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 between Proof Systems

Interoperability

Proof System A

L

Translator A $\rightarrow$ C

Proof System B

L $\Rightarrow$ T

Translator B $\rightarrow$ C

Proof System C

L $\Rightarrow$ T

T
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 between Proof Systems

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

Proof System A

Proof System B

Proof System C

Translator A $\rightarrow$ C

Translator B $\rightarrow$ C

L $\Rightarrow$ T

L $\Rightarrow$ T

L $\Rightarrow$ T
Solution 2/2: Theorem Transfer

Proof System A

Translator $A \rightarrow C$

Proof System B

Translator $B \rightarrow C$

Proof System C

$\text{L} \rightarrow \text{T}$
Solution 2/2: Theorem Transfer

Proof System A

Translator A → C

L

Translator B → C

L ⇒ T

Proof System B

L ⇒ L

Proof System C

L ⇒ T

L
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)
Interoperability between Proof Systems

Dedukti

- Hol
- FocaLiZe
- Zenon Modulo
- iProver Modulo
- VeriT
- PVS
- Coq
- Matita
- Agda
- PVS
Theorem Transfer

Proof System A

Translator
A → Dedukti

Proof System B

Translator
B → Dedukti

Dedukti

FoCaLiZe

L ⇒ L

L ⇒ T
Outline

1. MathTransfer

2. Interoperability Methodology and Case Study
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

PeanoAxioms

1-Axioms

\(\leq\)-Morph(A : \(\leq\)-Axioms)

\(\leq\)-Axioms

1-Morph(A : 1-Axioms)

\(\geq\)-Morph(A : \(\geq\)-Axioms)

\(\geq\)-Axioms

\(\times\)-Morph(A : \(\times\)-Axioms)

\(\times\)-Axioms

\(\text{PeanoMorph(A : PeanoAxioms)}\)

\(\text{PeanoAxioms}\)
MathTransfer

**Axioms**
- Self \(0\)
- \(\ldots\)

**Properties**
- is transitive
- is antisymmetric
- is reflexive

**PeanoAxioms**
- \(\exists \text{ succ} \) \(\vdash \text{ succ} \text{ is injective} \)
- \(\exists \text{ succ} \) \(\vdash \text{ succ} \text{ is surjective} \)

**MathTransfer**

\[
\begin{align*}
\times\text{-Transfer}(A : \times\text{-Properties}) & \\
\text{theorem} \forall m. m \times 0 = 0 & \\
\text{theorem} \forall mn. m \times \text{ succ}(n) = (m \times n) + m & \\
\text{theorem} \forall mnp. (m \times n) \times p = m \times (n \times p) & \\
\text{theorem} \forall mn. m \times n = n \times m & \\
\text{theorem} \forall mnp. m \times n = m \times p \iff (n = p \lor m = 0) & \\
\text{theorem} \forall mnp. m \times p = n \times p \iff (m = n \lor p = 0) & \\
\text{theorem} \forall mn. m \times n = 0 \iff (m = 0 \lor n = 0) & \\
\end{align*}
\]

\[
\begin{align*}
\times\text{-Morph}(A : \times\text{-Axioms}) & \\
\text{theorem} \forall xy : A. f(x \times_A y) = f(x) \times f(y) & \\
\end{align*}
\]

**\times\text{-Axioms**}
- \(\times : \text{Self} \to \text{Self} \to \text{Self} \)
- \(\text{axiom} \forall n. 0 \times n = 0\)
- \(\text{axiom} \forall mn. S(m) \times n = n + (m \times n)\)
MathTransfer in numbers

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

1. MathTransfer

2. Interoperability Methodology and Case Study
Case Study

- $A = \text{HOL (OpenTheory)}$
- $B = \text{Coq}$
- $T = \text{correctness of the Sieve of Eratosthenes}$
Start

OpenTheory

FoCaLiZe

Coq

MathTransfer

Dedukti
Step 1/8

- OpenTheory
- FoCaLiZe
- MathTransfer
- Coq
- Dedukti
Step 1/8: Identify and prove the lemma in HOL

Prime Divisor Lemma

\[ L := \forall n \neq 1. \exists p. \text{prime}(p) \land p \mid n \]

Already proved in OpenTheory natural-prime library
Step 2/8

- **OpenTheory**: L
- **FoCaLiZe**: MathTransfer
- **Coq**: L \Rightarrow T

**Dedukti**
Step 2/8: Prove L → T in Coq

... 

\textbf{Definition} \textit{eratosthenes} \ n := \ ...

\textbf{Section} \ correctness\_proof.

\textbf{Hypothesis} \ \textit{prime\_divisor} :
  \textbf{forall} \ n : \ \textit{nat}, \ n \ <> \ 1 \ ->
  \textbf{exists} \ p : \ \textit{nat}, \ \textit{prime} \ p \ \&\& \ \textit{divides} \ p \ n.

\textbf{Theorem} \ \textit{correctness} \ p \ n :
  \textit{In} \ p \ (\textit{eratosthenes} \ n) \ <-> \ (p \ <= \ 2 + \ n \ \&\& \ \textit{prime} \ p).
\textbf{Proof}. \ ... \ \textit{Qed}.

\textbf{End} \ correctness\_proof.
Step 2/8: Prove $L \rightarrow T$ in Coq

... 

**Definition** eratosthenes $n := \ldots$

**Section** correctness_proof.

**Hypothesis** prime_divisor :
  **forall** $n : \text{nat}$, $n \neq 1$ ->
  **exists** $p : \text{nat}$, prime $p \land \text{divides } p n$.

**Theorem** correctness $p n$ :
  In $p \ (\text{eratosthenes } n) \leftrightarrow (p \leq 2 + n \land \text{prime } p)$.
**Proof.** ... Qed.

End correctness_proof.

The whole Coq development takes about 1300 lines (31K).
Step 2/8

OpenTheory

FoCaLiZe

MathTransfer

Coq

\[ L \Rightarrow T \]
OpenTheory

FoCaLiZe

Coq

Holide

MathTransfer

Coqine

Dedukti

$L \Rightarrow T$

$L$

$L$

$L$
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

Coq

FoCaLiZe

Holide

Coqine

Dedukti

L

L ⇒ T

NatPrime

MathTransfer

L

L ⇒ T
Step 5/8

OpenTheory

- L

FoCaLiZe

- HolNat, CoqNat
- NatPrime
- MathTransfer

Coq

- L ⇒ T

Holide

- L

Coqine

- L ⇒ T

Dedukti
Step 5/8: Instantiate the hierarchy

- Axioms
  - Axiom 1: $S(0)$
  - Axiom 2: $\forall m. m \times 0 = 0$
  - Axiom 3: $\forall mn. m \times \text{succ}(n) = (m \times n) + m$
  - Axiom 4: $\forall mnp. (m \times n) \times p = m \times (n \times p)$
  - Axiom 5: $\forall mn. m \times n = n \times m$
  - Axiom 6: $\forall mnp. m \times n = m \times p \iff (n = p \lor m = 0)$
  - Axiom 7: $\forall mnp. m \times p = n \times p \iff (m = n \lor p = 0)$
  - Axiom 8: $\forall mn. m \times n = 0 \iff (m = 0 \lor n = 0)$

- Properties
  - Axiom: $\forall m. m \times 0 = 0$
  - Axiom: $\forall mn. m \times \text{succ}(n) = (m \times n) + m$
  - Axiom: $\forall mnp. (m \times n) \times p = m \times (n \times p)$
  - Axiom: $\forall mn. m \times n = n \times m$
  - Axiom: $\forall mnp. m \times n = m \times p \iff (n = p \lor m = 0)$
  - Axiom: $\forall mnp. m \times p = n \times p \iff (m = n \lor p = 0)$
  - Axiom: $\forall mn. m \times n = 0 \iff (m = 0 \lor n = 0)$

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

- Transfer(A : X-Properties)
  - Theorem: $\forall m. m \times 0 = 0$
  - Theorem: $\forall mn. m \times \text{succ}(n) = (m \times n) + m$
  - Theorem: $\forall mnp. (m \times n) \times p = m \times (n \times p)$
  - Theorem: $\forall mn. m \times n = n \times m$
  - Theorem: $\forall mnp. m \times n = m \times p \iff (n = p \lor m = 0)$
  - Theorem: $\forall mnp. m \times p = n \times p \iff (m = n \lor p = 0)$
  - Theorem: $\forall mn. m \times n = 0 \iff (m = 0 \lor n = 0)$
Step 5/8: Instantiate the hierarchy

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

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

\[ A := X - \text{HOL} \]

\[ X - \text{Transfer}(A : X - \text{Properties}) \]
\[
\text{theorem } \forall m. \ m \times 0 = 0 \\
\text{theorem } \forall mn. \ m \times \text{succ}(n) = (m \times n) + m \\
\text{theorem } \forall mnp. \ (m \times n) \times p = m \times (n \times p) \\
\]

\[ X - \text{Properties} \]
\[
\text{axiom } \forall m. \ m \times 0 = 0 
\]
Step 5/8: Instantiate the hierarchy

- Instantiation of the OpenTheory / Coq developments takes benefit of Zenon Modulo
  - Coq: \( S(m) \times n = n + (m \times n) \)
  - HOL: \( S(m) \times n = (m \times n) + n \)

- Morphisms defined using polymorphic function iteration
Step 5/8

OpenTheory

L

Holide

FoCaLiZe

HolNat, CoqNat

NatPrime

MathTransfer

Coq

L ⇒ T

Coqine

L ⇒ T

Dedukti
Step 6/8

Interoperability

- **OpenTheory**
  - L

- **FoCaLiZe**
  - L ⇒ L
  - HolNat, CoqNat
  - NatPrime
  - MathTransfer

- **Coq**
  - L ⇒ T

- **Holide**
  - L

- **Coqine**
  - L ⇒ T

- **Dedukti**
  - L ⇒ T
Step 8/8

OpenTheory

FoCaLiZe

Coq

Holide

Focalide

Coqine

L

HolNat, CoqNat

NatPrime

MathTransfer

L ⇒ L

L ⇒ T

L ⇒ L

T

L ⇒ T

Dedukti
## Case Study

<table>
<thead>
<tr>
<th></th>
<th>Coq</th>
<th>FoCaLiZe</th>
<th>Zenon Modulo</th>
<th>Dedukti</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Code</td>
<td>31K</td>
<td>55K</td>
<td></td>
<td>9K</td>
</tr>
<tr>
<td>Generated Dedukti</td>
<td>828K</td>
<td>784K</td>
<td>597K</td>
<td></td>
</tr>
</tbody>
</table>
### 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)
Thank you for your attention!