Metatheory

Proposition: a statement that is either true or false

- $4+1=5$ — true
- $4+1=6$ — false
- Not True $\Rightarrow$ False — true
- False $\not\Rightarrow$ — false

Proof: a series of steps demonstrating a proposition to be true where each step is either from an assumption or immediate from previous steps

Proof Tree: the tree structure an inference proof uses to build from axioms to a conclusion

$$\dfrac{\dfrac{\dfrac{}{\varnothing \vdash 1 : Int} \qquad \dfrac{}{\varnothing \vdash 2 : Int}}{\varnothing \vdash 1+2 : Int} \qquad \dfrac{}{\varnothing \vdash 3 : Int}}{\varnothing \vdash 1+2+3 : Int}$$

Natural Induction

1. Define a proposition function $P(n)$
2. Prove $P(b)$ (for base case b)
3. Assuming $P(k)$ is true, prove $P(k+1)$ is true.

Conclusion: $\forall n \in \mathbb{Z}.\ n \geq b$ then $P(n)$ is true.

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

Prove that "sumlist lst" evaluates to the sum of all elements in lst.

By induction on length of lst. Let $P(n)$ be the statement "the expression 'sumlist lst' evaluates to the sum of all elements in lst when lst has length n".

We show $P(0)$. If lst has length 0 then lst = []. By the operational semantics of match in OCaml, sumlist lst evaluates to 0. 0 is the sum of no elements; this case is complete.

We show, assuming $P(k)$, that $P(k+1)$. If lst has length $k>0$, by opsem of OCaml match, we let h be the first element of lst and t be the rest of the list. If lst has length $k+1$ then t has length k. By the inductive hypothesis, "sumlist t" evaluates to the sum of all elements in t. So "h + sumlist t" evaluates to the sum of all elements in lst. This case is finished.

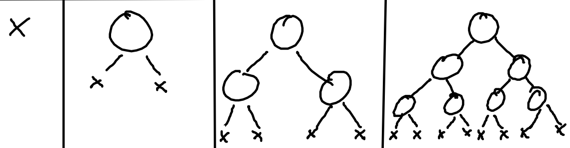
By principle of mathematical induction, $P(n)$ is true for all $n \geq 0$.

$\square$

QED

A perfect binary tree (pbt) is either empty or has two subtrees which are pbts of the same height.

h	-1	0	1	2



Prove that all pbts have $2^{h+1}-1$ nodes where h is the height of the tree.

Proof: by induction on height of pbt.

$$P(n) \equiv \text{A pbt of height } n \text{ has } 2^{n+1}-1 \text{ nodes.}$$

Base case $P(-1)$: a pbt of height -1 is an empty tree and has 0 nodes.

$$2^{-1+1}-1 = 2^0 - 1 = 1 - 1 = 0. \checkmark$$

Inductive step: $P(k)$ implies $P(k+1)$. A pbt of height $h+1$ has a root with two subtrees of the same height. These subtrees must have height h. By the inductive hypothesis, because these trees have height h, they each contain $2^{h+1}-1$ nodes. Including the root, this tree has $2\cdot(2^{h+1}-1)+1 = 2^{h+2}-2+1 = 2^{h+2}-1$ nodes. $\checkmark$

BOOL

$$e ::= v \mid e \text{ And } e \mid e \text{ Or } e \mid \text{Not } e$$
$$v ::= \text{True} \mid \text{False}$$

$$\frac{}{v \Rightarrow v} \qquad \frac{e \Rightarrow \text{True}}{\text{Not } e \Rightarrow \text{False}} \qquad \frac{e \Rightarrow \text{False}}{\text{Not } e \Rightarrow \text{True}} \qquad \frac{e_1 \Rightarrow \text{True} \quad e_2 \Rightarrow \text{True}}{e_1 \text{ And } e_2 \Rightarrow \text{True}} \qquad \frac{\text{False} \in \{v_1, v_2\} \quad e_1 \Rightarrow v_1 \quad e_2 \Rightarrow v_2}{e_1 \text{ And } e_2 \Rightarrow \text{False}}$$

$$\frac{e_1 \Rightarrow \text{False} \quad e_2 \Rightarrow \text{False}}{e_1 \text{ Or } e_2 \Rightarrow \text{False}} \qquad \frac{\text{True} \in \{v_1, v_2\} \quad e_1 \Rightarrow v_1 \quad e_2 \Rightarrow v_2}{e_1 \text{ Or } e_2 \Rightarrow \text{True}}$$

"BOOL is normalizing." $\forall e. \exists v.\ e \Rightarrow v.$

Proof by strong induction on height of e.

Base case: e has height 0. Therefore, e is a value v. By Value Rule, $v \Rightarrow v$. ✓

Inductive step: e has height k+1. Therefore, e is one of e_1 And e_2, e_1 Or e_2, or Not e'. Proceed by case analysis on form of e.

- e = Not e'. Then e' has height k. By ind hyp., $e' \Rightarrow v'$ for some v'. If v' is True then Not $e' \Rightarrow$ False. If v' = False, then Not $e' \Rightarrow$ True. So Not $e' \Rightarrow v$ for some v. ✓
- $e = e_1$ And e_2. Then e_1 and e_2 have height $\leq k$. By ind hyp., $e_1 \Rightarrow v_1$ and $e_2 \Rightarrow v_2$. If $v_1 = v_2 =$ True, then e_1 And $e_2 \Rightarrow$ True. Otherwise, either v_1 = False or v_2 = False and so e_1 And $e_2 \Rightarrow$ False. In either case, e_1 And $e_2 \Rightarrow v$ for some v.
- $e = e_1$ Or e_2. Proof in this case proceeds in the same fashion as the e_1 And e_2 case.

QED