Induction / Recursion

Natural numbers

Axiomatic definition (Peano):

- 1 is a natural number.
- Every natural number n has a successor, $S(n)$, also noted $n+1$
- $s(n)$ is a natural number.
  $n = m$ iff $S(n) = S(m)$
  $\forall n \in \mathbb{N}, S(n) \neq 1$

If K is a set such that:
$1 \in K$
if $k \in K$, then $k+1 \in K$ $\forall k \in \mathbb{N}$
then $K = \mathbb{N}$

Induction: We wish to prove claim(1), ... claim(n) is true $\forall n \in \mathbb{N}$.

If we show:

Step 1: Prove that claim(1) is true.

Step 2: Prove that $\forall k \in \mathbb{N}$, claim(k) is true $\rightarrow$ claim($k+1$) is true.

The principle of mathematical induction allows us to conclude that claim(n) is true, $\forall n \in \mathbb{N}$.

Let K the the set of numbers n such that

claim (n) is true.

- From step 1, claim (1) is true, therefore $1 \in K$.
- From step 2, if k is in K, claim (k) is true, then claim $(k+1)$ is true, $k+1$ is in K.

Therefore ~~K~~ $= \mathbb{N}$.

Examples:

Claim (n): $1^3 + 2^3 + \dots n^3 = (1+2+\dots n)^2$

$LHS(n) = 1^3 + \quad n^3$

$RHS(n) = (1+\dots n)^2 = \frac{n^2(n+1)^2}{2}$

claim (n): $\underbrace{2\times 2 + 3\times 2^2 + 4\times 2^3 + \dots (n+1)2^n}_{LHS(n)} = \underbrace{n2^{n+1}}_{RHS(n)}$

Step 1: $\left.\begin{array}{l} LHS(1) = 2\times 2 = 4 \\ RHS(1) = 2^2 = 4 \end{array}\right)$ claim (1) is true.

Step 2: Assume claim (k) is true.

$LHS(k) = RHS(k)$

$LHS(k+1) = LHS(k) + (k+2)2^{k+1} = k2^{k+1} + k2^{k+1} + 2^{k+2}$

$= 2k2^{k+1} + 2^{k+2}$

$= (k+1)2^{k+1} = RHS(k+1)$

Therefore claim $(k+1)$ is true.

The method of proof by induction allows us to conclude that Claim(n) is true $\forall n \in \mathbb{N}$.

Careful:

Claim(n): $n^2 - n + 41$ is prime.

Claim(1): 41 is prime.
Claim(2): 43 is prime
Claim(3): 47 is prime
Claim(4): 53 is prime
Claim(5): 61 is prime.

$\vdots$

Seems to be true ... but:

Claim(41): $41^2 - 41 + 41 = 41^2$: not prime!

Recursive definitions

Sometimes it is difficult to define an object explicitly. It might be easier to define it with respect to itself → this is called a recursion.

Two steps:

Step 1: specify the value at 0/1

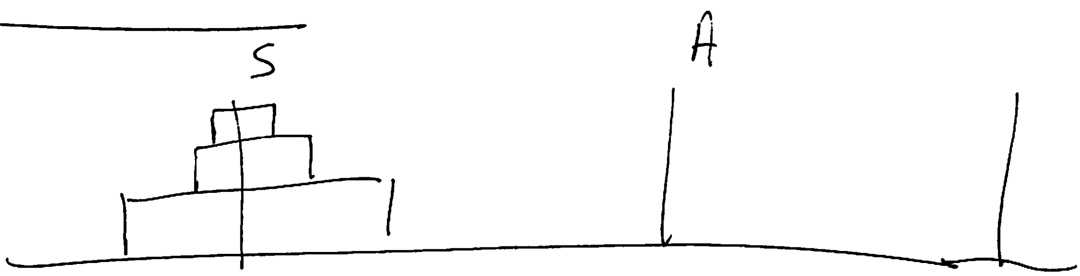
Step 2: give a rule for finding its value at a ~~integer~~ natural number n, from its value at smaller natural numbers.

Example: Fibonacci:

$$f_0 = 0 \qquad f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2}$$

Hanoi towers



Problem: move N disks from the source (S) to the destination (D), using an auxiliary (A), such that you always have disks piled in decreasing order of size.

Algorithm:

```
Function HanoiTower (N, S, D, A)
    if N == 1
        Move (1, S, D)
    else
        HanoiTower (N-1, S, A, D)
        Move (1, S, D)
        HanoiTower (N-1, A, D, S)
```

Is this algorithm correct?

claim (n) : Hanoi Tower (n, P_1, P_2, P_3) is correct

Proof by induction:

Step 1: claim (1): Hanoi Tower (1, $P_1 = S$, $P_2 = D$, $P_3 = A$) is correct.

Step 2: Let $k \in \mathbb{N}$. Assume claim (k) is true, namely Hanoi Tower (k, S, D, A) is correct.

HanoiTower ($k+1$, S, D, A):

Hanoi Tower (k, S, A, D) $\leftarrow$ correct

move (1, S, D) $\leftarrow$ correct

Hanoi Tower (k, A, D, S) $\leftarrow$ correct.

Complexity? Let $T(n)$ be the number of operations.

Step 1: $T(1) = 1$

Step 2: at ~~step~~ n :

$T(n-1)$ move

1

$T(n-1)$

$$T(n) = 2\,T(n-1) + 1$$

Can we find a closed form?

n	$T(n)$	
1	1	
2	$2\times1+1=3$	$4-1$
3	$2\times3+1=7$	$8-1$
4	$2\times7+1=15$	$16-1$

Claim (n): $\quad T(n) = \underbrace{2^n - 1}_{RHS(n)}$

Step 1: $\quad n=1 \quad \left.\begin{array}{l} T(1)=1 \\ RHS(1) = 2-1 = 1 \end{array}\right\}$ Claim (1) is true.

Step 2: Let us suppose claim (k) is true, $k \in \mathbb{N}$.

$$T(k) = 2^k - 1$$

$$\begin{aligned} T(k+1) &= 2\,T(k) + 1 \\ &= 2(2^k - 1) + 1 \\ &= 2^{k+1} - 1 = RHS(k+1) \end{aligned}$$

The method of proof by induction allows us to conclude that Claim (n) is true, $\forall n \in \mathbb{N}$.