Generic Translations between Dedukti Theories

Thomas Traversié

Ongoing work with Florian Rabe (University of Erlangen)

Theories in the $\lambda\Pi$-calculus modulo theory

- Many different theories of the $\lambda\Pi$-calculus modulo theory are **related**
  - "$\mathbb{S}$ can be expressed in $\mathbb{T}$"
  - "$\mathbb{S}$ can be embedded in $\mathbb{T}$"

- We would like to **exchange proofs** from a source theory $\mathbb{S}$ to a target theory $\mathbb{T}$ using **generic** translations that can be instantiated

We identify translation templates for the $\lambda\Pi$-calculus modulo theory

Syntax of the $\lambda\Pi$-calculus modulo theory

- We use the **three-level hierarchy** *à la* LF [Saillard, 2015]

$$\begin{array}{ll} \textit{Objects} & M, N ::= c \mid x \mid \lambda x : A.\ M \mid M\ N \\ \textit{Types} & A, B ::= a \mid \Pi x : A.\ B \mid \lambda x : A.\ B \mid A\ M \\ \textit{Kinds} & K ::= \mathrm{Type} \mid \Pi x : A.\ K \\ \textit{Terms} & t, u ::= M \mid A \mid K \mid \mathrm{Kind} \end{array}$$

- Theories are composed of typed constants and rewrite rules

$$\textit{Theories} \qquad \mathbb{T} ::= \varnothing \mid \mathbb{T}, c : A \mid \mathbb{T}, a : K \mid \mathbb{T}, M \hookrightarrow N \mid \mathbb{T}, A \hookrightarrow B$$

Outline

Theory morphisms in a nutshell

- Principle: represent the **constants** of $\mathbb{S}$ by **terms** in $\mathbb{T}$
- Example: morphism from $\{\wedge, \neg, \forall\}$ to $\{\vee, \neg, \exists\}$
  $\hookrightarrow$ We represent $\wedge$ using $\vee$ and $\neg$
  $\hookrightarrow$ We represent $\forall$ using $\exists$ and $\neg$
- Correspond to **signature morphisms** in LF [Harper et al, 1994]
  Extended to the $\lambda\Pi$-calculus modulo theory [Felicissimo, 2022]

Theory morphisms in LF

- Translation μ

$$
\begin{array}{lcl@{\qquad}lcl}
\mu(x) & = & x & \mu(\lambda x : A.\ M) & = & \lambda x : \mu(A).\ \mu(M) \\
\mu(c) & = & \mu_c \text{ (parameter)} & \mu(\lambda x : A.\ B) & = & \lambda x : \mu(A).\ \mu(B) \\
\mu(a) & = & \mu_a \text{ (parameter)} & \mu(\Pi x : A.\ B) & = & \Pi x : \mu(A).\ \mu(B) \\
\mu(M\ N) & = & \mu(M)\ \mu(N) & \mu(\Pi x : A.\ K) & = & \Pi x : \mu(A).\ \mu(K) \\
\mu(A\ M) & = & \mu(A)\ \mu(M) & \mu(\mathrm{Kind}) & = & \mathrm{Kind} \\
\mu(\mathrm{Type}) & = & \mathrm{Type} & & &
\end{array}
$$

- μ is a **theory morphism** from $\mathbb{S}$ to $\mathbb{T}$ when
  1. for every $c : A \in \mathbb{S}$, there exists a term μ_c such that $\vdash_{\mathbb{T}} \mu_c : \mu(A)$
  2. for every $a : K \in \mathbb{S}$, there exists a term μ_a such that $\vdash_{\mathbb{T}} \mu_a : \mu(K)$

Theory morphisms in the λΠ-calculus modulo theory

- Translation μ

$$
\begin{array}{lcl@{\qquad}lcl}
\mu(x) & = & x & \mu(\lambda x : A.\ M) & = & \lambda x : \mu(A).\ \mu(M) \\
\mu(c) & = & \mu_c \text{ (parameter)} & \mu(\lambda x : A.\ B) & = & \lambda x : \mu(A).\ \mu(B) \\
\mu(a) & = & \mu_a \text{ (parameter)} & \mu(\Pi x : A.\ B) & = & \Pi x : \mu(A).\ \mu(B) \\
\mu(M\ N) & = & \mu(M)\ \mu(N) & \mu(\Pi x : A.\ K) & = & \Pi x : \mu(A).\ \mu(K) \\
\mu(A\ M) & = & \mu(A)\ \mu(M) & \mu(\mathrm{Kind}) & = & \mathrm{Kind} \\
\mu(\mathrm{Type}) & = & \mathrm{Type} & & &
\end{array}
$$

- μ is a **theory morphism** from $\mathbb{S}$ to $\mathbb{T}$ when
  1. for every $c : A \in \mathbb{S}$, there exists a term μ_c such that $\vdash_{\mathbb{T}} \mu_c : \mu(A)$
  2. for every $a : K \in \mathbb{S}$, there exists a term μ_a such that $\vdash_{\mathbb{T}} \mu_a : \mu(K)$
  3. for every $\ell \hookrightarrow r \in \mathbb{S}$, we have $\mu(\ell) \equiv_{\beta\mathcal{R}} \mu(r)$ in $\mathbb{T}$

Theorems about theory morphisms

- **Conversions are preserved** by theory morphisms
  1. If $A \equiv_{\beta\mathcal{R}} B$ in $\mathbb{S}$ then $\mu(A) \equiv_{\beta\mathcal{R}} \mu(B)$ in $\mathbb{T}$
  2. If $K \equiv_{\beta\mathcal{R}} K'$ in $\mathbb{S}$ then $\mu(K) \equiv_{\beta\mathcal{R}} \mu(K')$ in $\mathbb{T}$

- **Representation theorem**
  1. If $\Gamma \vdash_{\mathbb{S}} M : A$ then $\mu(\Gamma) \vdash_{\mathbb{T}} \mu(M) : \mu(A)$
  2. If $\Gamma \vdash_{\mathbb{S}} A : K$ then $\mu(\Gamma) \vdash_{\mathbb{T}} \mu(A) : \mu(K)$
  3. If $\Gamma \vdash_{\mathbb{S}} K : \mathrm{Kind}$ then $\mu(\Gamma) \vdash_{\mathbb{T}} \mu(K) : \mathrm{Kind}$

Example of theory morphisms

- Morphism from subset $\{\wedge, \neg, \forall\}$ to subset $\{\vee, \neg, \exists\}$

- Parameters

$$\mu(\wedge) = \lambda p, q : Prop.\ \neg(\neg p \vee \neg q)$$

$$\mu(\forall) = \lambda a : Set.\ \lambda p : El\ a \to Prop.\ \neg(\exists\ a\ (\lambda x : El\ a.\ \neg(p\ x)))$$

- The rewrite rules for encoding higher-order logic

$$El\ (x \rightsquigarrow y) \hookrightarrow El\ x \to El\ y$$

$$El\ o \hookrightarrow Prop$$

satisfy the condition of theory morphisms

Limitations [Rabe and Sojakova, 2013]

- Church encoding of simple type theory: terms are **intrinsically typed**

$$t : tm\ A$$

- Curry encoding of simple type theory: terms are **externally typed**

$$t : tm \text{ with } \pi : t \# A$$

- Theory morphism from Church to Curry encoding **erases the typing information**

$$t : tm\ A \Longrightarrow \mu(t) : tm$$

- We would like to recover a proof of $\mu(t) \# \mu(A)$

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Logical relations in a nutshell

- Principle: recover the proofs of **invariants** maintained by theory morphisms

- If $M : A$ in $\mathbb{S}$ then $\mu(M) : \mu(A)$ and $\rho(M) : \rho(A)\ \mu(M)$
  – $\rho(A)$ is a **predicate** encoding the invariant
  – $\rho(M)$ is **proof** that $\mu(M)$ satisfies the invariant

- Example: preserve the typing information from Church encoding to Curry encoding

- Logical relations for LF [Rabe and Sojakova, 2013]
  $\approx$ Parametricity for PTS [Bernardy et al, 2010]

Logical relations in LF and in the $\lambda\Pi$-calculus modulo theory

- Translation ρ

$$
\begin{array}{lcl}
\rho(x) & = & x^* \\
\rho(c) & = & \rho_c \text{ (parameter)} \\
\rho(a) & = & \rho_a \text{ (parameter)} \\
\rho(M\ N) & = & \rho(M)\ \mu(N)\ \rho(N) \\
\rho(A\ M) & = & \rho(A)\ \mu(M)\ \rho(M) \\
\rho(\lambda x : A.\ M) & = & \lambda x : \mu(A).\ \lambda x^* : \rho(A)\ x.\ \rho(M) \\
\rho(\lambda x : A.\ B) & = & \lambda x : \mu(A).\ \lambda x^* : \rho(A)\ x.\ \rho(B) \\
\rho(\Pi x : A.\ B) & = & \lambda f : \mu(\Pi x : A.\ B).\ \Pi x : \mu(A).\ \Pi x^* : \rho(A)\ x.\ \rho(B)\ (f\ x) \\
\rho^R(\Pi x : A.\ K) & = & \Pi x : \mu(A).\ \Pi x^* : \rho(A)\ x.\ \rho^{R\ x}(K) \\
\rho^R(\text{Type}) & = & \mu(R) \to \text{Type} \\
\rho(\text{Kind}) & = & \text{Kind}
\end{array}
$$

- Extra parameter R is used because we cannot abstract over types

Logical relations in LF and in the $\lambda\Pi$-calculus modulo theory

- **In LF**: ρ is a logical relation between $\mathbb{S}$ and $\mathbb{T}$ when
  1. for every $c : A \in \mathbb{S}$, there exists a term ρ_c such that $\vdash_{\mathbb{T}} \rho_c : \rho(A)\ \mu(c)$
  2. for every $a : K \in \mathbb{S}$, there exists a term ρ_a such that $\vdash_{\mathbb{T}} \rho_a : \rho^a(K)$

- **In the $\lambda\Pi$-calculus modulo theory**: ρ is a logical relation between $\mathbb{S}$ and $\mathbb{T}$ when
  1. for every $c : A \in \mathbb{S}$, there exists a term ρ_c such that $\vdash_{\mathbb{T}} \rho_c : \rho(A)\ \mu(c)$
  2. for every $a : K \in \mathbb{S}$, there exists a term ρ_a such that $\vdash_{\mathbb{T}} \rho_a : \rho^a(K)$
  3. for every $\ell \hookrightarrow r \in \mathbb{S}$, we have $\rho(\ell) \equiv_{\beta\mathcal{R}} \rho(r)$ in $\mathbb{T}$

- **Conversions are preserved** by logical relations
  1. If $A \equiv_{\beta\mathcal{R}} B$ in $\mathbb{S}$ then $\rho(A) \equiv_{\beta\mathcal{R}} \rho(B)$ in $\mathbb{T}$
  2. If $K \equiv_{\beta\mathcal{R}} K'$ in $\mathbb{S}$ then $\rho^R(K) \equiv_{\beta\mathcal{R}} \rho^R(K')$ in $\mathbb{T}$

- **Abstraction theorem**
  1. If $\Gamma \vdash_{\mathbb{S}} M : A$, then $\rho(\Gamma) \vdash_{\mathbb{T}} \rho(M) : \rho(A)\ \mu(M)$
  2. If $\Gamma \vdash_{\mathbb{S}} A : K$, then $\rho(\Gamma) \vdash_{\mathbb{T}} \rho(A) : \rho^A(K)$
  3. If $\Gamma \vdash_{\mathbb{S}} K : \mathrm{Kind}$ and $\Gamma \vdash_{\mathbb{S}} A : K$, then we have $\rho(\Gamma) \vdash_{\mathbb{T}} \rho^A(K) : \mathrm{Kind}$

Example of logical relations [Rabe and Sojakova, 2013]

- Theory morphism from Church encoding with intrinsically typed terms

$$t : tm\ A$$

to Curry encoding with externally typed terms

$$t : tm \text{ with } \pi : t \# A$$

erases the type

$$t : tm\ A \Longrightarrow \mu(t) : tm$$

- Logical relation allow to **recover the typing information**

$$\rho(t) : \mu(t) \# \mu(A)$$

Outline

Theory embeddings in a nutshell

- Motivation
  - Logical relations provide proofs of the invariants
  - What if the **invariants are essential** to perform the translation?

- Principle
  - We **mutually** define a morphism and a relation
  - The proofs of the invariants are incorporated to the morphisms

- Generalization of interpretation of theories [Traversié, 2024]

Mutually defined translations

- Translations m and r

$$
\begin{array}{lcl}
m(x) & = & x \\
m(c) & = & m_c \text{ (parameter)} \\
m(a) & = & m_a \text{ (parameter)} \\
m(M\ N) & = & m(M)\ m(N)\ r(N) \\
m(A\ M) & = & m(A)\ m(M)\ r(M) \\
m(\lambda x : A.\ M) & = & \lambda x : m(A).\ \lambda x^* : r(A)\ x.\ m(M) \\
m(\lambda x : A.\ B) & = & \lambda x : m(A).\ \lambda x^* : r(A)\ x.\ m(B) \\
m(\Pi x : A.\ B) & = & \Pi x : m(A).\ \Pi x^* : r(A)\ x.\ m(B) \\
m(\Pi x : A.\ K) & = & \Pi x : m(A).\ \Pi x^* : r(A)\ x.\ m(K) \\
m(\text{Type}) & = & \text{Type} \\
m(\text{Kind}) & = & \text{Kind}
\end{array}
$$

$$
\begin{array}{lcl}
r(x) & = & x^* \\
r(c) & = & r_c \text{ (parameter)} \\
r(a) & = & r_a \text{ (parameter)} \\
r(M\ N) & = & r(M)\ m(N)\ r(N) \\
r(A\ M) & = & r(A)\ m(M)\ r(M) \\
r(\lambda x : A.\ M) & = & \lambda x : m(A).\ \lambda x^* : r(A)\ x.\ r(M) \\
r(\lambda x : A.\ B) & = & \lambda x : m(A).\ \lambda x^* : r(A)\ x.\ r(B) \\
r(\Pi x : A.\ B) & = & \lambda f : m(\Pi x : A.\ B).\ \Pi x : m(A). \\
 & & \Pi x^* : r(A)\ x.\ r(B)\ (f\ x\ x^*) \\
r^R(\Pi x : A.\ K) & = & \Pi x : m(A).\ \Pi x^* : r(A)\ x.\ r^{R\ x}(K) \\
r^R(\text{Type}) & = & m(R) \to \text{Type} \\
r(\text{Kind}) & = & \text{Kind}
\end{array}
$$

- r corresponds to logical relations and m now depends on r

Theory embeddings in LF and in the λΠ-calculus modulo theory

- m and r are a **theory embedding** of $\mathbb{S}$ into $\mathbb{T}$ when

  1. for every $c : A \in \mathbb{S}$, there exist terms m_c and r_c such that

     $$\vdash_{\mathbb{T}} m_c : m(A) \text{ and } \vdash_{\mathbb{T}} r_c : r(A)\ m_c$$

  2. for every $a : K \in \mathbb{S}$, there exist terms m_a and r_a such that

     $$\vdash_{\mathbb{T}} m_a : m(K) \text{ and } \vdash_{\mathbb{T}} r_a : r^a(K)$$

  3. for every $\ell \hookrightarrow r \in \mathbb{S}$, we have $m(\ell) \equiv_{\beta\Sigma} m(r)$ and $r(\ell) \equiv_{\beta\Sigma} r(r)$ in $\mathbb{T}$

- In LF, we only have the first two conditions

Embedding theorem

1. If $\Gamma \vdash_{\mathbb{S}} M : A$ then $mr(\Gamma) \vdash_{\mathbb{T}} m(M) : m(A)$
2. If $\Gamma \vdash_{\mathbb{S}} A : K$ then $mr(\Gamma) \vdash_{\mathbb{T}} m(A) : m(K)$
3. If $\Gamma \vdash_{\mathbb{S}} K : \text{Kind}$ then $mr(\Gamma) \vdash_{\mathbb{T}} m(K) : \text{Kind}$

4. If $\Gamma \vdash_{\mathbb{S}} M : A$ then $mr(\Gamma) \vdash_{\mathbb{T}} r(M) : r(A)\ m(M)$
5. If $\Gamma \vdash_{\mathbb{S}} A : K$ then $mr(\Gamma) \vdash_{\mathbb{T}} r(A) : r^A(K)$
6. If $\Gamma \vdash_{\mathbb{S}} K : \text{Kind}$ and $\Gamma \vdash_{\mathbb{S}} A : K$, then we have $mr(\Gamma) \vdash_{\mathbb{T}} r^A(K) : \text{Kind}$

Examples of theory embeddings

- Translation from **natural numbers** to **integers**
  - Impossible to use theory morphism: natural numbers are non-negative integers
  - Invariant inserted: non-negativity of the integers encoding natural numbers

- Translation from **sorted logic** to **unsorted logic**
  - Encode sorts into predicates
  - Invariant inserted: sort predicate

Outline

Implementation in Dedukti

- The user chooses the translation to apply

- The translated file is produced

```
def and_mu :Prop_mu -> Prop_mu -> Prop_mu
  := TODO.
```

and the user must **fill in the parameters**

```
def and_mu :Prop_mu -> Prop_mu -> Prop_mu
  := p => q => not (or (not p) (not q)).
```

- For now, the condition on rewrite rules has to be checked by the user

In practice

- Available on GitHub

https://github.com/thomastraversie/TranslationTemplates

- Several **examples** encoded in DEDUKTI
  - From natural numbers to integers
  - Between subsets of connectives
  - From classical logic to intuitionistic logic
  - From sorted logic to unsorted logic
  - From Church to Curry encoding
  - From deduction to computation

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Takeaway message

- Three **translations templates** for the $\lambda\Pi$-calculus modulo theory
  - Theory morphisms
  - Logical relations
  - Theory embeddings
- **Implemented** in the DEDUKTI language
  - Conditions on constants checked automatically
  - Conditions on rewrite rules not supported yet
- Allow to easily **transfer proofs** between theories