Conservativity Made Easy

Thiago Felicissimo

18 July 2026 @ Tribute to Gilles Dowek

INRIA

Based on FSCD'22, extended by my PhD thesis

Thank you Gilles!

Back to the beginning

2007 Seminal paper by Gilles and Denis Cousineau:

Embedding Pure Type Systems in the Lambda-Pi-Calculus Modulo

Denis Cousineau and Gilles Dowek

École polytechnique and INRIA
LIX, École polytechnique, 91128 Palaiseau Cedex, France
Cousineau@lix.polytechnique.fr
http://www.lix.polytechnique.fr/~cousineau
Gilles.Dowek@polytechnique.edu
http://www.lix.polytechnique.fr/~dowek

Abstract. The lambda-Pi-calculus allows to express proofs of minimal predicate logic. It can be extended, in a very simple way, by adding computation rules. This leads to the lambda-Pi-calculus modulo. We show in this paper that this simple extension is surprisingly expressive and, in particular, that all functional Pure Type Systems, such as the system F, or the Calculus of Constructions, can be embedded in it. And, moreover, that this embedding is conservative under termination hypothesis.

Introduction of $\lambda\Pi$-modulo (Dedukti) and encoding of (functional) Pure Type Systems (PTSs)

Established the foundation for most future Dedukti encodings

What does “correct encoding” mean?

Let $\mathscr{S}$ a functional PTS spec, $\mathbb{T}_{\mathscr{S}}$ theory in Dedukti, and $[\![-]\!]$ translation function

What does “correct encoding” mean?

Let $\mathscr{S}$ a functional PTS spec, $\mathbb{T}_{\mathscr{S}}$ theory in Dedukti, and $[\![-]\!]$ translation function

Soundness: If $\Gamma \vdash_{\mathscr{S}} t : A$ then $\mathbb{T}_{\mathscr{S}} \rhd [\![\Gamma]\!] \vdash_{\mathsf{dk}} [\![t]\!] : \mathrm{Tm}\ [\![A]\!]$

What does "correct encoding" mean?

Let $\mathscr{S}$ a functional PTS spec, $\mathbb{T}_{\mathscr{S}}$ theory in Dedukti, and $[\![-]\!]$ translation function

Soundness: If $\Gamma \vdash_{\mathscr{S}} t : A$ then $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} [\![t]\!] : \mathrm{Tm}\ [\![A]\!]$

Easy, by induction on derivation

What does "correct encoding" mean?

Let $\mathscr{S}$ a functional PTS spec, $\mathbb{T}_{\mathscr{S}}$ theory in Dedukti, and $[\![-]\!]$ translation function

Soundness: If $\Gamma \vdash_{\mathscr{S}} t : A$ then $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} [\![t]\!] : \mathrm{Tm}\ [\![A]\!]$

Easy, by induction on derivation

Conservativity: If $\Gamma \vdash_{\mathscr{S}} A\ \mathsf{type}$ and $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ then $\Gamma \vdash_{\mathscr{S}} t' : A$ for some t'

What does “correct encoding” mean?

Let $\mathscr{S}$ a functional PTS spec, $\mathbb{T}_{\mathscr{S}}$ theory in Dedukti, and $[\![-]\!]$ translation function

Soundness: If $\Gamma \vdash_{\mathscr{S}} t : A$ then $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} [\![t]\!] : \mathrm{Tm}\ [\![A]\!]$

Easy, by induction on derivation

Conservativity: If $\Gamma \vdash_{\mathscr{S}} A$ **type** and $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ then $\Gamma \vdash_{\mathscr{S}} t' : A$ for some t'

<u>Hard</u>, not all Dedukti terms t of type $\mathrm{Tm}\ [\![A]\!]$ correspond to some PTS term $t' : A$ in $\mathscr{S}$

What does "correct encoding" mean?

Let $\mathscr{S}$ a functional PTS spec, $\mathbb{T}_{\mathscr{S}}$ theory in Dedukti, and $\llbracket - \rrbracket$ translation function

Soundness: If $\Gamma \vdash_{\mathscr{S}} t : A$ then $\mathbb{T}_{\mathscr{S}} \triangleright \llbracket \Gamma \rrbracket \vdash_{\mathsf{dk}} \llbracket t \rrbracket : \mathrm{Tm}\ \llbracket A \rrbracket$

Easy, by induction on derivation

Conservativity: If $\Gamma \vdash_{\mathscr{S}} A\ \mathsf{type}$ and $\mathbb{T}_{\mathscr{S}} \triangleright \llbracket \Gamma \rrbracket \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ \llbracket A \rrbracket$ then $\Gamma \vdash_{\mathscr{S}} t' : A$ for some t'

Hard, not all Dedukti terms t of type $\mathrm{Tm}\ \llbracket A \rrbracket$ correspond to some PTS term $t' : A$ in $\mathscr{S}$

Fortunately, correspondence can usually be established for Dedukti β-normal forms

Consequence Conservativity result of Gilles and Denis assumes **normalization** of $\mathbb{T}_{\mathscr{S}}$

Addressing the normalization assumption

Subsequent works provided ways of addressing this additional assumption:

Addressing the normalization assumption

Subsequent works provided ways of addressing this additional assumption:

2017 Gilles provided a notion of normalization model for Dedukti, applied to establish conservativity for encodings of HOL and CoC

Addressing the normalization assumption

Subsequent works provided ways of addressing this additional assumption:

2017 Gilles provided a notion of normalization model for Dedukti, applied to establish conservativity for encodings of HOL and CoC

2014 Using a logical relations technique, Ali Assaf proved the 2007 PTS encoding conservative *without* the normalization assumption

Addressing the normalization assumption

Subsequent works provided ways of addressing this additional assumption:

2017 Gilles provided a notion of normalization model for Dedukti, applied to establish conservativity for encodings of HOL and CoC

2014 Using a logical relations technique, Ali Assaf proved the 2007 PTS encoding conservative *without* the normalization assumption

Problem Both approaches rely on quite technical arguments and thus lacked adoption

As a consequence, most works simply leave conservativity as a conjecture...

The root of the problem

The ELF strategy In the Edinburgh LF (ELF) β-reduction is always normalizing, therefore conservativity proofs can consider β-normal forms without loss of generality

The root of the problem

The ELF strategy In the Edinburgh LF (ELF) β-reduction is always normalizing, therefore conservativity proofs can consider β-normal forms without loss of generality

Problem Adding rewrite rules in Dedukti can break normalization of β

The root of the problem

The ELF strategy In the Edinburgh LF (ELF) β-reduction is always normalizing, therefore conservativity proofs can consider β-normal forms without loss of generality

Problem Adding rewrite rules in Dedukti can break normalization of β

Classic example Gilles and Denis' rule for encoding dependent functions

$$\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$$

Implements functions of encoded theory as functions of Dedukti, β translated as β

Therefore, encoding of non-terminating theory yields non-termination in Dedukti

The root of the problem

The ELF strategy In the Edinburgh LF (ELF) β-reduction is always normalizing, therefore conservativity proofs can consider β-normal forms without loss of generality

Problem Adding rewrite rules in Dedukti can break normalization of β

Classic example Gilles and Denis' rule for encoding dependent functions

$$\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$$

Implements functions of encoded theory as functions of Dedukti, β translated as β

Therefore, encoding of non-terminating theory yields non-termination in Dedukti

Idea Find a "safe" fragment of rewrite rules that does not jeopardize termination of β

A criterion for normalization of β in Dedukti

We say a rewrite rule is *arrow-producing* when its RHS contains some $\to$

A criterion for normalization of β in Dedukti

We say a rewrite rule is *arrow-producing* when its RHS contains some $\to$

Theorem If no rewrite rules are arrow-producing and $(\dots)$, then β is normalizing in Dedukti

A criterion for normalization of β in Dedukti

We say a rewrite rule is *arrow-producing* when its RHS contains some $\to$

Theorem If no rewrite rules are arrow-producing and $(\dots)$, then β is normalizing in Dedukti

Proof We adapt the well-known *dependency-erasure* technique to Dedukti

Concretely, we define translation to the simply-typed λ-calculus that erases type dependency

It preserves β-reduction sequences, so normalization of β in Dedukti follows

A criterion for normalization of β in Dedukti

We say a rewrite rule is *arrow-producing* when its RHS contains some $\to$

Theorem If no rewrite rules are arrow-producing and $(\dots)$, then β is normalizing in Dedukti

Proof We adapt the well-known *dependency-erasure* technique to Dedukti

Concretely, we define translation to the simply-typed λ-calculus that erases type dependency

It preserves β-reduction sequences, so normalization of β in Dedukti follows

Key point To handle conversion rule, we need erasure of types to be invariant under rewriting

$$\begin{array}{rcc}
\mathrm{Tm}\ \mathrm{U} \longmapsto \mathrm{Ty} & \overset{\checkmark}{\leadsto} & |\mathrm{Tm}\ \mathrm{U}| = \iota \quad = \quad \iota = |\mathrm{Ty}| \\
\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x) & \overset{\text{✗}}{\leadsto} & |\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B})| = \iota \quad \neq \quad \iota \to \iota = |(x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)|
\end{array}$$

Revising the Dedukti methodology

Equipped with the presented criterion, I propose the following:

Revising the Dedukti methodology

Equipped with the presented criterion, I propose the following:

Revised Dedukti Methodology Ban arrow-producing rules in encodings, then by given criterion it suffices to prove conservativity for β-normal forms, as in the ELF approach

Revising the Dedukti methodology

Equipped with the presented criterion, I propose the following:

Revised Dedukti Methodology Ban arrow-producing rules in encodings, then by given criterion it suffices to prove conservativity for β-normal forms, as in the ELF approach

Methodology illustrated with a new encoding of PTSs, with easy conservativity proof

Arrow-producing rule $\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$ replaced by

$$\lambda : ((x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)) \to \mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B})$$

$$@ : \mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \to (\mathtt{u} : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ \mathtt{u})$$

$$@\ (\lambda\ \mathtt{t})\ \mathtt{u} \overset{\boldsymbol{\beta}}{\longmapsto} \mathtt{t}\ \mathtt{u}$$

Revising the Dedukti methodology

Equipped with the presented criterion, I propose the following:

Revised Dedukti Methodology Ban arrow-producing rules in encodings, then by given criterion it suffices to prove conservativity for β-normal forms, as in the ELF approach

Methodology illustrated with a new encoding of PTSs, with easy conservativity proof

Arrow-producing rule $\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$ replaced by

$$\lambda : ((x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)) \to \mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B})$$

$$@ : \mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \to (\mathtt{u} : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ \mathtt{u})$$

$$@\ (\lambda\ \mathtt{t})\ \mathtt{u} \overset{\boldsymbol{\beta}}{\longmapsto} \mathtt{t}\ \mathtt{u}$$

Remark For non-terminating PTS, while $\beta + \boldsymbol{\beta}$ is non-normalizing, β still normalizes

A last technical annoyance

Problem Not all β-normal t with $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ correspond to some PTS term

The culprit: partially applied subterms, lack of η-expansion

A last technical annoyance

Problem Not all β-normal t with $\mathbb{T}_{\mathscr{S}} \rhd [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ correspond to some PTS term

The culprit: partially applied subterms, lack of η-expansion

Some solutions:

A last technical annoyance

Problem Not all β-normal t with $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ correspond to some PTS term

The culprit: partially applied subterms, lack of η-expansion

Some solutions:

Gilles and Denis 2007 First prove that terms can be η-expanded (not always true in Dedukti)

A last technical annoyance

Problem Not all β-normal t with $\mathbb{T}_{\mathscr{S}} \triangleright \llbracket \Gamma \rrbracket \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ \llbracket A \rrbracket$ correspond to some PTS term

The culprit: partially applied subterms, lack of η-expansion

Some solutions:

Gilles and Denis 2007 First prove that terms can be η-expanded (not always true in Dedukti)

My FSCD'22 paper Consider symbols with *fixed arities*, quite inelegant solution

A last technical annoyance

Problem Not all β-normal t with $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ correspond to some PTS term

The culprit: partially applied subterms, lack of η-expansion

Some solutions:

Gilles and Denis 2007 First prove that terms can be η-expanded (not always true in Dedukti)

My FSCD'22 paper Consider symbols with *fixed arities*, quite inelegant solution

My PhD thesis Back-translation function $|-|$ that η-expands *at back-translation time*

If B in $|\Pi\ A\ B|$ is not a λ, we want $|\Pi\ A\ B| = |\Pi\ A\ (y.B\ y)|$, so we *define*

$$|\Pi\ A\ B| := \Pi y : |A|.|B\ y|$$

A last technical annoyance

Problem Not all β-normal t with $\mathbb{T}_{\mathscr{S}} \triangleright [\![\Gamma]\!] \vdash_{\mathsf{dk}} t : \mathrm{Tm}\ [\![A]\!]$ correspond to some PTS term

The culprit: partially applied subterms, lack of η-expansion

Some solutions:

Gilles and Denis 2007 First prove that terms can be η-expanded (not always true in Dedukti)

My FSCD'22 paper Consider symbols with *fixed arities*, quite inelegant solution

My PhD thesis Back-translation function $|-|$ that η-expands *at back-translation time*

If B in $|\Pi\ A\ B|$ is not a λ, we want $|\Pi\ A\ B| = |\Pi\ A\ (y.B\ y)|$, so we *define*

$$|\Pi\ A\ B| := \Pi y : |A|.|B\ y|$$

If B was already a λ, we've just created a β_0 redex, and so $|-|$ must also eliminate those

Conclusion

Takeaway message Conservativity can be made easy, by avoiding arrow-producing rules

Problems related to partially applied terms can also be solved without much difficulty

Conclusion

Takeaway message Conservativity can be made easy, by avoiding arrow-producing rules

Problems related to partially applied terms can also be solved without much difficulty

Problem Replacing $\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$ by explicit symbols $@, \lambda$ and rule $\boldsymbol{\beta}$ leads to terms with more annotation

$$(\lambda x.t)u \quad \rightsquigarrow \quad @\ [\![A]\!]\ (x.[\![B]\!])\ (\lambda\ [\![A]\!]\ (x.[\![B]\!])\ (x.[\![t]\!]))\ [\![u]\!]$$

Conclusion

Takeaway message Conservativity can be made easy, by avoiding arrow-producing rules

Problems related to partially applied terms can also be solved without much difficulty

Problem Replacing $\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$ by explicit symbols $@, \lambda$ and rule $\boldsymbol{\beta}$ leads to terms with more annotation

$$(\lambda x.t)u \quad \rightsquigarrow \quad @\ [\![A]\!]\ (x.[\![B]\!])\ (\lambda\ [\![A]\!]\ (x.[\![B]\!])\ (x.[\![t]\!]))\ [\![u]\!]$$

From my perspective, main obstacle for the adoption of this approach

My proposal of a Dedukti-like LF integrating bidirectional typing addresses this problem

Conclusion

Takeaway message Conservativity can be made easy, by avoiding arrow-producing rules

Problems related to partially applied terms can also be solved without much difficulty

Problem Replacing $\mathrm{Tm}\ (\Pi\ \mathtt{A}\ \mathtt{B}) \longmapsto (x : \mathrm{Tm}\ \mathtt{A}) \to \mathrm{Tm}\ (\mathtt{B}\ x)$ by explicit symbols $@, \lambda$ and rule $\boldsymbol{\beta}$ leads to terms with more annotation

$$(\lambda x.t)u \quad \rightsquigarrow \quad @\ [\![A]\!]\ (x.[\![B]\!])\ (\lambda\ [\![A]\!]\ (x.[\![B]\!])\ (x.[\![t]\!]))\ [\![u]\!]$$

From my perspective, main obstacle for the adoption of this approach

My proposal of a Dedukti-like LF integrating bidirectional typing addresses this problem

Thank you for your attention, and thank you Gilles!