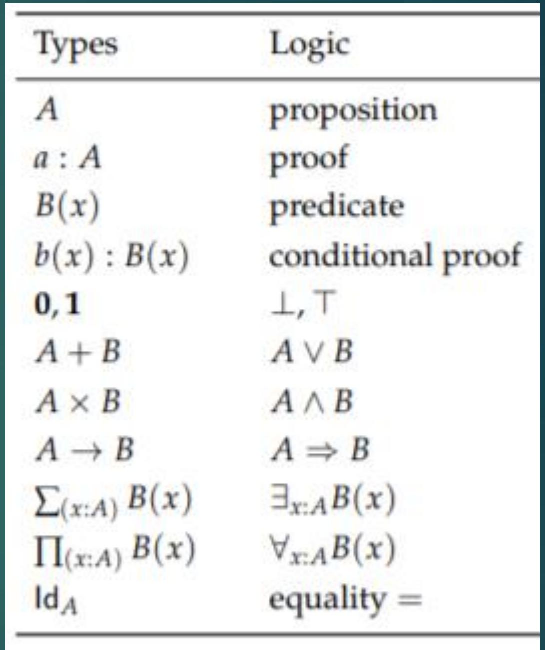
A Taste of PL

THEORY AND APPLICATIONS

PL Research?

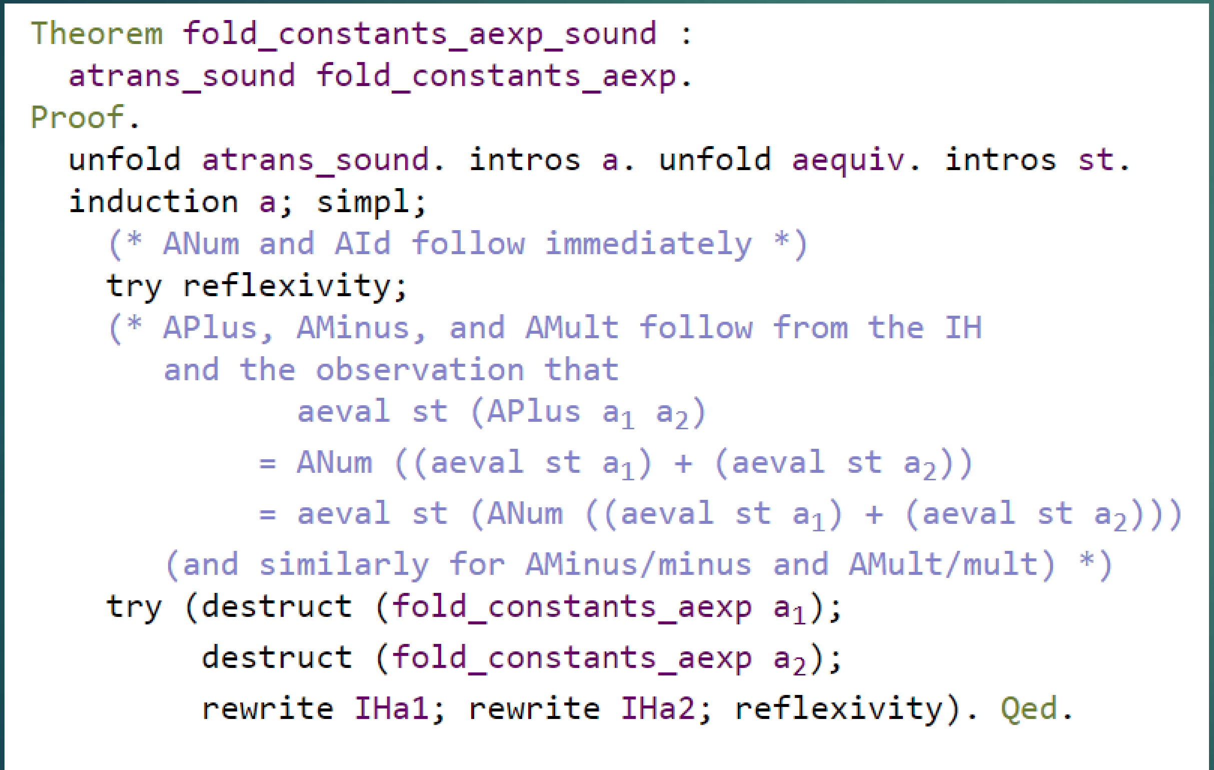
- Software Engineering meets Formal Mathematics
- A programming language as a Logic
  - And proofs as programs
- How to prove things about programs



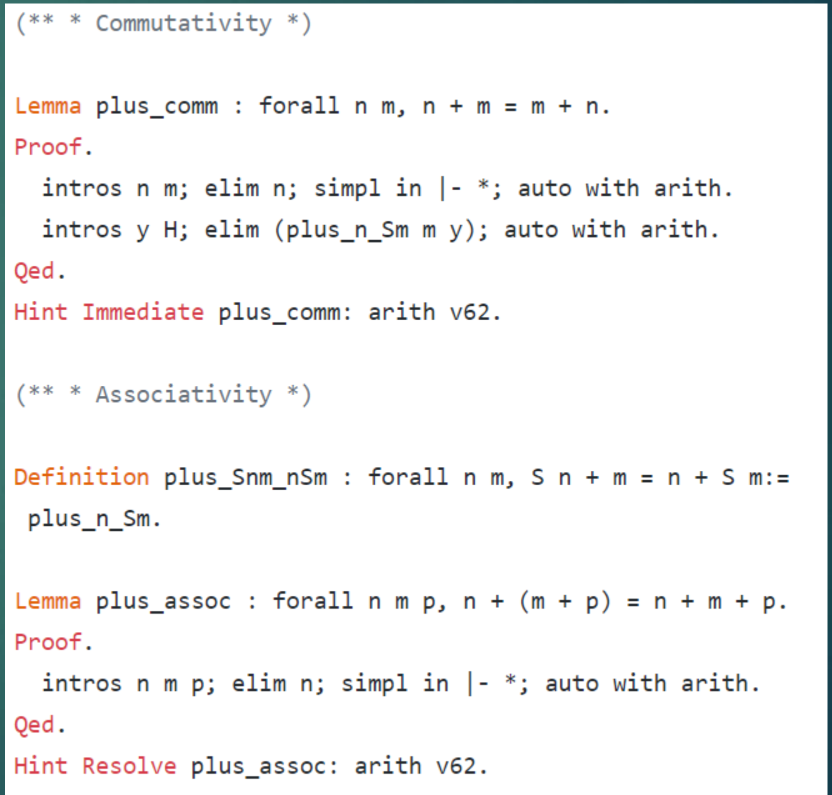
Types	Logic
A	proposition
$a : A$	proof
$B(x)$	predicate
$b(x) : B(x)$	conditional proof
$\mathbf{0}, \mathbf{1}$	$\bot, \top$
$A + B$	$A \vee B$
$A \times B$	$A \wedge B$
$A \to B$	$A \Rightarrow B$
$\sum_{(x:A)} B(x)$	$\exists_{x:A} B(x)$
$\prod_{(x:A)} B(x)$	$\forall_{x:A} B(x)$
Id_A	equality $=$

The Tools

- Coq, Agda, Idris, NuPRL, Cedille …



```
Theorem fold_constants_aexp_sound :
  atrans_sound fold_constants_aexp.
Proof.
  unfold atrans_sound. intros a. unfold aequiv. intros st.
  induction a; simpl;
    (* ANum and AId follow immediately *)
    try reflexivity;
    (* APlus, AMinus, and AMult follow from the IH
       and the observation that
              aeval st (APlus a1 a2)
            = ANum ((aeval st a1) + (aeval st a2))
            = aeval st (ANum ((aeval st a1) + (aeval st a2)))
       (and similarly for AMinus/minus and AMult/mult) *)
    try (destruct (fold_constants_aexp a1);
         destruct (fold_constants_aexp a2);
         rewrite IHa1; rewrite IHa2; reflexivity). Qed.
```

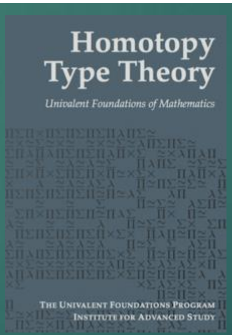


```
(** * Commutativity *)

Lemma plus_comm : forall n m, n + m = m + n.
Proof.
  intros n m; elim n; simpl in |- *; auto with arith.
  intros y H; elim (plus_n_Sm m y); auto with arith.
Qed.
Hint Immediate plus_comm: arith v62.

(** * Associativity *)

Definition plus_Snm_nSm : forall n m, S n + m = n + S m:=
 plus_n_Sm.

Lemma plus_assoc : forall n m p, n + (m + p) = n + m + p.
Proof.
  intros n m p; elim n; simpl in |- *; auto with arith.
Qed.
Hint Resolve plus_assoc: arith v62.
```

Some Projects

- Compcert
- Rust
- Deep Spec

deep spec


Homotopy
Type Theory
Univalent Foundations of Mathematics
The Univalent Foundations Program
Institute for Advanced Study

COMPCERT

Sources

- https://softwarefoundations.cis.upenn.edu/plf-current/Equiv.html#lab22
- https://github.com/maximedenes/native-coq/blob/master/theories/Arith/Plus.v
- https://www.heidelberg-laureate-forum.org/wp-content/uploads/2013/10/Homotopy-Type-Theory_Univalent-Foundations-of-Mathematics.pdf
- https://deepspec.org/page/Project/
- https://www.rust-lang.org/
- http://www.kframework.org/index.php/K_Publications
- http://fsl.cs.illinois.edu/FSL/papers/2015/hathhorn-ellison-rosu-2015-pldi/hathhorn-ellison-rosu-2015-pldi-public.pdf
- http://plv.mpi-sws.org/rustbelt/