First-order logic in Dedukti

Gilles Dowek

As usual

Difficult (interesting) encodings made first (PTS, CIC, ...)

Easy ones (first-order logic, ...) never written down (like SN($\lambda_{\rightarrow}$))

Yet:

- verifying conservativity is important (Dedukti weak enough to avoid exotic terms, impredicativity ?)
- takes some time and energy to write down the clearest proofs (some surprises sometimes)

The context Δ

A language $\mathcal{L}$

$\iota : Type$,

for each function symbol f (of arity n) of $\mathcal{L}$

$$\dot{f} : \underbrace{\iota \rightarrow \ldots \rightarrow \iota}_{n} \rightarrow \iota,$$

for each predicate symbol f (of arity n) of $\mathcal{L}$

$$\dot{P} : \underbrace{\iota \rightarrow \ldots \rightarrow \iota}_{n} \rightarrow Type$$

Can be extended to many-sorted logic

The translation

- $|x| = x$
- $|f(t_1, ..., t_n)| = (\dot{f}\ |t_1|\ ...\ |t_n|)$
- $|P(t_1, ..., t_n)| = (\dot{P}\ |t_1|\ ...\ |t_n|)$
- $|A \Rightarrow B| = \Pi x : |A|\ |B|$
- $|\forall x\ A| = \Pi x : \iota\ |A|$

The term $|t|$ has type ι, (resp. $|A|$ has type $Type$) in $\Delta, x_1 : \iota, ..., x_n : \iota$

where $\{x_1, ..., x_n\} \supseteq FV(t)$ (resp. $FV(A)$)

Correctness

If $\Gamma \vdash A$ provable then there exists a term π such that

$$\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma| \vdash \pi : |A|$$

where $\{x_1, ..., x_n\} \supseteq FV(\Gamma \vdash A)$

No rewrite rules ($\lambda\Pi$)

Simple induction on the structure of the proof

Conservativity

If

$$\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma| \vdash \pi : |A|$$

where $\{x_1, ..., x_n\} \supseteq FV(\Gamma \vdash A)$

then $\Gamma \vdash A$ provable

The order of the lemmas

Lemma 0: Confluence, termination, existence and uniqueness of normal forms

Lemma 1: A normal well-typed term has the form $Type$, $\Pi x : A\ B$, $\lambda x : A\ t$ or $(f\ t_1\ ...\ t_n)$

Lemma 1': A normal term of type $T : Type$ has the form $\lambda x : A\ t$ or $(f\ t_1\ ...\ t_n)$

Lemma 1": A normal term of an atomic type $T : Type$ has the form $(f\ t_1\ ...\ t_n)$

1, 1', 1" for all rewrite systems and contexts (atomic normal)

The order of the lemmas

Lemma 2: In $\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma|$ a normal term of type ι is the translation of a first-order term

Induction over term structure, from Lemma 1", with a analysis of the possible f's

Finally the Theorem

$$\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma| \vdash \pi : |A|$$

Induction on π

- $\pi = \lambda x : T\ \pi'$, $|A|$ has the form $\Pi x : T\ T'$

  translation of a proposition

  T a proposition or ι, T' a proposition B

  $$\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma|, x : T \vdash \pi' : |B|$$

  (swapping context elements) + IH + intro rule

- $\pi = (f\ t_1\ ...\ t_n)$

  f in $|\Gamma|$

  induction on k, the type of $(f\ t_1\ ...\ t_k)$ is a the translation of a proposition and this proposition is provable in Γ

  - $k = 0$, axiom
  - $((f\ t_1\ ...\ t_k)\ t_{k+1})$

    The type of $(f\ t_1\ ...\ t_k)$ is a product and the translation of a proposition, 2 cases + Lemma 2 or IH + elim

II. Intuitionistic Logic in Dedukti (from Alexis Dorra's work)

The problem and the solution

Include $\wedge$, $\vee$, $\exists$, $\top$, $\perp$

Use the “impredicative” encoding of the connectors in simple type theory

The context Δ

A language $\mathcal{L}$

$$\iota : Type, o : Type,$$

$$\dot{f} : \underbrace{\iota \to \ldots \to \iota}_{n} \to \iota, \dot{P} : \underbrace{\iota \to \ldots \to \iota}_{n} \to o$$

$$\dot{\Rightarrow} : o \to o \to o, \dot{\forall} : (\iota \to o) \to o, \dot{\wedge} : o \to o \to o,$$

$$\dot{\vee} : o \to o \to o, \dot{\exists} : (\iota \to o) \to o, \dot{\top} : o, \dot{\bot} : o$$

$$\varepsilon : o \to Type$$

The translation

Terms translated as usual

- $|P(t_1, ..., t_n)| = (\dot{P}\ |t_1|\ ...\ |t_n|)$
- $|A \Rightarrow B| = (\dot{\Rightarrow}\ |A|\ |B|)$, $|A \wedge B| = (\dot{\wedge}\ |A|\ |B|)$, etc.
- $|\forall x\ A| = (\dot{\forall}\ \lambda x : \iota\ |A|)$, $|\exists x\ A| = (\dot{\exists}\ \lambda x : \iota\ |A|)$

The term $|t|$ has type ι, (resp. $|A|$ has type o) in

$\Delta, x_1 : \iota, ..., x_n : \iota$

where $\{x_1, ..., x_n\} \supseteq FV(t)$ (resp. $FV(A)$)

$\|A\| = (\varepsilon\ |A|)$

Rewrite rules

$$\varepsilon(\dot{\Rightarrow}\ A\ B) \longrightarrow \Pi\_ : \varepsilon(A)\ \varepsilon(B)$$

$$\varepsilon(\dot{\forall}\ A) \longrightarrow \Pi x : \iota\ \varepsilon(A\ x)$$

$$\varepsilon(\dot{\wedge}\ A\ B) \longrightarrow \Pi P : o\ ((\varepsilon(A) \Rightarrow \varepsilon(B) \Rightarrow \varepsilon(P)) \Rightarrow \varepsilon(P))$$

etc.

Two issues to worry about

Impredicativity?

Exotic terms in $\iota \longrightarrow o$?

Correctness

If $\Gamma \vdash A$ provable then there exists a term π such that

$$\Delta, x_1 : \iota, ..., x_n : \iota, \|\Gamma\| \vdash \pi : \|A\|$$

where $\{x_1, ..., x_n\} \supseteq FV(\Gamma \vdash A)$

Simple induction on the structure of the proof

Conservativity

If

$$\Delta, x_1 : \iota, ..., x_n : \iota, \|\Gamma\| \vdash \pi : \|A\|$$

where $\{x_1, ..., x_n\} \supseteq FV(\Gamma \vdash A)$

then $\Gamma \vdash A$ provable

The order of the lemmas

Lemma 0: Confluence, termination, existence and uniqueness of normal forms?

Lemma 1: A normal well-typed term has the form $Type$, $\Pi x : A\ B$, $\lambda x : A\ t$ or $(f\ t_1\ ...\ t_n)$

Lemma 1': A normal term of type $T : Type$ has the form $\lambda x : A\ t$ or $(f\ t_1\ ...\ t_n)$

Lemma 1'': A normal term of an atomic normal type $T : Type$ has the form $(f\ t_1\ ...\ t_n)$

The order of the lemmas

Lemma 2: In $\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma|$ a normal term of type ι is the translation of a first-order term

Lemma 2': In $\Delta, x_1 : \iota, ..., x_n : \iota, |\Gamma|$ a normal term of type o is the translation of a first-order proposition

Induction over term structure, from Lemma 1”, with a analysis of the possible f's

Finally the Theorem

$$\Delta, x_1 : \iota, ..., x_n : \iota, \|\Gamma\| \vdash \pi : \|A\|$$

Induction on π

But ...

A new problem

In minimal logic, recursion in the case $\pi = \lambda x : T\ \pi'$ introduced variable of type $|A|$ (in the case of a $\Rightarrow$) or ι (in the case of a $\forall$)

Handled by the induction hypothesis

$$\Delta, x_1 : \iota, ..., x_n : \iota, \|\Gamma\| \vdash \pi : \|A\|$$

Now if $\rho : (\varepsilon\ A)$ and $\rho' : (\varepsilon\ B)$

$$\lambda P : o\ \lambda \alpha : ((\varepsilon\ A) \Rightarrow (\varepsilon\ B) \Rightarrow (\varepsilon\ P))\ (\alpha\ \rho\ \rho')$$

has type $\Pi P : o\ (((\varepsilon\ A) \Rightarrow (\varepsilon\ B) \Rightarrow (\varepsilon\ P)) \Rightarrow (\varepsilon\ P))$
i.e $\varepsilon(\dot{\wedge}\ A\ B)$

A new problem

In minimal logic, recursion in the case $\pi = \lambda x : T\ \pi'$ introduced variable of type $|A|$ (in the case of a $\Rightarrow$) or ι (in the case of a $\forall$)

Handled by the induction hypothesis

$$\Delta, x_1 : \iota, ..., x_n : \iota, \|\Gamma\| \vdash \pi : \|A\|$$

Now if $\rho : (\varepsilon\ A)$ and $\rho' : (\varepsilon\ B)$

$$\lambda P : o\ \lambda\alpha : ((\varepsilon\ A) \Rightarrow (\varepsilon\ B) \Rightarrow (\varepsilon\ P))\ (\alpha\ \rho\ \rho')$$

has type $\Pi P : o\ (((\varepsilon\ A) \Rightarrow (\varepsilon\ B) \Rightarrow (\varepsilon\ P)) \Rightarrow (\varepsilon\ P))$
i.e $\varepsilon(\dot{\wedge}\ A\ B)$

An very elegant solution

$$\mathcal{L}_p = \mathcal{L} \cup \{P_1, ..., P_p\}$$

$$\Delta_p, x_1 : \iota, ..., x_n : \iota, \|\Gamma\| \vdash \pi : \|A\|$$

then

$$\Gamma \vdash_{\mathcal{L}_p} A$$

Proof: business as usual (induction on the structure of π + a new

Lemma: in first-order logic if $\Gamma \vdash_{\mathcal{L} \cup \{P\}} A$ and B is a proposition in $\mathcal{L}$ then $(B/P)\Gamma \vdash_{\mathcal{L}} (B/P)A$

Two issues to worry about

Impredicativity?

Predicative polymorphism: $(\Pi P : o\ A) : Type$

Two issues to worry about

Exotic terms in $\iota \longrightarrow o$?

Is there a function null : $\iota \longrightarrow o$ that takes the value $\dot{\top}$ at 0 and $\dot{\bot}$ elsewhere?

(What would be $(\dot{\forall}\ \text{null})$ the translation of?)

No

Unlike with inductive types, no closed term such as
$(Rec\ \dot{\top}\ \lambda x \lambda y\ \dot{\bot})$

But ...

In $\lambda\Pi$-modulo we can express a such function if we add the rules

$$(\text{null } 0) \longrightarrow \dot{\top}$$

$$(\text{null } (S\ x)) \longrightarrow \dot{\bot}$$

and a symbol null

If we have a symbol null in $\lambda\Pi$-modulo, we have it in the logic and

$(\dot{\forall} \text{ null})$ is the translation of $\forall x\ (\text{null}(x))$

The rules can be expressed by rules or axioms $\text{null}(0)$ and $\forall x\ (\neg\text{null}(S(x)))$

III. Future work: Permissive Nominal Logic in Dedukti

Permissive Nominal Logic

An extension of first-order logic with binders: $\lambda, \{|\}, \int$

Two kind of variables: bound (x) and quantified (X)

Substitutions of quantified variables must capture bound variables sometimes e.g.

$$\forall T \forall U \ (\mathsf{app}((\lambda x\, T), U) = \mathsf{subst}(T, x, U))$$

To each quantified variable is associated a permission set defining the capturable bound variables

Do we need this logic?

Everything can be done in HOL

Encode binders by λ's (HOAS)

A translation from PNL to HOL

Soundness and completeness proved by semantic means

What about a translation to $\lambda\Pi$ (modulo) and a syntactic proof?

Make Dedukti a prover for PNL