Programs

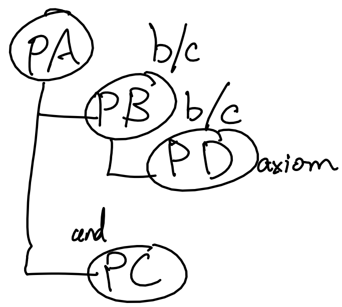
Proofs about programs $\Gamma \vdash e : \tau \setminus E$ ($e \Rightarrow v$)

Proofs about proofs about programs (proofs about proof systems)

Proposition — statement which can be evaluated for its truth

Proof — demonstration of the truth of a proposition

Proof tree —

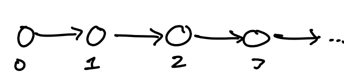


Induction

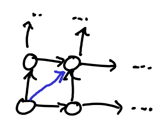
$P(n) \equiv$ ~~~

1. Define a proposition function P on a partial order.
2. Prove base case. For $\mathbb{N}$, $P(0)$.
3. Prove inductive step. For $\mathbb{N}$, $P(n)$ implies $P(n+1)$.

natural numbers



partial orders



Prove that any number of the form $2n$ (for $n \in \mathbb{N}$) is even.

1. $P(n) \equiv 2n$ is even.
2. $P(0) \equiv 0$ is even. By defn.
3. $P(n)$ implies $P(n+1)$. $P(n+1) \equiv 2(n+1)$ is even.

WTS $2n+2$ is even.

Given Any even number $+2$ is even.

Sufficient to show $2n$ is even.

$P(n) \equiv 2n$ is even is our <u>inductive hypothesis</u>. QED.

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

Prove by induction that <u>sum lst</u> returns the sum of all numbers in the integer list <u>lst</u>.

1. $P(n) \equiv$ For a list of length n, <u>sum</u> on that list returns the arithmetic sum of its elements.
2. $P(0) \equiv$ For a list of length 0, <u>sum</u> returns the arithmetic sum of its elements.

   <u>Proof.</u>

   Because xs = [], the match will evaluate the first branch, returning 0.

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

   $P(n+1) \equiv$ For a list of length $n+1$, <u>sum</u> returns the arithmetic sum of the elements in the list.

   Assume $P(n)$.

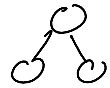
   <u>Proof.</u>

   Because $n \in \mathbb{N}$, $n+1 > 0$. So the match will evaluate the second branch. By ind. hyp. and b/c t has length n, <u>sum t</u> will return the sum of the last n elements. The sum of h and this value is equal to the sum of lst.

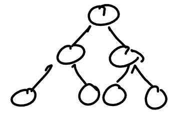
$\square$

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

Induction on Trees



height = 2



height 3

Full binary tree — each level has as many nodes as possible or has 0 nodes

Prove by induction: full binary tree of height h has $2^h - 1$ nodes.

1. $P(n) \equiv$ full b.t. of height n has $2^n - 1$ nodes.
2. $P(1) \equiv$ full b.t. of height 1 has $2^1 - 1 = 1$ node.

   There exists a single tree of height 1. It has 1 node by inspection.

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

   Assume for some $n \in \mathbb{N}$ that full b.t. of height n has $2^n - 1$ nodes. Prove that full b.t. of height $n+1$ has $2^{n+1} - 1$ nodes.

   A full b.t. of height $n+1$ consists of a root node, a left subtree, and a right subtree. Both subtrees are full b.t. of height n. By ind. hyp., they each contain $2^n - 1$ nodes. So full b.t. of height $n+1$ contains $2^n - 1 + 2^n - 1 + 1$ nodes. $2^n + 2^n - 1 \cancel{-1} \cancel{+1} = 2^{n+1} - 1$ ✓

BOOL Normalization

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

"BOOL is normalizing."

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

OpSem

$$\frac{}{v \Rightarrow v} \qquad \frac{e \Rightarrow \text{True}}{\text{Not } e \Rightarrow \text{False}} \qquad \frac{e \Rightarrow \text{False}}{\text{Not } e \Rightarrow \text{True}} \qquad \ldots\ldots$$

Prove by ind. that BOOL is normalizing.
Prove for all expressions e that, for some v, $e \Rightarrow v$.

For e of height n, $e \Rightarrow v$ for some v.

$P(n) \equiv$ Proof statement holds for e of height n.

Proof.

By induction on the height of e. For a base case, consider all e of height 1. There are two such expressions. Both are values; therefore, Value Rule applies and the expressions evaluate to themselves (as values)

Prove $P(n+1)$ assuming $P(n)$ is true.

In the inductive case, our e has one of three forms. Proceed by case analysis on the form of e.

$e =$

If $e = \text{Not } e'$, then e' has lesser height than e. By the inductive hypothesis, $e' \Rightarrow v'$ for some v'. This is the premise of the Not rule. Therefore, $\text{Not } e' \Rightarrow v$ for some v. So $e \Rightarrow v$ and this case is finished.

If $e = e_1 \text{ And } e_2$ then e_1 and e_2 have lesser height than e. By ind. hyp. $e_1 \Rightarrow v_1$ and $e_2 \Rightarrow v_2$ for some v_1 and v_2. These are the premises of the And rule. So $e_1 \text{ And } e_2 \Rightarrow v$ for some v. So $e \Rightarrow v$ and this case is finished.

If $e = e_1 \text{ Or } e_2$ then

Exactly the same as And except using "Or".

QED

"The Or case proceeds in the same fashion as the And case."

Weak induction

$$P(n) \xRightarrow{\text{implies}} P(n+1)$$

Strong induction

$$\left(\forall k \in \{0, \ldots, n\}.\ P(k)\right) \Rightarrow P(n+1)$$