Soundness of embeddings in the $\lambda\Pi$-calculus modulo rewriting

Ali Assaf

Deducteam, Inria Paris-Rocquencourt
École Polytechnique

Parsifal seminar
October 16, 2014

The $\lambda\Pi$-calculus

- Simplest typed λ-calculus with dependent types
- Expresses proofs of first-order logic through the Curry-Howard correspondence
- Used as a logical framework

The $\lambda\Pi$-calculus

sorts	s	$::=$	$\textbf{Type} \mid \textbf{Kind}$
terms	M, N, A, B	$::=$	$x \mid s \mid \Pi x : A.\, B \mid \lambda x : A.\, M \mid M\, N$
contexts	Γ	$::=$	$\cdot \mid \Gamma, x : A$

$$\boxed{\Gamma \vdash M : A}$$

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{}{\Gamma \vdash \textbf{Type} : \textbf{Kind}} \qquad \frac{\Gamma \vdash A : \textbf{Type} \qquad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi x : A.\, B : s}$$

$$\frac{\Gamma \vdash A : \textbf{Type} \qquad \Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B} \qquad \frac{\Gamma \vdash M : \Pi x : A.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\, N : \{N/x\}\, B}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \textbf{Type} \qquad A \equiv_\beta B}{\Gamma \vdash M : B}$$

$$\boxed{WF\,(\Gamma)}$$

$$\frac{}{WF\,(\cdot)} \qquad \frac{WF\,(\Gamma) \qquad \Gamma \vdash A : s}{WF\,(\Gamma, x : A)}$$

Using $\lambda\Pi$ as a logical framework

Logical framework in Wikipedia:

> *In logic, a logical framework provides a means to define (or present) a logic as a signature in a higher-order type theory in such a way that provability of a formula in the original logic reduces to a type inhabitation problem in the framework type theory.*

Using $\lambda\Pi$ as a logical framework

Logical framework in Wikipedia:

> *In logic, a logical framework provides a means to define (or present) a logic as a signature in a higher-order type theory in such a way that provability of a formula in the original logic reduces to a type inhabitation problem in the framework type theory.*

To embed a given theory X in $\lambda\Pi$, one must:

1. define a *signature* context Σ in $\lambda\Pi$ describing the theory X
2. define a *translation* from the terms of X to the terms of $\lambda\Pi$ in the context Σ.

System F in $\lambda\Pi$

Define the signature context Σ as:

$$
\begin{array}{lcl}
\mathsf{type} & : & \mathbf{Type} \\
\mathsf{arrow} & : & \mathsf{type} \to \mathsf{type} \to \mathsf{type} \\
\mathsf{forall} & : & (\mathsf{type} \to \mathsf{type}) \to \mathsf{type} \\
\\
\mathsf{term} & : & \mathsf{type} \to \mathbf{Type} \\
\mathsf{lam} & : & (\mathsf{term}\, A \to \mathsf{term}\, B) \to \mathsf{term}\,(\mathsf{arrow}\, A\, B) \\
\mathsf{app} & : & \mathsf{term}\,(\mathsf{arrow}\, A\, B) \to \mathsf{term}\, A \to \mathsf{term}\, B \\
\mathsf{Lam} & : & (\Pi A : \mathsf{type}.\, \mathsf{term}\,(F\ A)) \to \mathsf{term}\,(\mathsf{forall}\, F) \\
\mathsf{App} & : & \mathsf{term}\,(\mathsf{forall}\, F) \to \Pi A : \mathsf{type}.\, \mathsf{term}\,(F\ A)
\end{array}
$$

System F in $\lambda\Pi$

Translate the types and the terms as:

$$
\begin{aligned}
[\alpha] &= \alpha \\
[A \to B] &= \mathsf{arrow}\,[A]\,[B] \\
[\forall\alpha : \textbf{Type}.\, B] &= \mathsf{forall}\,(\lambda\alpha : \mathsf{type}.\, [B])
\end{aligned}
$$

$$
\begin{aligned}
[x] &= x \\
[\lambda x : A.\, M] &= \mathsf{lam}\,(\lambda x : \mathsf{term}\,[A].\, [M]) \\
[M\,N] &= \mathsf{app}\,[M]\,[N] \\
[\Lambda\alpha : \textbf{Type}.\, M] &= \mathsf{Lam}\,(\lambda\alpha : \mathsf{type}.\, [M]) \\
[M\,\langle A\rangle] &= \mathsf{App}\,[M]\,[A]
\end{aligned}
$$

System F in $\lambda\Pi$

Example

The identity function $\mathrm{id} = \Lambda\alpha : \mathbf{Type}.\, \lambda x : \alpha.\, x$ is translated as:

$$[\mathrm{id}] \quad = \quad \mathsf{Lam}\,(\lambda\alpha : \mathsf{type}.\, \mathsf{lam}\,(\lambda x : \mathsf{term}\,\alpha.\, x))$$

The type $A = \forall\alpha : \mathbf{Type}.\, \alpha \to \alpha$ is translated as:

$$[A] \quad = \quad \mathsf{forall}\,(\lambda\alpha : \mathsf{type}.\, \mathsf{arrow}\,\alpha\,\alpha)$$

Completeness

If M is well-typed then $[M]$ is well-typed in the context Σ:

$$\vdash M : A \quad \Longrightarrow \quad \Sigma \vdash [M] : \mathsf{term}\,[A]$$

Completeness

If M is well-typed then $[M]$ is well-typed in the context Σ:

$$\vdash M : A \quad \Longrightarrow \quad \Sigma \vdash [M] : \mathsf{term}\,[A]$$

Define $[\![A]\!] = \mathsf{term}\,[A]$:

$$\vdash M : A \quad \Longrightarrow \quad \Sigma \vdash [M] : [\![A]\!]$$

Completeness

If M is well-typed then $[M]$ is well-typed in the context Σ:

$$\vdash M : A \quad \Longrightarrow \quad \Sigma \vdash [M] : \mathsf{term}\,[A]$$

Define $[\![A]\!] = \mathsf{term}\,[A]$:

$$\vdash M : A \quad \Longrightarrow \quad \Sigma \vdash [M] : [\![A]\!]$$

If M is well-typed in Γ then $[M]$ is well-typed in the context $\Sigma, [\![\Gamma]\!]$:

$$\Gamma \vdash M : A \quad \Longrightarrow \quad \Sigma, [\![\Gamma]\!] \vdash [M] : [\![A]\!]$$

System F in $\lambda\Pi$

Example

The self-application of id is well-typed in the empty context:

$$\vdash \mathsf{id}\,\langle A\rangle\,\mathsf{id} : A$$

Its translation is well-typed in Σ:

$$\Sigma \vdash \mathsf{app}\,(\mathsf{App}\,[\mathsf{id}]\,[\![A]\!])\,[\mathsf{id}] : [\![A]\!]$$

Completeness

1. If M is a proof of (has type) A in X then $[M]$ is a proof of (has type) $[\![A]\!]$ in $\lambda\Pi$.

Completeness

1. If M is a proof of (has type) A in X then $[M]$ is a proof of (has type) $[\![A]\!]$ in $\lambda\Pi$.
2. If A is provable (is inhabited) in X then $[\![A]\!]$ is provable (is inhabited) in $\lambda\Pi$.

Completeness

1. If M is a proof of (has type) A in X then $[M]$ is a proof of (has type) $[\![A]\!]$ in $\lambda\Pi$.
2. If A is provable (is inhabited) in X then $[\![A]\!]$ is provable (is inhabited) in $\lambda\Pi$.
3. If X is inconsistent (every A is inhabited) then $\lambda\Pi$ is inconsistent (every $[\![A]\!]$ is inhabited).

Completeness

1. If M is a proof of (has type) A in X then $[M]$ is a proof of (has type) $[\![A]\!]$ in $\lambda\Pi$.
2. If A is provable (is inhabited) in X then $[\![A]\!]$ is provable (is inhabited) in $\lambda\Pi$.
3. If X is inconsistent (every A is inhabited) then $\lambda\Pi$ is inconsistent (every $[\![A]\!]$ is inhabited).

What about the converse?

Soundness

1. **Consistency:** if X is consistent then $\lambda\Pi$ is consistent.

Soundness

1. **Consistency:** if X is consistent then $\lambda\Pi$ is consistent.
2. **Conservativity:** if $[\![A]\!]$ is provable in $\lambda\Pi$ then A is provable in X.

Soundness

1. **Consistency:** if X is consistent then $\lambda\Pi$ is consistent.
2. **Conservativity:** if $[\![A]\!]$ is provable in $\lambda\Pi$ then A is provable in X.
3. **Adequacy:** every (normal) proof in $\lambda\Pi$ corresponds to a proof in X.

Soundness

1. **Consistency:** if X is consistent then $\lambda\Pi$ is consistent.
2. **Conservativity:** if $[\![A]\!]$ is provable in $\lambda\Pi$ then A is provable in X.
3. **Adequacy:** every (normal) proof in $\lambda\Pi$ corresponds to a proof in X.

These are important properties for a logical framework!

Summary

- **Source** $= X$, **Target** $= \lambda\Pi$
- **Embedding** = signature Σ + translation $[\cdot]$
- **Completeness** = typing in $X \implies$ typing in $\lambda\Pi$
- **Soundness** = typing in $\lambda\Pi \implies$ typing in X

Limitations of $\lambda\Pi$

The embedding does not preserve term (proof) reduction :

$$M \longrightarrow^* M' \quad \not\Longrightarrow \quad [M] \longrightarrow^* \left[M'\right]$$

Limitations of $\lambda\Pi$

The embedding does not preserve term (proof) reduction :

$$M \longrightarrow^* M' \quad \not\Rightarrow \quad [M] \longrightarrow^* \left[M'\right]$$

The embedding does not preserve term (proof) equivalence:

$$M \equiv M' \quad \not\Rightarrow \quad [M] \equiv \left[M'\right]$$

Limitations of $\lambda\Pi$

Systems with dependent types (e.g. the calculus of constructions) have a conversion rule:

$$\frac{\Gamma \vdash M : A \qquad A \equiv B}{\Gamma \vdash M : B}$$

In $\lambda\Pi$, $[\![\Gamma]\!] \vdash [M] : [\![A]\!]$ but $[\![\Gamma]\!] \nvdash [M] : [\![B]\!]$ (no completeness).

Approach 1: Introduce explicit equivalence judgements and a conversion term:

$$
\begin{array}{rcl}
\mathsf{equiv} & : & \mathsf{type} \to \mathsf{type} \to \mathbf{Type} \\
\mathsf{refl} & : & \mathsf{equiv}\, M\, M \\
\mathsf{beta} & : & \mathsf{equiv}\, (\mathsf{app}\, (\mathsf{lam}\, F)\, N)\, (F\, N) \\
\ldots & & \\
\mathsf{conv} & : & \mathsf{term}\, A \to \mathsf{equiv}\, \mathsf{A}\, \mathsf{B} \to \mathsf{term}\, B
\end{array}
$$

Cons:

- Need to explicitely give the equivalence derivations.
- Adding conv pollutes the structure of the terms and needs to be taken care of in the equivalence relation.

Approach 2: Translate typing derivations instead of λ-terms

$$\begin{array}{rcl}
\text{term} & : & \mathbf{Type} \\
\text{lam} & : & (\text{term} \to \text{term}) \to \text{term} \\
\ldots & & \\
\text{hastype} & : & \text{term} \to \text{type} \to \mathbf{Type} \\
\text{typelam} & : & (\Pi x : \text{term}.\, \text{hastype}\, x\, A \to \text{hastype}\, (F\, x)\, B) \to \\
 & & \text{hastype}\, (\text{lam}\, F)\, (\text{arrow}\, A\, B) \\
\ldots & &
\end{array}$$

Pros:

- conv does not interfere with the structure of the λ-terms.

Cons:

- Lose Curry-Howard correspondence?
- Still need to explicitly give the equivalence derivations.

The $\lambda\Pi$-calculus modulo rewriting

Idea: extend the conversion rule of the $\lambda\Pi$-calculus with a rewrite system R:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \mathbf{Type} \qquad A \equiv_{\beta R} B}{\Gamma \vdash M : B}$$

The $\lambda\Pi$-calculus modulo rewriting

Idea: extend the conversion rule of the $\lambda\Pi$-calculus with a rewrite system R:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \mathbf{Type} \qquad A \equiv_{\beta R} B}{\Gamma \vdash M : B}$$

Add rewrite rules so that the translation preserves reduction.

The $\lambda\Pi$-calculus modulo rewriting

Idea: extend the conversion rule of the $\lambda\Pi$-calculus with a rewrite system R:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \mathbf{Type} \qquad A \equiv_{\beta R} B}{\Gamma \vdash M : B}$$

Add rewrite rules so that the translation preserves reduction (in addition to binding and typing).

Preserving reduction

Signature context Σ:

$$\begin{array}{rcl} \mathsf{type} & : & \mathbf{Type} \\ \mathsf{arrow} & : & \mathsf{type} \to \mathsf{type} \to \mathsf{type} \\ & & \\ \mathsf{term} & : & \mathsf{type} \to \mathbf{Type} \\ \mathsf{lam} & : & (\mathsf{term}\, A \to \mathsf{term}\, B) \to \mathsf{term}\, (\mathsf{arrow}\, A\, B) \\ \mathsf{app} & : & \mathsf{term}\, (\mathsf{arrow}\, A\, B) \to \mathsf{term}\, A \to \mathsf{term}\, B \end{array}$$

Rewrite rules R:

$$\mathsf{app}\, (\mathsf{lam}\, F)\, N \quad \longrightarrow \quad F\, N$$

Preserving reduction

Signature context Σ:

$$\begin{array}{rcl} \mathsf{type} & : & \mathbf{Type} \\ \mathsf{arrow} & : & \mathsf{type} \to \mathsf{type} \to \mathsf{type} \end{array}$$

$$\begin{array}{rcl} \mathsf{term} & : & \mathsf{type} \to \mathbf{Type} \\ \mathsf{lam} & : & (\mathsf{term}\, A \to \mathsf{term}\, B) \to \mathsf{term}\,(\mathsf{arrow}\, A\, B) \\ \mathsf{app} & : & \mathsf{term}\,(\mathsf{arrow}\, A\, B) \to \mathsf{term}\, A \to \mathsf{term}\, B \end{array}$$

Rewrite rules R:

$$\begin{array}{rcl} \mathsf{term}\,(\mathsf{arrow}\, A\, B) & \longrightarrow & \mathsf{term}\, A \to \mathsf{term}\, B \\ \mathsf{lam}\, F & \longrightarrow & F \\ \mathsf{app}\, M\, N & \longrightarrow & M\, N \end{array}$$

Preserving reduction

Signature context Σ:

$$
\begin{array}{rcl}
\mathsf{type} & : & \mathbf{Type} \\
\mathsf{arrow} & : & \mathsf{type} \to \mathsf{type} \to \mathsf{type} \\
\\
\mathsf{term} & : & \mathsf{type} \to \mathbf{Type}
\end{array}
$$

Rewrite rules R:

$$
\mathsf{term}\,(\mathsf{arrow}\,A\,B) \quad \longrightarrow \quad \mathsf{term}\,A \to \mathsf{term}\,B
$$

Translation:

$$
\begin{array}{rcl}
[\lambda x : A.\, M] & = & \lambda x : [\![A]\!].\, [M] \\
[M\,N] & = & [M]\,[N]
\end{array}
$$

Preserving reduction

Theorem

If $M \longrightarrow M'$ *then* $[M] \longrightarrow^{+} [M']$.

Preserving reduction

Theorem

If $M \longrightarrow M'$ *then* $[M] \longrightarrow^{+} [M']$.

Corollary

If $M \longrightarrow^{*} M'$ *then* $[M] \longrightarrow^{*} [M']$.

Preserving reduction

Theorem

If $M \longrightarrow M'$ *then* $[M] \longrightarrow^{+} [M']$.

Corollary

If $M \longrightarrow^{*} M'$ *then* $[M] \longrightarrow^{*} [M']$.

Corollary

If $M \equiv M'$ *then* $[M] \equiv [M']$.

Completeness

Recovered typing preservation.

Theorem (Cousineau & Dowek 2007)

If $\Gamma \vdash M : A$ *then* $\Sigma, [\![\Gamma]\!] \vdash [M] : [\![A]\!]$.

Works for any functional pure type system:

- System F
- Calculus of constructions
- Simple type theory

Completeness

Recovered typing preservation.

Theorem (Cousineau & Dowek 2007)

If $\Gamma \vdash M : A$ *then* $\Sigma, [\![\Gamma]\!] \vdash [M] : [\![A]\!]$.

Works for any functional pure type system:

- System F
- Calculus of constructions
- Simple type theory

What about soundness?

On termination and soundness

Link between termination and soundness:

- $\lambda\Pi$ is strongly normalizing

On termination and soundness

Link between termination and soundness:

- $\lambda\Pi$ is strongly normalizing
- Adding axioms (Σ) does not influence termination

On termination and soundness

Link between termination and soundness:

- $\lambda\Pi$ is strongly normalizing
- Adding axioms (Σ) does not influence termination
- Can be used to show soundness:
  - **Consistency:** there is no normal term of type $[\![\bot]\!]$

On termination and soundness

Link between termination and soundness:

- $\lambda\Pi$ is strongly normalizing
- Adding axioms (Σ) does not influence termination
- Can be used to show soundness:
  - **Consistency:** there is no normal term of type $[\![\bot]\!]$
  - **Adequacy:** if $[\![\Gamma]\!] \vdash M : [\![A]\!]$ and M is a normal form, then $M = [N]$ for some N such that $\Gamma \vdash N : A$

On termination and soundness

Link between termination and soundness:

- $\lambda\Pi$ is strongly normalizing
- Adding axioms (Σ) does not influence termination
- Can be used to show soundness:
  - **Consistency:** there is no normal term of type $[\![\bot]\!]$
  - **Adequacy:** if $[\![\Gamma]\!] \vdash M : [\![A]\!]$ and M is a normal form, then $M = [N]$ for some N such that $\Gamma \vdash N : A$
  - **Conservativity:** if $[\![\Gamma]\!] \vdash M : [\![A]\!]$ then M reduces to a normal form $[N]$ for some N such that $\Gamma \vdash N : A$.

On termination and soundness

- Adding rewrite rules (R) can break strong normalization:
  - because $\longrightarrow_R$ does not terminate

On termination and soundness

- Adding rewrite rules (R) can break strong normalization:
  - because $\longrightarrow_R$ does not terminate
  - or because $\longrightarrow_R \cup \longrightarrow_\beta$ does not terminate

On termination and soundness

- Adding rewrite rules (R) can break strong normalization:
  - because $\longrightarrow_R$ does not terminate
  - or because $\longrightarrow_R \cup \longrightarrow_\beta$ does not terminate
  - or even because $\longrightarrow_\beta$ does not terminate for well-typed terms

On termination and soundness

- Adding rewrite rules (R) can break strong normalization:
  - because $\longrightarrow_R$ does not terminate
  - or because $\longrightarrow_R \cup \longrightarrow_\beta$ does not terminate
  - or even because $\longrightarrow_\beta$ does not terminate for well-typed terms
- Need to find other solutions.

Summary

- $\lambda\Pi$ embeddings do not preserve reduction.
- Obstacle for embedding theories with dependent types.
- Adding rewrite rules to $\lambda\Pi$ helps recover completeness...
- ... but can break soundness.

Approach 1: termination models

Idea: build a model for $\lambda\Pi/X$

- in the algebra of reducibility candidates
- or in a general notion of Π-algebra.

Approach 1: termination models

Idea: build a model for $\lambda\Pi/X$

- in the algebra of reducibility candidates
- or in a general notion of Π-algebra.

The model implies strong normalization of $\lambda\Pi/X$. Use this to prove soundness (consistency, adequacy, conservativity).

Approach 1: termination models

Idea: build a model for $\lambda\Pi/X$

- in the algebra of reducibility candidates
- or in a general notion of Π-algebra.

The model implies strong normalization of $\lambda\Pi/X$. Use this to prove soundness (consistency, adequacy, conservativity).

Theorem (Dowek 2014)

There is a model for $\lambda\Pi/\mathbb{STT}$ and for $\lambda\Pi/\mathbb{COC}$.

Approach 1: termination models

Idea: build a model for $\lambda\Pi/X$

- in the algebra of reducibility candidates
- or in a general notion of Π-algebra.

The model implies strong normalization of $\lambda\Pi/X$. Use this to prove soundness (consistency, adequacy, conservativity).

Theorem (Dowek 2014)

There is a model for $\lambda\Pi/\mathbb{STT}$ and for $\lambda\Pi/\mathbb{COC}$.

Problem: implies strong normalization in X, so at least as hard to prove as strong normalization in X.

Approach 2: relative normalization

If $[\![\Gamma]\!] \vdash M : [\![A]\!]$, what can we say about M?

Approach 2: relative normalization

If $\llbracket \Gamma \rrbracket \vdash M : \llbracket A \rrbracket$, what can we say about M?

Example

If X is the simply-typed λ-calculus, the polymorphic identity function is not well-typed:

$$\beta : \textbf{Type} \nvdash (\lambda \alpha : \textbf{Type}.\, \lambda x : \alpha.\, x)\, \beta : \beta \to \beta$$

It is well-typed in $\lambda\Pi/X$:

$$\beta : \mathsf{type} \vdash (\lambda \alpha : \mathsf{type}.\, \lambda x : \mathsf{term}\, \alpha.\, x)\, \beta : \mathsf{term}\, \beta \to \mathsf{term}\, \beta$$

Approach 2: relative normalization

If $[\![\Gamma]\!] \vdash M : [\![A]\!]$, what can we say about M?

Example

If X is the simply-typed λ-calculus, the polymorphic identity function is not well-typed:

$$\beta : \textbf{Type} \nvdash (\lambda\alpha : \textbf{Type}.\, \lambda x : \alpha.\, x)\, \beta : \beta \to \beta$$

It is well-typed in $\lambda\Pi/X$:

$$\beta : \mathsf{type} \vdash (\lambda\alpha : \mathsf{type}.\, \lambda x : \mathsf{term}\, \alpha.\, x)\, \beta : \mathsf{term}\, \beta \to \mathsf{term}\, \beta$$

But it reduces to $\lambda x : \mathsf{term}\, \beta.\, x = [\lambda x : \beta.\, x]$, a term that is well-typed in X.

Approach 2: relative normalization

If $[\![\Gamma]\!] \vdash M : [\![A]\!]$, what can we say about M?

Example

If X is the simply-typed λ-calculus, the polymorphic identity function is not well-typed:

$$\beta : \textbf{Type} \nvdash (\lambda\alpha : \textbf{Type}.\, \lambda x : \alpha.\, x)\, \beta : \beta \to \beta$$

It is well-typed in $\lambda\Pi/X$:

$$\beta : \mathsf{type} \vdash (\lambda\alpha : \mathsf{type}.\, \lambda x : \mathsf{term}\, \alpha.\, x)\, \beta : \mathsf{term}\, \beta \to \mathsf{term}\, \beta$$

But it reduces to $\lambda x : \mathsf{term}\, \beta.\, x = [\lambda x : \beta.\, x]$, a term that is well-typed in X.

Idea: reduce only what is necessary.

Erasure

Define an erasure from $\lambda\Pi/X$ to X:

$$\begin{array}{rcl} |x| & = & x \\ |\lambda x : A.\, M| & = & \lambda x : \|A\|.\, |M| \\ |M\, N| & = & |M|\, |N| \end{array}$$

$$\begin{array}{rcl} \|\mathsf{term}\, A\| & = & |A| \\ \|A \to B\| & = & \|A\| \to \|B\| \end{array}$$

Erasure is the inverse of the translation:

$$\begin{array}{rcl} |[M]| & = & M \\ \|[\![A]\!]\| & = & A \end{array}$$

Proving soundness

What statement should we prove?

- If $[\![\Gamma]\!] \vdash M : [\![A]\!]$ in $\lambda\Pi/X$ then $\Gamma \vdash |M| : A$ in X?

Proving soundness

What statement should we prove?

- ~~If $[\![\Gamma]\!] \vdash M : [\![A]\!]$ in $\lambda\Pi/X$ then $\Gamma \vdash |M| : A$ in X?~~

  $\beta : \mathsf{type} \vdash (\lambda\alpha : \mathsf{type}.\, \lambda x : \mathsf{term}\, \alpha.\, x)\, \beta : \mathsf{term}\, \beta \to \mathsf{term}\, \beta$

Proving soundness

What statement should we prove?

- $\cancel{\text{If } [\![\Gamma]\!] \vdash M : [\![A]\!] \text{ in } \lambda\Pi/X \text{ then } \Gamma \vdash |M| : A \text{ in } X?}$

  $\beta : \mathsf{type} \vdash (\lambda\alpha : \mathsf{type}.\, \lambda x : \mathsf{term}\, \alpha.\, x)\, \beta : \mathsf{term}\, \beta \to \mathsf{term}\, \beta$

- If $[\![\Gamma]\!] \vdash M : [\![A]\!]$ in $\lambda\Pi/X$ then $M \longrightarrow^* M'$ such that $\Gamma \vdash |M'| : A$ in X?

Proving soundness

What statement should we prove?

- ~~If $[\![\Gamma]\!] \vdash M : [\![A]\!]$ in $\lambda\Pi/X$ then $\Gamma \vdash |M| : A$ in X?~~
  $\beta : \mathsf{type} \vdash (\lambda\alpha : \mathsf{type}.\, \lambda x : \mathsf{term}\, \alpha.\, x)\, \beta : \mathsf{term}\, \beta \to \mathsf{term}\, \beta$

- ~~If $[\![\Gamma]\!] \vdash M : [\![A]\!]$ in $\lambda\Pi/X$ then $M \longrightarrow^* M'$ such that $\Gamma \vdash |M'| : A$ in X?~~

$$\frac{[\![\Gamma]\!] \vdash M : \Pi x : {\color{red}A}.\, [\![B]\!] \qquad [\![\Gamma]\!] \vdash N : {\color{red}A}}{[\![\Gamma]\!] \vdash M\, N : [\![B]\!]}$$

Proving soundness

What statement should we prove?

- If $\llbracket \Gamma \rrbracket \vdash M : A$ in $\lambda\Pi/X$ then $M \longrightarrow^* M'$ and $A \longrightarrow^* A'$ such that $\Gamma \vdash |M'| : \|A'\|$ in X?

Proving soundness

What statement should we prove?

- ~~If $[\![\Gamma]\!] \vdash M : A$ in $\lambda\Pi/X$ then $M \longrightarrow^* M'$ and $A \longrightarrow^* A'$ such that $\Gamma \vdash |M'| : \|A'\|$ in X?~~

$$\frac{[\![\Gamma]\!], x : {\color{red}A} \vdash M : B}{[\![\Gamma]\!] \vdash \lambda x : A.\, M : \Pi x : {\color{red}A}.\, B}$$

Proving soundness

What statement should we prove?

- ~~If $\llbracket\Gamma\rrbracket \vdash M : A$ in $\lambda\Pi/X$ then $M \longrightarrow^* M'$ and $A \longrightarrow^* A'$ such that $\Gamma \vdash |M'| : \|A'\|$ in X?~~

$$\frac{\llbracket\Gamma\rrbracket, x : \color{red}{A} \vdash M : B}{\llbracket\Gamma\rrbracket \vdash \lambda x : A.\, M : \Pi x : \color{red}{A}.\, B}$$

- If $\Gamma \vdash M : A$ in $\lambda\Pi/X$ then $\Gamma \longrightarrow^* \Gamma'$, $M \longrightarrow^* M'$, and $A \longrightarrow^* A'$ such that $\|\Gamma'\| \vdash |M'| : \|A'\|$ in X?

Proving soundness

What statement should we prove?

- ~~If $\llbracket\Gamma\rrbracket \vdash M : A$ in $\lambda\Pi/X$ then $M \longrightarrow^* M'$ and $A \longrightarrow^* A'$ such that $\Gamma \vdash |M'| : \|A'\|$ in X?~~

$$\frac{\llbracket\Gamma\rrbracket, x : {\color{red}A} \vdash M : B}{\llbracket\Gamma\rrbracket \vdash \lambda x : A.\, M : \Pi x : {\color{red}A}.\, B}$$

- ~~If $\Gamma \vdash M : A$ in $\lambda\Pi/X$ then $\Gamma \longrightarrow^* \Gamma'$, $M \longrightarrow^* M'$, and $A \longrightarrow^* A'$ such that $\|\Gamma'\| \vdash |M'| : \|A'\|$ in X?~~

$$\vdash \lambda\alpha : \mathrm{type}.\, \lambda x : \mathrm{term}\,\alpha.\, x : \Pi\alpha : .\,\mathrm{term}\,\beta \to \mathrm{term}\,\beta$$

Proving soundness

What have we learned?

1. $\lambda\Pi/X$ can type more terms than X.
2. These terms can be used to construct proofs for the translation of X types.
3. The $\lambda\Pi/X$ terms that inhabit the translation of X types can be reduced to the translation of X terms.

Proving soundness

What have we learned?

1. $\lambda\Pi/X$ can type more terms than X.
2. These terms can be used to construct proofs for the translation of X types.
3. The $\lambda\Pi/X$ terms that inhabit the translation of X types can be reduced to the translation of X terms.

Need higher-order reasonning.

Reducibility

Let Γ' be a context in X. Define the predicate $\Gamma' \vDash M : A$ by induction on A:

Reducibility

Let Γ' be a context in X. Define the predicate $\Gamma' \vDash M : A$ by induction on A:

- If $A = \text{type}$ then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ such that $\Gamma' \vdash |M'| : \mathbf{Type}$.

Reducibility

Let Γ' be a context in X. Define the predicate $\Gamma' \vDash M : A$ by induction on A:

- If $A =$ type then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ such that $\Gamma' \vdash |M'| : \textbf{Type}$.

- If $A =$ term B then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ and $B \longrightarrow^* B'$ such that $\Gamma' \vdash |M'| : |B'|$.

Reducibility

Let Γ' be a context in X. Define the predicate $\Gamma' \vDash M : A$ by induction on A:

- If $A = \text{type}$ then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ such that $\Gamma' \vdash |M'| : \mathbf{Type}$.
- If $A = \text{term}\, B$ then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ and $B \longrightarrow^* B'$ such that $\Gamma' \vdash |M'| : |B'|$.
- If $A = \Pi x : B.\, C$ then $\Gamma' \vDash M : A$ when for for all N such that $\Gamma' \vDash N : B$, $\Gamma' \vDash M\, N : \{N/x\}\, C$.

Reducibility

Let Γ' be a context in X. Define the predicate $\Gamma' \vDash M : A$ by induction on A:

- If $A = \text{type}$ then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ such that $\Gamma' \vdash |M'| : \textbf{Type}$.

- If $A = \text{term}\, B$ then $\Gamma' \vDash M : A$ when $M \longrightarrow^* M'$ and $B \longrightarrow^* B'$ such that $\Gamma' \vdash |M'| : |B'|$.

- If $A = \Pi x : B.\, C$ then $\Gamma' \vDash M : A$ when for for all N such that $\Gamma' \vDash N : B$, $\Gamma' \vDash M\, N : \{N/x\}\, C$.

If σ is a substitution mapping variables to terms:

- $\Gamma' \vDash \sigma : \Gamma$ when $\Gamma' \vDash \sigma\,(x) : \sigma\,(A)$ for all $(x : A) \in \Gamma$

Soundness

Theorem

If $\Gamma \vdash M : A$ *in* $\lambda\Pi/X$ *then for any* X *context* Γ' *and substitution* σ *such that* $\Gamma' \vDash \sigma : \Gamma$, $\Gamma' \vDash \sigma(M) : \sigma(A)$.

Proof.

By induction on the derivation of $\Gamma \vdash M : A$. $\square$

Soundness

Theorem

If $\Gamma \vdash M : A$ *in* $\lambda\Pi/X$ *then for any* X *context* Γ' *and substitution* σ *such that* $\Gamma' \vDash \sigma : \Gamma$, $\Gamma' \vDash \sigma(M) : \sigma(A)$.

Proof.

By induction on the derivation of $\Gamma \vdash M : A$. □

Corollary (Conservativity)

If $[\![\Gamma]\!] \vdash M : [\![A]\!]$ *then* $M \longrightarrow^* M'$ *such that* $\Gamma \vdash |M'| : A$.

Proof.

By taking the identity substitution, $\|\sigma([\![A]\!])\| = \|[\![A]\!]\| = A$. □

Relative normalization

- Avoid complex techniques such as reducibility candidates.
- Works for non-terminating theories!
- For pure type systems, $\lambda\Pi/X$ corresponds to a conservative completion of X.

Summary

- **Strong normalization** = all terms in $\lambda\Pi/X$ are strongly normalizing
  - Proved using termination models
- **Relative normalization** = terms in $\lambda\Pi/X$ can be reduced to terms in X.
  - Proved by using reducibility on a more general statement
- Both approaches show conservativity of $\lambda\Pi/X$

Conclusion

- The $\lambda\Pi$-calculus modulo rewriting can be used as a logical framework
- We use it for logical embeddings that preserve reduction
- Soundness needs to be handled carefully through models or reducibility techniques

Conclusion

Questions?