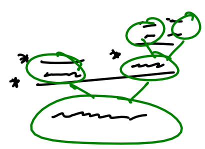
Proof Terminology

* Proposition — a statement which is either true or false
  - $1+1=2$ true
  - $1+1=3$ false
  - $(\text{Function } a \to a-1)\ 4 \Longrightarrow 3$ true
  - $4 \not\Rightarrow$ false
  - $\emptyset \vdash a : \text{Int}$ false
* Proof — a logical demonstration of truth of a proposition
  every step follows directly from things that we assume or know from prev. steps
* Proof tree — a tree-shaped proof based on inference rules



Induction

1. Define proposition fuction: $P(n) \equiv$ (some proposition)
2. Prove $P(0)$.
3. Prove that, given $P(n)$ for <u>any</u> n, $P(n+1)$ is true.

```
let rec sum lst =
  match lst with
  | [] -> 0
  | h :: t -> h + sum t
;;
```

Propose: sum lst will evaluate to the sum of all numbers in lst.

1. Let $P(n) \equiv$ "sum lst will evaluate to the sum of all numbers in lst when lst has length n."

2. Prove $P(0)$; that is: prove sum lst will evaluate to the sum of all numbers in lst when lst is empty.

   If lst = [], then sum lst will evaluate to 0 (by the semantics of match). 0 is the additive identity and so is the sum of no numbers.

```
let rec sum lst =
  match lst with
  | [] -> 0
  | h :: t -> h + sum t
;;
```

3. Prove that $P(n)$ implies $P(n+1)$.

   Want to show that sum lst evaluates to the sum of all numbers in lst when lst has length $n+1$. We know that $n+1 > 0$ (as $n \in \mathbb{N}$). So lst is not empty. So the function will evaluate "h + sum t" and return the result.

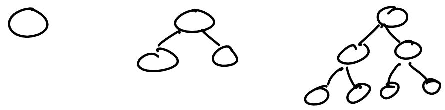
   By inductive hypothesis, "sum t" evaluates to the sum of all numbers in t. So "h + sum t" evaluates to the sum of all numbers in "h :: t". Since h :: t = lst, we are finished.

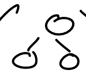
$\square$

QED

Binary Trees

A "perfect" binary tree is either empty, a singleton, or has two children which are perfect binary trees of the same height.

Eg.  not 

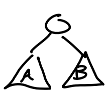
$h($$) = 1 \qquad h(\circ) = 0 \qquad h(\times) = -1$

The number of nodes in a perfect binary tree of height h is $2^{h+1} - 1$.

By induction on the height h of the PBT. We have three cases:

$\times$ 1. The tree is empty. Empty trees have 0 nodes. $0 = 2^{-1+1} - 1 = 2^0 - 1 = 1 - 1 = 0$

$\circ$ 2. The tree is a singleton. Singleton trees have 1 node. $1 = 2^{0+1} - 1 = 2 - 1 = 1$



3. The tree has two children of equal height which are PBTs.
Let the children be named A and B. By defn of "height", at least one child has height $h-1$. Because this is a PBT, both A and B have height $h-1$.
By inductive hypothesis, each of A and B have $2^h - 1$ nodes. Together, they have $2^{h+1} - 2$ nodes. With the root, we have $2^{h+1} - 1$ nodes. $\square$

Strong Induction:
* $P(0)$
* $P(0) \wedge P(1) \wedge \cdots \wedge P(n)$ implies $P(n+1)$

$\boxed{\text{BOOL}}$ $e ::= v \mid e \text{ And } e \mid e \text{ Or } e \mid \text{Not } e$

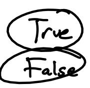
$v ::= \text{True} \mid \text{False}$

"BOOL is normalizing."

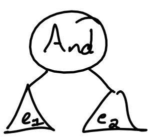
Normalizing: $\forall e \exists v.\ e \Rightarrow v$

$\boxed{\text{Proof}}$

By induction on the height of e.



If e has height 0, then e is a value v. By the Value rule, $v \Rightarrow v$ so $e \Rightarrow v$.
If e has height >0, then e is of one of the forms e_1 And e_2, e_1 Or e_2, or Not e_1.
We have three cases.



If $e = e_1$ And e_2: observe e_1 and e_2 have height less than e. So by induction $e_1 \Rightarrow v_1$ and $e_2 \Rightarrow v_2$ for some v_1 and v_2. All pairs of v_1 and v_2 have a defined logical conjuction. So by the And rule, $e \Rightarrow v$.

$$\text{And}\ \frac{e_1 \Rightarrow v_1 \qquad e_2 \Rightarrow v_2 \qquad v \text{ is the logical conjunction of } v_1 \text{ and } v_2}{e_1 \text{ And } e_2 \Rightarrow v}$$