Translation Templates between Dedukti Theories

KWARC Seminar

Thomas Traversié

Logical frameworks

- The **Edinburgh Logical Framework** (LF) [Harper et al, 1993]
  – λ-calculus + dependent types
  – Implemented in Twelf
- The **$\lambda\Pi$-calculus modulo rewriting** [Cousineau and Dowek, 2007]
  – λ-calculus + dependent types + rewrite rules
  – Implemented in Dedukti

The $\lambda\Pi$-calculus modulo rewriting

- Syntax

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

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

- Terms considered modulo the **conversion** $\equiv_{\beta\mathcal{R}}$
  - Generated by β-reduction and rewrite rules
  - Reflexive, symmetric and transitive

Interoperability between proof systems

- Many proof systems = need for **interoperability**
  - Re-use proofs
  - Re-check proofs

- Dedukti = **middleware** between proof systems
  - Exchange proofs to/from Dedukti
  - Exchange proofs inside Dedukti

Goal: automate translations of proofs between Dedukti theories

Translations inside logical frameworks

- Meta-level mechanisms on **LF** logics
  - Theory morphisms [Harper et al, 1994]
  - Logical relations [Rabe and Sojakova, 2013]

- Meta-level mechanisms on **Dedukti** theories
  - Theory morphisms [Felicissimo, 2022]
  - Individual interpretations [T., 2024]
  - No general meta-theorem

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Theory morphisms – Translation

- Principle: **represent** the theory $\mathbb{S}$ inside the theory $\mathbb{T}$

- Translation μ

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

Theory morphisms – Definition

- **In LF**

  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 $\vdash_{\mathbb{T}} \mu_a : \mu(K)$

- **In the $\lambda\Pi$-calculus modulo rewriting**

  Additionally, for every $\ell \hookrightarrow r \in \mathbb{S}$, we require in $\mathbb{T}$

  ✗ The rewrite rule $\mu(\ell) \hookrightarrow \mu(r)$

  ✗ The rewriting $\mu(\ell) \hookrightarrow^*_{\beta\mathcal{R}} \mu(r)$

  ✓ The conversion $\mu(\ell) \equiv_{\beta\mathcal{R}} \mu(r)$

Theorems about theory morphisms

- **Convertibility is 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 $\mathbb{T}$

- **Preservation 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 $\mu(\Gamma) \vdash_{\mathbb{T}} \mu(K) : \mathrm{Kind}$

Church encoding of simple type theory

Terms are intrinsically typed

$tp : \text{Type}$

$\leadsto : tp \to tp \to tp$

$tm : tp \to \text{Type}$

$lam : \Pi A, B : tp.\ (tm\ A \to tm\ B) \to tm\ (A \leadsto B)$

$app : \Pi A, B : tp.\ tm\ (A \leadsto B) \to tm\ A \to tm\ B$

$app\ A\ B\ (lam\ A\ B\ f)\ u \hookrightarrow f\ u$

Curry encoding of simple type theory

Terms are externally typed with the relation relation $\#$

$tp : \text{Type}$

$\rightsquigarrow : tp \to tp \to tp$

$tm : \text{Type}$

$lam : (tm \to tm) \to tm$

$app : tm \to tm \to tm$

$app\ (lam\ f)\ u \hookrightarrow f\ u$

$\# : tm \to tp \to \text{Type}$

$\#_{lam} : \Pi A, B : tp.\ \Pi f : tm \to tm.\ (\Pi x : tm.\ x \# A \to (f\ x) \# B) \to (lam\ f) \# (A \rightsquigarrow B)$

$\#_{app} : \Pi A, B : tm.\ \Pi f, x : tm.\ f \# (A \rightsquigarrow B) \to x \# A \to (app\ f\ x) \# B$

$\#_{app}\ A\ B\ (lam\ f)\ u\ (\#_{lam}\ A\ B\ f\ f^*)\ u^* \hookrightarrow f^*\ u\ u^*$

Church-Curry theory morphism

- Parameters expressed in the Curry encoding

  $\mu(tp) = tp$

  $\mu(\rightsquigarrow) = \rightsquigarrow$

  $\mu(tm) = \lambda A : tp.\ tm$

  $\mu(lam) = \lambda A, B : tp.\ \lambda f : tm \to tm.\ lam\ f$

  $\mu(app) = \lambda A, B : tp.\ \lambda f : tm.\ \lambda x : tm.\ app\ f\ x$

- We check that $\mu(app\ A\ B\ (lam\ A\ B\ f)\ u)$ and $\mu(f\ u)$ are convertible in the Curry encoding

- The theory morphism **erases** the typing information

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

Theory morphisms – Examples

- Translation between different **subsets of connectives**, for instance from $\{\wedge, \neg, \forall\}$ to $\{\vee, \neg, \exists\}$
- Translation from **classical** logic to **intuitionistic** logic
- Translation from natural deduction encoded via **constants** to natural deduction encoded via **rewrite rules**

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Logical relations – Principle

- The Church-Curry morphism erases the typing information

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

  but we would like to **recover** a proof of $\mu(t) \# \mu(A)$

- General idea: 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

Logical relations – Translation

- Translation ρ

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

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

Logical relations – Definition

- **In LF**

  ρ is a logical relation over μ 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 rewriting**

  Additionally, for every $\ell \hookrightarrow r \in \mathbb{S}$, we require $\rho(\ell) \equiv_{\beta\mathcal{R}} \rho(r)$ in $\mathbb{T}$

Theorems about logical relations

- **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 $\rho(\Gamma) \vdash_{\mathbb{T}} \rho^A(K) : \mathrm{Kind}$

Church-Curry logical relation

- Parameters expressed in the Curry encoding

$$\rho(tm) = \lambda A : tp.\ \lambda A^* : unit.\ \lambda x : tm.\ x \# A$$

$$\rho(lam) = \lambda A.\ \lambda A^*.\ \lambda B.\ \lambda B^*.\ \lambda f : tm \to tm.\ \lambda f^* : (\Pi x : tm.\ x \# A \to (f\ x) \# B).$$
$$\#_{lam}\ A\ B\ f\ f^*$$

$$\rho(app) = \lambda A.\ \lambda A^*.\ \lambda B.\ \lambda B^*.\ \lambda f : tm.\ \lambda f^* : f \# (A \rightsquigarrow B).\ \lambda x : tm.\ \lambda x^* : x \# A.$$
$$\#_{app}\ A\ B\ f\ x\ f^*\ x^*$$

- We check that $\rho(app\ A\ B\ (lam\ A\ B\ f)\ u)$ and $\rho(f\ u)$ are convertible in the Curry encoding

- The logical relation **preserves** the typing information

$$t : tm\ A \Longrightarrow \mu(t) : tm \text{ and } \rho(t) : \mu(t) \# \mu(A)$$

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Theory embeddings – Principle

- Motivation
  - Logical relations provide proofs of the invariants
  - What if the **invariants are essential** to perform the translation?
- General idea
  - We **mutually** define a morphism and a relation
  - The proofs of the invariants are incorporated to the morphisms
- Generalization of interpretation of theories [T., 2024]

Theory embeddings – Mutually defined translations

- Translations m and r

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

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

- r corresponds to a logical relation and m now depends on r

Theory embeddings – Definition

- 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

Theorems about theory embeddings

- **Conversions are preserved** by theory embeddings

- **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 $mr(\Gamma) \vdash_{\mathbb{T}} r^A(K) : \text{Kind}$

Theory embeddings – Examples

- Translation from **natural numbers** to **integers**
  Invariant: natural numbers are represented by positive integers

- Translation from **sorted** logic to **unsorted** logic
  Invariant: sorted terms are represented by unsorted terms that satisfy the sort predicate

Theory embeddings – Limitations

- We artificially insert invariants **everywhere**, even when they are not needed
- We can use theory morphisms with **depend pairs** instead, and later eliminate these dependent pairs

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Implementation in Dedukti

- The user specifies the translation to apply, the source file, the target file and the name of the output
- The output file is generated and the user must **fill in the parameters**
  - The conditions on constants are checked automatically
  - The conditions on rewrite rules have to be checked by the user
- All the **results and proofs** of the source theory are now expressed in the target theory

In practice

- Available online

  `https://github.com/Deducteam/TranslationTemplates`

- All the examples have been implemented in Dedukti

Outline

Theory Morphisms

Logical Relations

Theory Embeddings

Implementation

Conclusion

Takeaway message

- Three **translations templates** for the $\lambda\Pi$-calculus modulo rewriting
  - Theory morphisms
  - Logical relations
  - Theory embeddings

- Implemented in Dedukti to easily **transfer proofs** between theories

- Future work
  - Study the limits of the different translation mechanisms
  - Improve theory embeddings