; \forall n>0$

- if $x = 0.0$
  and $n = -m < 0$
  then $0.0^n = 1.0 / (0.0^m)$

$$= \frac{1.0}{0.0}$$ Divisim by zero

Pow int. version

$$x^n = \begin{cases} \text{Undefined} & \text{if } n < 0 \\ \text{Undefined} & \text{if } x = 0 \wedge n = 0 \\ 1 & \text{if } x \neq 0 \ \& \ n = 0 \\ x \cdot x^{n-1} & \text{otherwise} \end{cases}$$

Sml

```
Exception negExponent ;
Exception ZeroPowerZero ;
fun int pow (x, n) =
    if n < 0
    then raise negExponent
    Else if n = 0 then if x = 0 then
                       raise ZeroPowerZero
                       Else 1
    Else x * intpow (x, n-1)
```