FoCaLiZe and Dedukti to the Rescue for Proof Interoperability

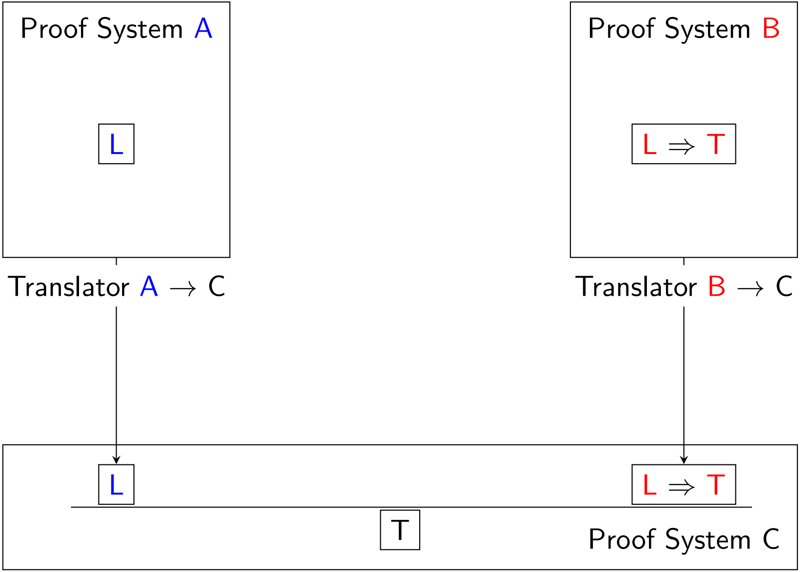
Raphaël Cauderlier [1] Catherine Dubois [2]

[1]Université Paris-Diderot, IRIF, Paris, France

[2]ENSIIE, Samovar, Évry, France

October 23th, 2017

Interoperability



Interoperability: Motivation

- Proof development is expensive
    - 4-color theorem, Kepler conjecture, Feit-Thomson theorem

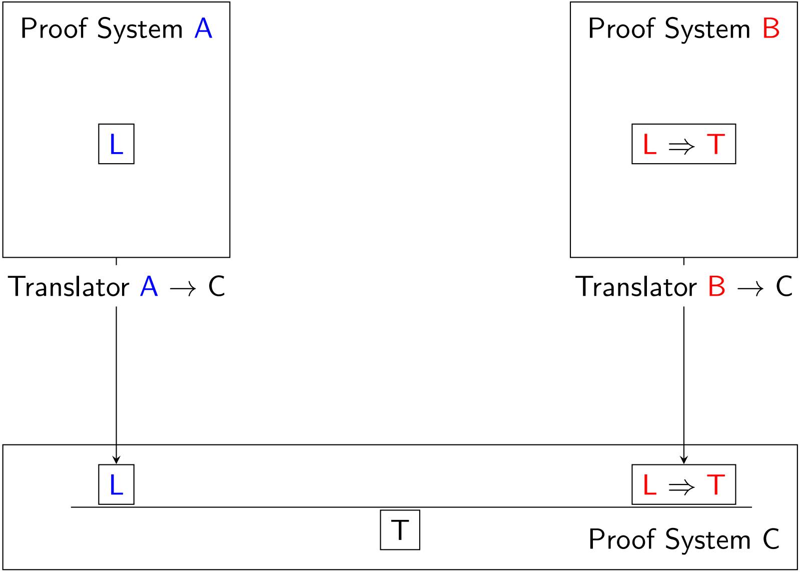
Interoperability: Motivation

- Proof development is expensive
  - 4-color theorem, Kepler conjecture, Feit-Thomson theorem
- Proof assistants are specializing
  - Counterexamples, proof by reflection, decision procedures, ...

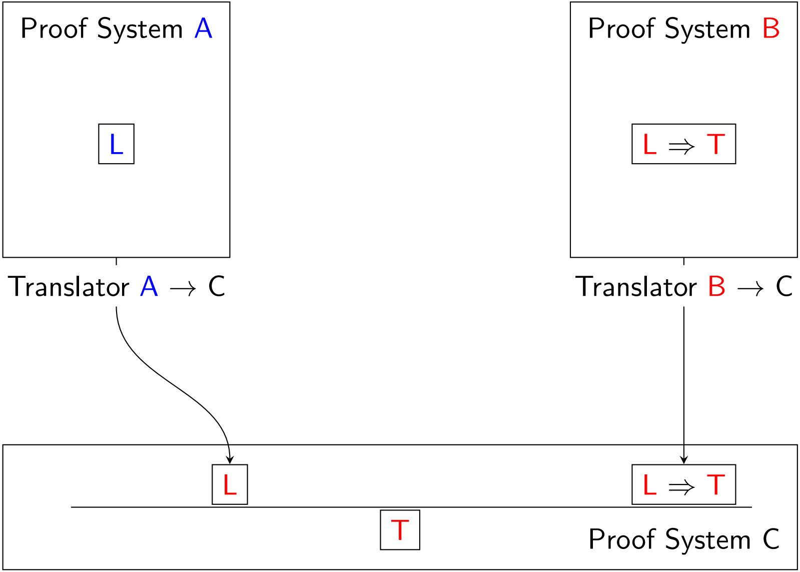
Interoperability: Obstacles

- Logical problem:
  We need to combine the logics of A and B in a consistent way.
- Mathematical problem: L and L are not identical
  Theories such as arithmetic are independently defined in System A and System B.
  We need to identify similar concepts.

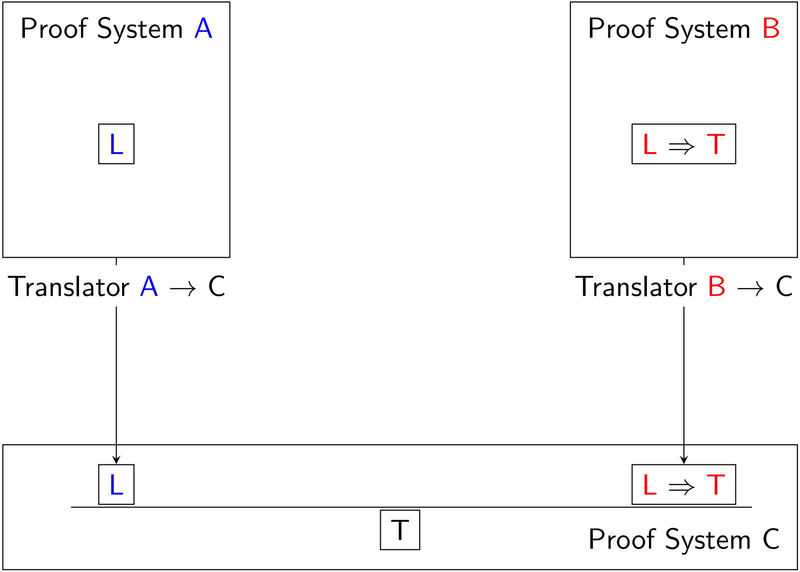
Solution 1/2: Parameterized Translators



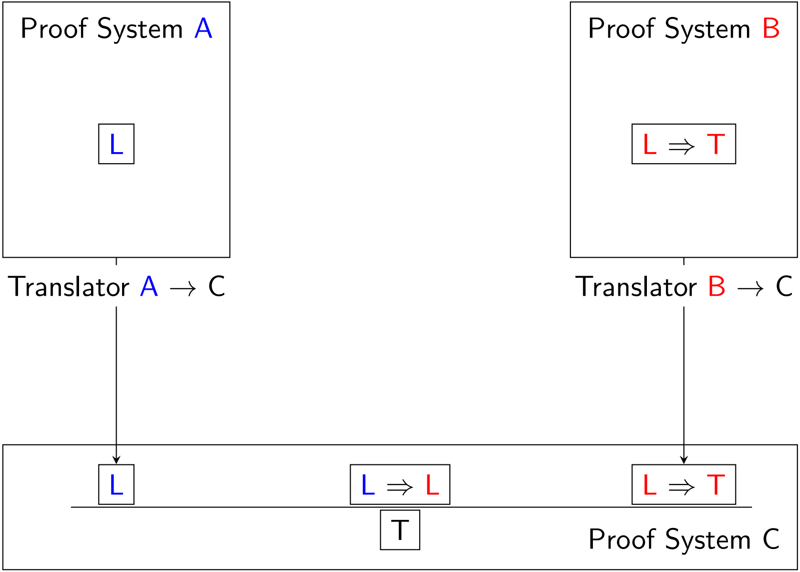
Solution 1/2: Parameterized Translators



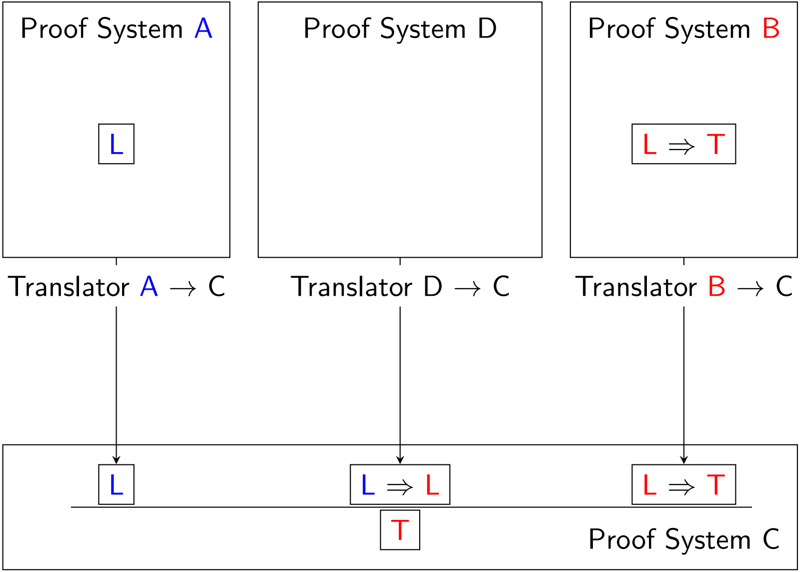
Solution 2/2: Theorem Transfer



Solution 2/2: Theorem Transfer



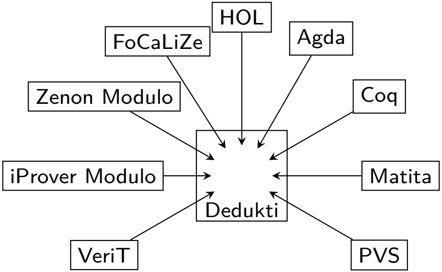
Solution 2/2: Theorem Transfer



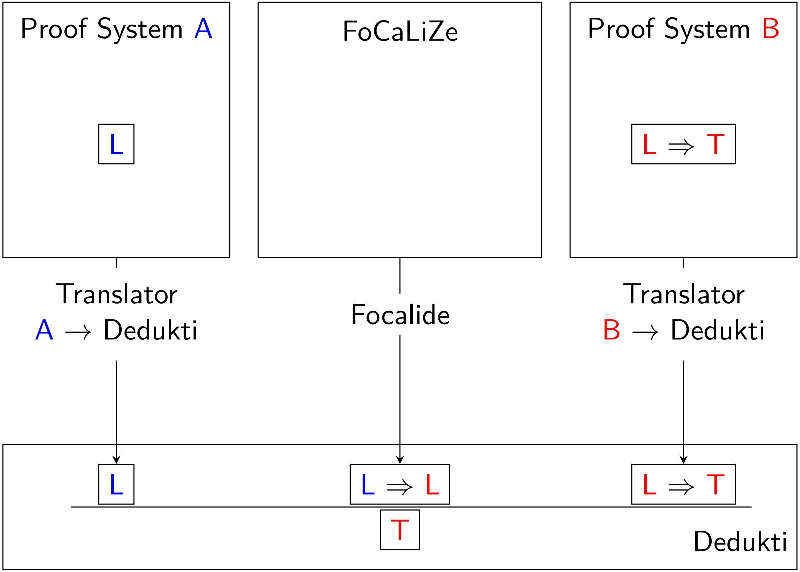
Duplication of Developments

- Solution 1: Parameterized Translators
  Identify syntactically
  Examples:
  - Hol Light $\rightarrow$ Coq (Keller, Werner)
  - Hol Light $\rightarrow$ Isabelle/HOL (Kaliszyk, Krauss)
  - Alignments in MMT (Müller, Rothgang, Liu, Rabe)
- Solution 2: Theorem Transfer
  Identify mathematically, automate reasoning modulo isomorphism
  - `transfer` tactic in Isabelle (Huffman, Kuncar)
  - `transfer` tactic in Coq (Zimmermann, Herbelin)
  - `transfer` tactic in Dedukti (Cauderlier)

Dedukti


HOL
FoCaLiZe
Agda
Zenon Modulo
Coq
iProver Modulo
Matita
Dedukti
VeriT
PVS

Theorem Transfer



Outline

Outline

1. MathTransfer

2. Interoperability Methodology and Case Study

The MathTransfer Library

MathTransfer is a FoCaLiZe library of transfer theorems about natural number arithmetic.
`https://gitlab.math.univ-paris-diderot.fr/cauderlier/math_transfer`

FoCaLiZe, a Formal IDE

- Development of formally verified programs (ML)
- First-order specifications (Poly-FOL)
- Declarative proof language
- Integration of automated theorem provers (Zenon and Zenon Modulo)
- Modularity by object-oriented mechanisms
- Parameterized translators to Ocaml, Coq, and Dedukti

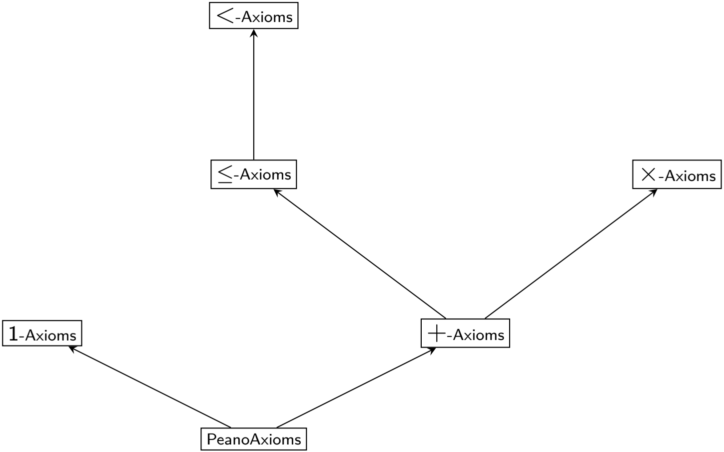
FoCaLiZe Species

- Species contain *methods* acting on the *representation* type of the species
- The representation is written `Self` inside the species
- Methods are either *computational* or *logical*
- Methods are either *declared* or *defined*
    - Declared computational = function/predicate symbol
    - Declared logical = axiom
    - Defined logical = theorem

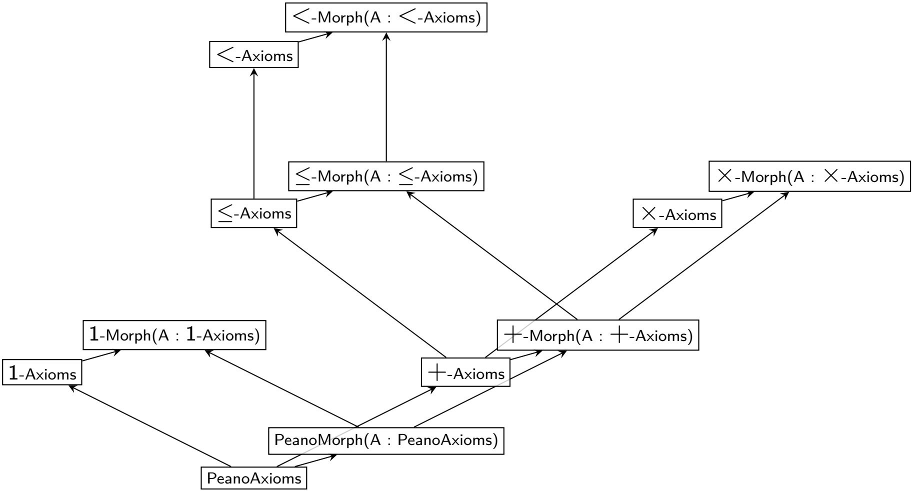
Modularity

- Species are extended by (multiple) inheritance
- During inheritance, we can
  - add new methods
  - define declared methods
  - define the representation
- A species is *complete* if its representation and all its methods are defined
- Complete species can be *instantiated* into *collections*
  - Translated as toplevel definitions and theorems
  - Used to parameterize other species

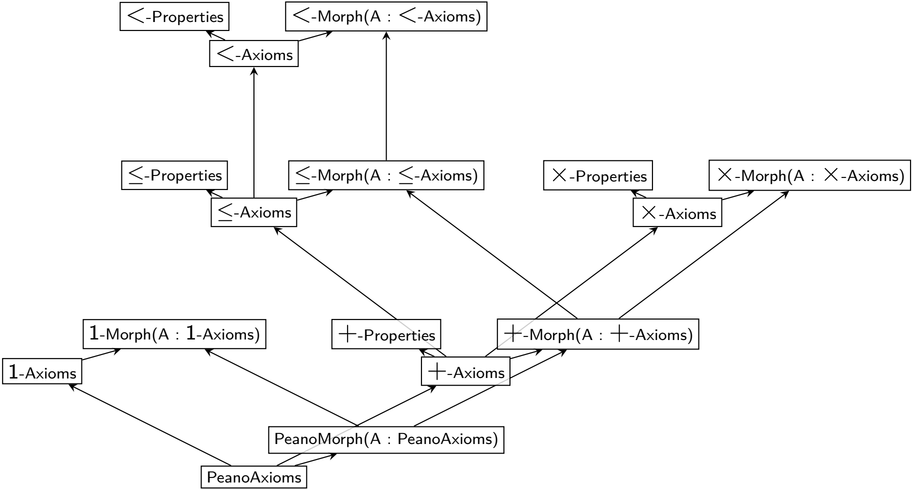
MathTransfer



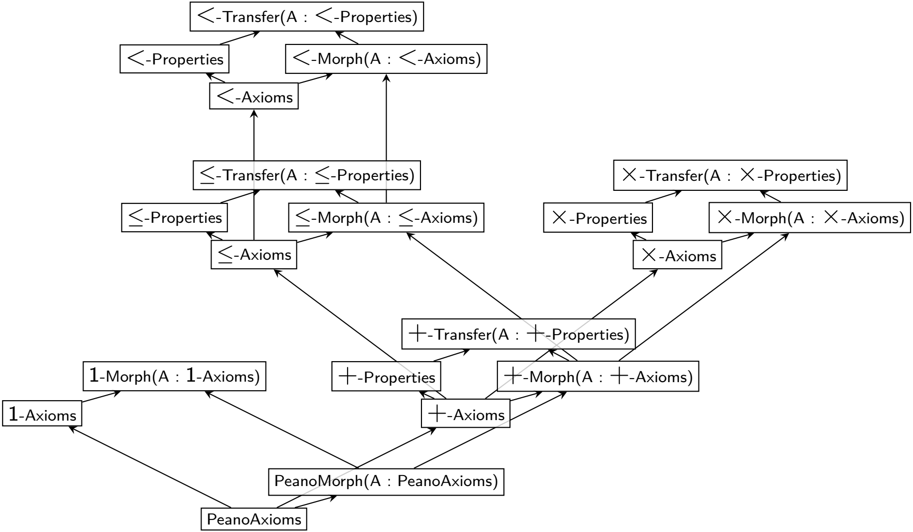
MathTransfer



MathTransfer



MathTransfer



MathTransfer

$\times$-Transfer(A : $\times$-Properties)
theorem $\forall m.\ m \times 0 = 0$
theorem $\forall mn.\ m \times \mathsf{succ}(n) = (m \times n) + m$
theorem $\forall mnp.\ (m \times n) \times p = m \times (n \times p)$
theorem $\forall mn.\ m \times n = n \times m$
theorem $\forall mnp.\ m \times n = m \times p \Leftrightarrow (n = p \vee m = 0)$
theorem $\forall mnp.\ m \times p = n \times p \Leftrightarrow (m = n \vee p = 0)$
theorem $\forall mn.\ m \times n = 0 \Leftrightarrow (m = 0 \vee n = 0)$

$\times$-Properties
axiom $\forall m.\ m \times 0 = 0$
axiom $\forall mn.\ m \times \mathsf{succ}(n) = (m \times n) + m$
axiom $\forall mnp.\ (m \times n) \times p = m \times (n \times p)$
axiom $\forall mn.\ m \times n = n \times m$
axiom $\forall mnp.\ m \times n = m \times p \Leftrightarrow (n = p \vee m = 0)$
axiom $\forall mnp.\ m \times p = n \times p \Leftrightarrow (m = n \vee p = 0)$
axiom $\forall mn.\ m \times n = 0 \Leftrightarrow (m = 0 \vee n = 0)$

$\times$-Morph(A : $\times$-Axioms)
theorem $\forall xy : A.\ f(x \times_A y) = f(x) \times f(y)$

$\times$-Axioms
$\times$: Self $\to$ Self $\to$ Self
axiom $\forall n.\ 0 \times n = 0$
axiom $\forall mn.\ \mathsf{S}(m) \times n = n + (m \times n)$

MathTransfer in numbers

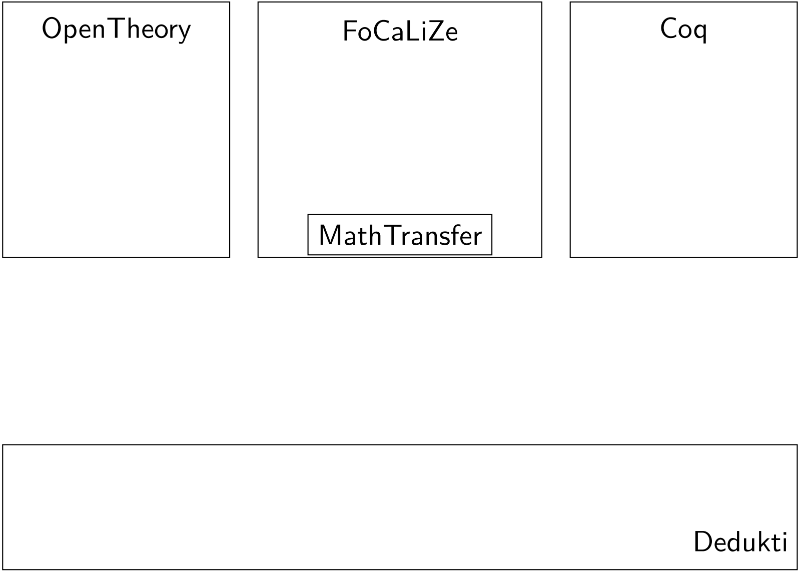
- 11 operations (0, S, 1, bit0, bit1, pred, $+$, $\times$, $\leq$, -, $<$)
- 84 transfer theorems
- 69 species
- 1771 lines, 74KB
- 1.7MB generated Dedukti code (71% from Zenon Modulo and the transfer tactic)

Outline

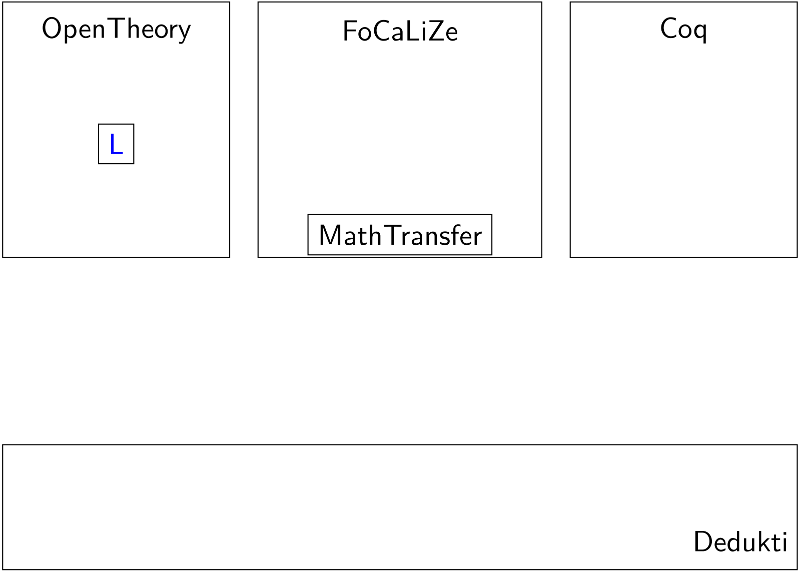
Case Study

- A = HOL (OpenTheory)
- B = Coq
- T = correctness of the Sieve of Eratosthenes

Start



Step 1/8



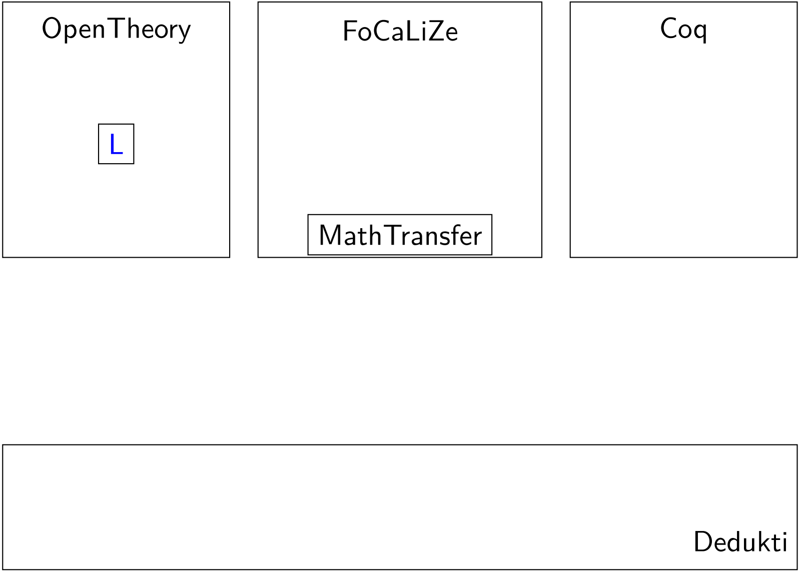
Step 1/8: Identify and prove the lemma in HOL

Prime Divisor Lemma

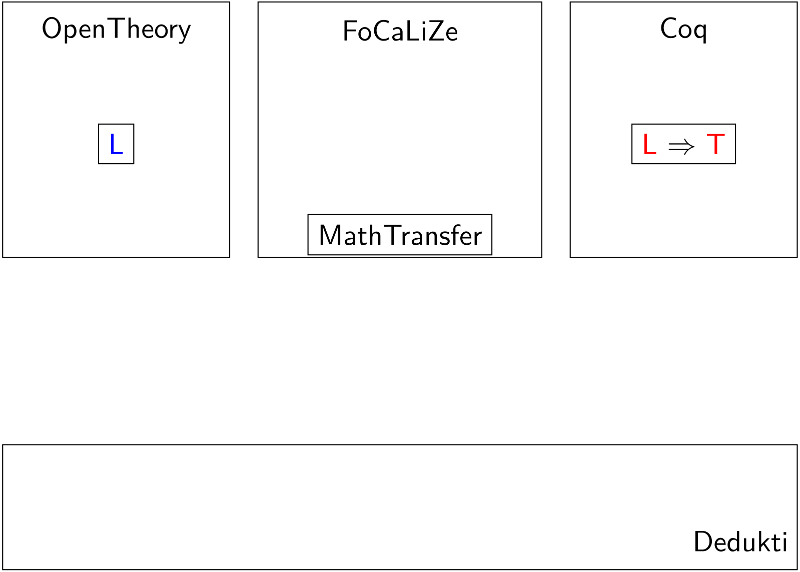
$$\mathsf{L} := \forall n \neq 1.\ \exists p.\ \mathsf{prime}(p) \wedge p \mid n$$

Already proved in OpenTheory `natural-prime` library

Step 1/8



Step 2/8



Step 2/8: Prove L → T in Coq

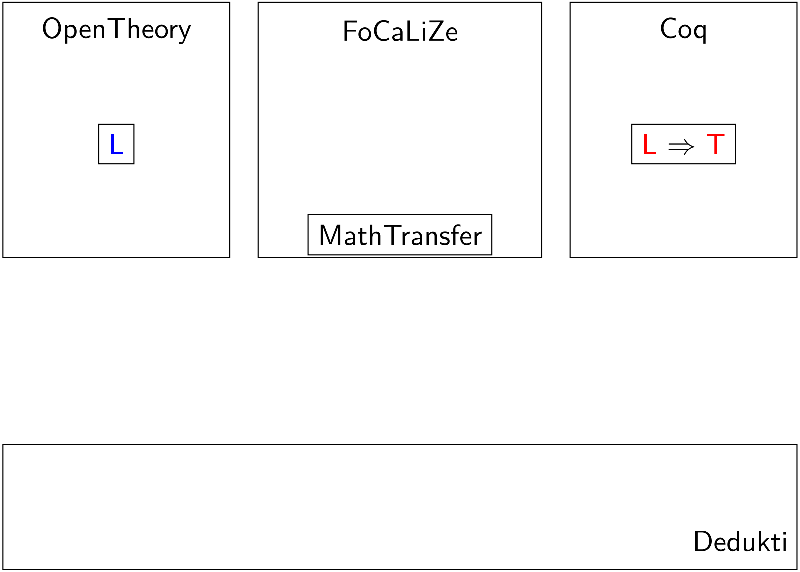
```
…

Definition eratosthenes n := …

Section correctness_proof.

Hypothesis prime_divisor :
  forall n : nat, n <> 1 ->
    exists p : nat, prime p /\ divides p n.

Theorem correctness p n :
  In p (eratosthenes n) <-> (p <= 2 + n /\ prime p).
Proof. … Qed.

End correctness_proof.
```

Step 2/8: Prove L $\to$ T in Coq

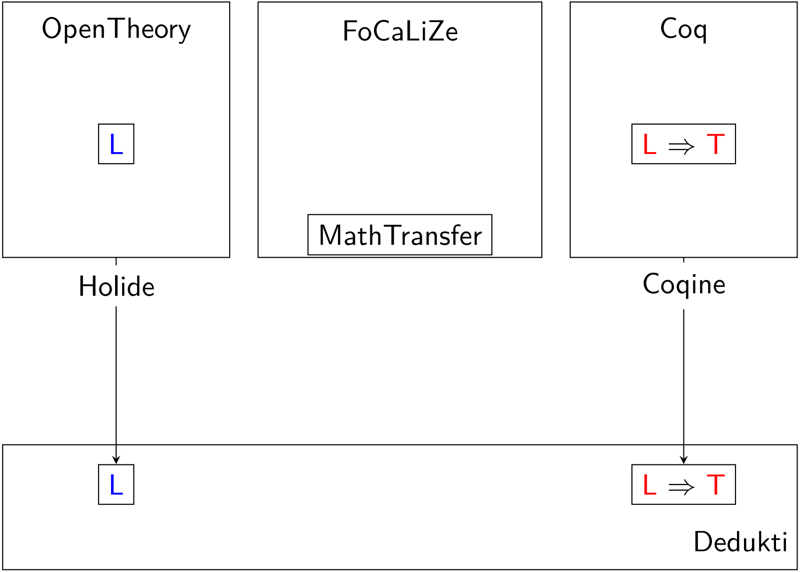
```
…

Definition eratosthenes n := …

Section correctness_proof.

Hypothesis prime_divisor :
  forall n : nat, n <> 1 ->
    exists p : nat, prime p /\ divides p n.

Theorem correctness p n :
  In p (eratosthenes n) <-> (p <= 2 + n /\ prime p).
Proof. … Qed.

End correctness_proof.
```

The whole Coq development takes about 1300 lines (31K).

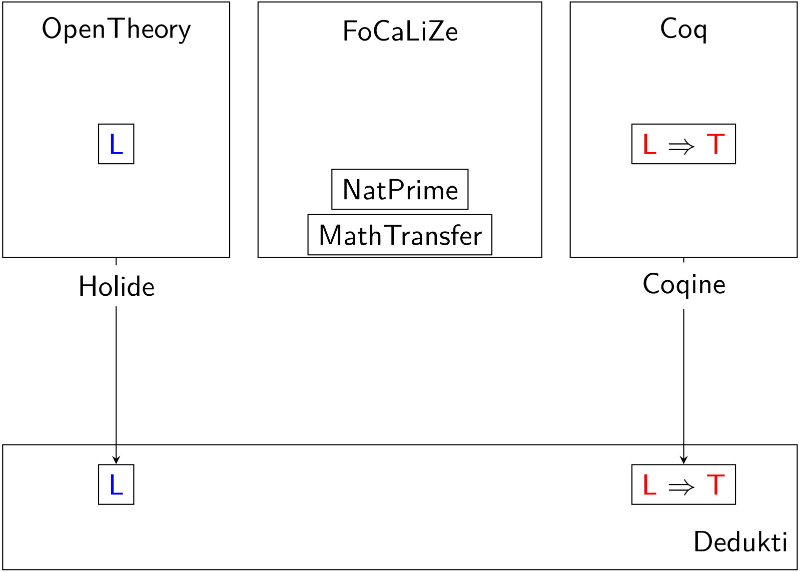
Step 2/8



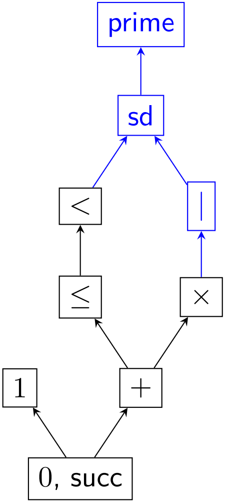
Step 3/8


OpenTheory
L
FoCaLiZe
MathTransfer
Coq
L ⇒ T
Holide
Coqine
L
L ⇒ T
Dedukti

Step 4/8

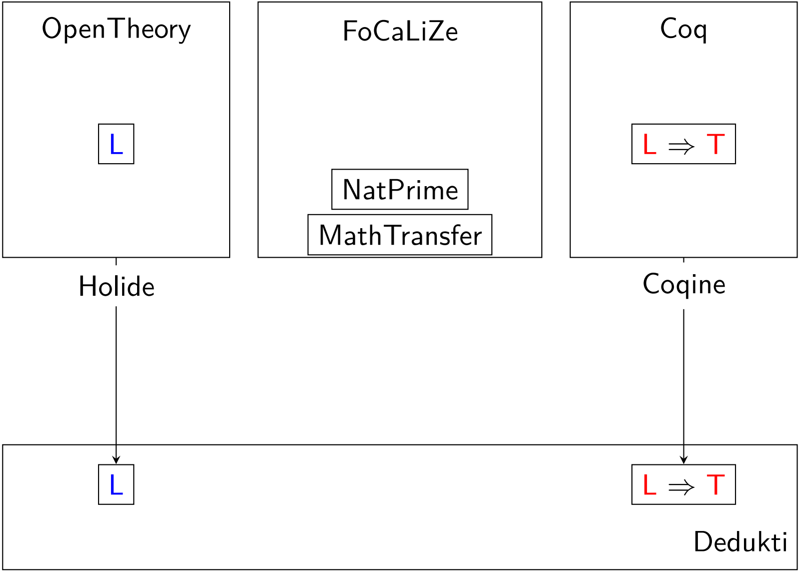

OpenTheory
L
FoCaLiZe
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Coqine
L
L ⇒ T
Dedukti

Step 4/8: Extend the MathTransfer hierarchies

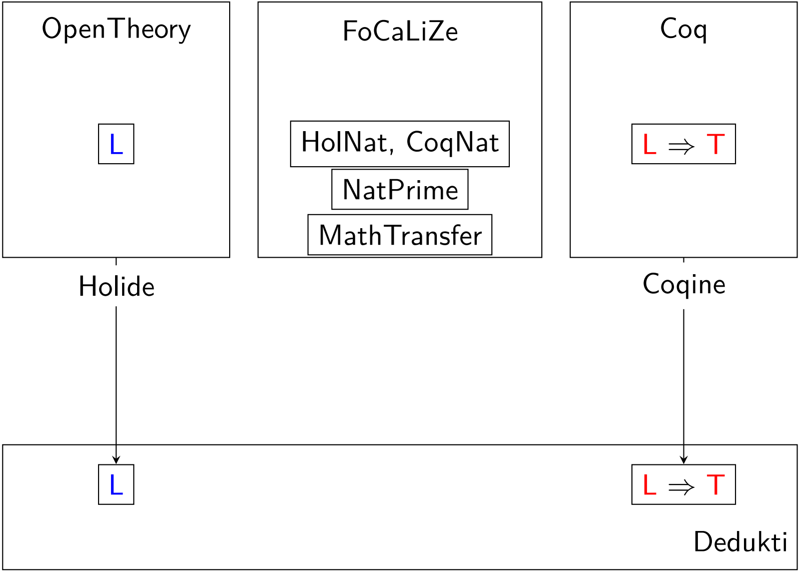


- 3 new operations: divisibility, strict divisibility, and primality
- The morphism hierarchy is also extended
- Properties and transfer hierarchies do not need to be extended

Step 4/8


OpenTheory
L
FoCaLiZe
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Coqine
L
L ⇒ T
Dedukti

Step 5/8


OpenTheory
L
FoCaLiZe
HolNat, CoqNat
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Coqine
L
L ⇒ T
Dedukti

Step 5/8: Instantiate the hierarchy

$\times$-Transfer(A : $\times$-Properties)
- theorem $\forall m.\ m \times 0 = 0$
- theorem $\forall mn.\ m \times \mathrm{succ}(n) = (m \times n) + m$
- theorem $\forall mnp.\ (m \times n) \times p = m \times (n \times p)$
- theorem $\forall mn.\ m \times n = n \times m$
- theorem $\forall mnp.\ m \times n = m \times p \Leftrightarrow (n = p \vee m = 0)$
- theorem $\forall mnp.\ m \times p = n \times p \Leftrightarrow (m = n \vee p = 0)$
- theorem $\forall mn.\ m \times n = 0 \Leftrightarrow (m = 0 \vee n = 0)$

$\times$-Properties
- axiom $\forall m.\ m \times 0 = 0$
- axiom $\forall mn.\ m \times \mathrm{succ}(n) = (m \times n) + m$
- axiom $\forall mnp.\ (m \times n) \times p = m \times (n \times p)$
- axiom $\forall mn.\ m \times n = n \times m$
- axiom $\forall mnp.\ m \times n = m \times p \Leftrightarrow (n = p \vee m = 0)$
- axiom $\forall mnp.\ m \times p = n \times p \Leftrightarrow (m = n \vee p = 0)$
- axiom $\forall mn.\ m \times n = 0 \Leftrightarrow (m = 0 \vee n = 0)$

$\times$-Morph(A : $\times$-Axioms)
- theorem $\forall xy : A.\ f(x \times_A y) = f(x) \times f(y)$

$\times$-Axioms
- $\times$: Self $\rightarrow$ Self $\rightarrow$ Self
- axiom $\forall n.\ 0 \times n = 0$
- axiom $\forall mn.\ S(m) \times n = n + (m \times n)$

Step 5/8: Instantiate the hierarchy

$\times$-Coq
let $(\times)$:= external Dedukti "Coq.mult"
theorem $\forall m.\ 0 \times n = 0$:= external Dedukti "Coq.refl"
theorem $\forall mn.\ \mathrm{succ}(m) \times n = (m \times n) + n$:= external Dedukti "Coq.refl"

$\times$-HOL
let $(\times)$:= external Dedukti "HOL.mult"
theorem $\forall m.\ 0 \times n = 0$:= external Dedukti ...
theorem $\forall mn.\ \mathrm{succ}(m) \times n = (m \times n) + n$:= external Dedukti ...
theorem $\forall m.\ m \times 0 = 0$:= external Dedukti ...
theorem $\forall mn.\ m \times \mathrm{succ}(n) = (m \times n) + m$:= external Dedukti ...
theorem $\forall mnp.\ (m \times n) \times p = m \times (n \times p)$:= external Dedukti ...
...

$A :=$ $\times$-HOL

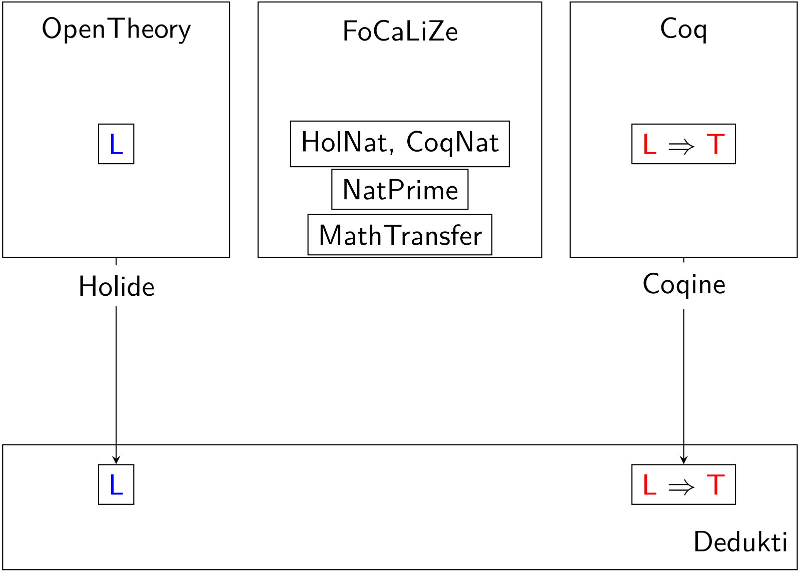
$\times$-Transfer(A : $\times$-Properties)
theorem $\forall m.\ m \times 0 = 0$
theorem $\forall mn.\ m \times \mathrm{succ}(n) = (m \times n) + m$
theorem $\forall mnp.\ (m \times n) \times p = m \times (n \times p)$
...

$\times$-Properties
axiom $\forall m.\ m \times 0 = 0$

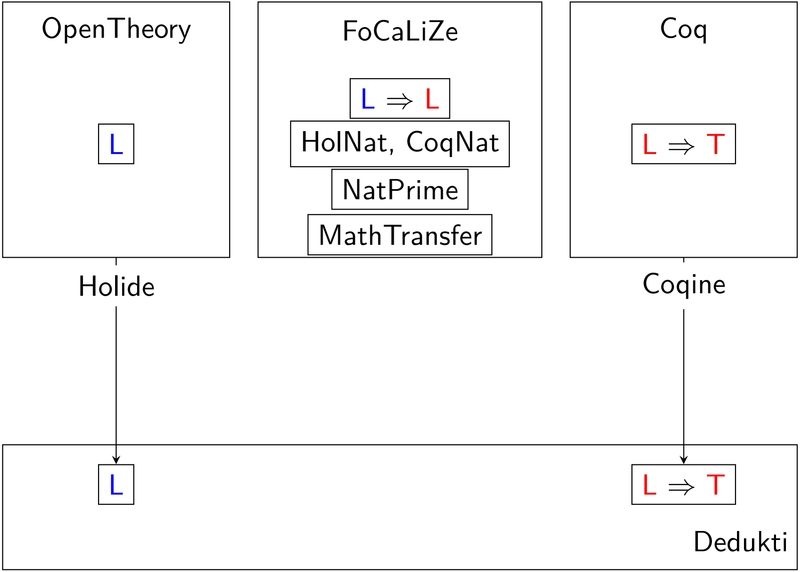
Step 5/8: Instantiate the hierarchy

- Instantiation of the OpenTheory / Coq developments takes benefit of Zenon Modulo
  - Coq: $\mathrm{S}(m) \times n = n + (m \times n)$
  - HOL: $\mathrm{S}(m) \times n = (m \times n) + n$
- Morphisms defined using polymorphic function iteration

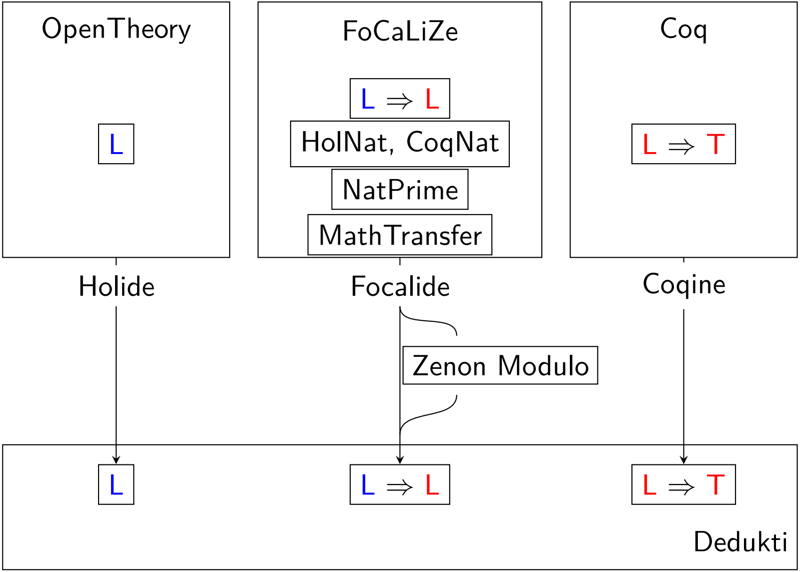
Step 5/8


OpenTheory
L
FoCaLiZe
HolNat, CoqNat
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Coqine
L
L ⇒ T
Dedukti

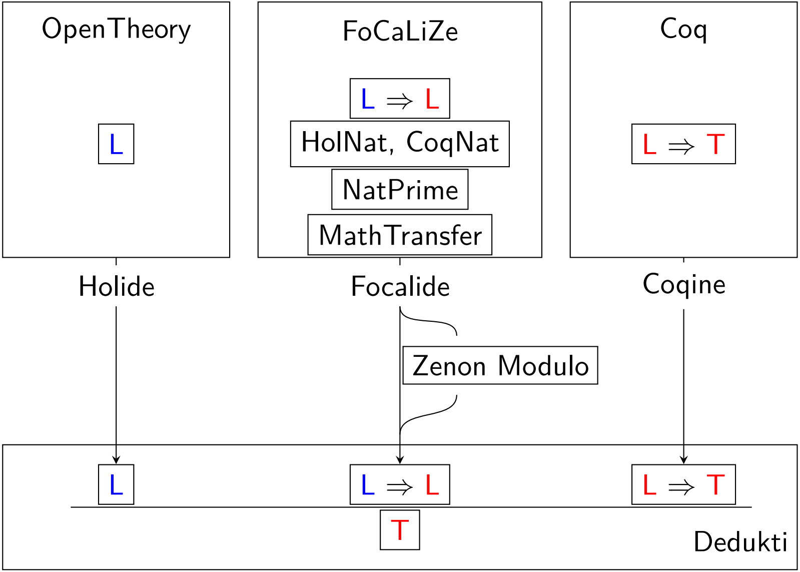
Step 6/8


OpenTheory
L
FoCaLiZe
L ⇒ L
HolNat, CoqNat
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Coqine
L
L ⇒ T
Dedukti

Step 7/8


OpenTheory
L
FoCaLiZe
L ⇒ L
HolNat, CoqNat
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Focalide
Coqine
Zenon Modulo
L
L ⇒ L
L ⇒ T
Dedukti

Step 8/8


OpenTheory
L
FoCaLiZe
L ⇒ L
HolNat, CoqNat
NatPrime
MathTransfer
Coq
L ⇒ T
Holide
Focalide
Coqine
Zenon Modulo
L
L ⇒ L
L ⇒ T
T
Dedukti

Case Study

	Coq	FoCaLiZe	Zenon Modulo	Dedukti
Source Code	31K	55K		9K
Generated Dedukti	828K	784K	597K	

Conclusion

- Contributions
  - The MathTransfer Library
  - An Interoperability Methodology
  - Case Study: Sieve of Eratosthenes in HOL + Coq
- Generic in the proof systems as long as they have:
  - Dedukti translators
  - Merged logics (⚠ consistency of $A \cup B$)

Future Work

- Extend the library ($\mathbb{Z}$, $\mathbb{R}$, algebra, data structures)
- Plug other logics/translators
- Improve proof automation
- Automate the discovery of isomorphic structure across formal libraries
- Develop backward translators (ongoing work by Thiré)

Questions?

Thank you for your attention!