A FRAMEWORK FOR DEFINING COMPUTATIONAL HIGHER-ORDER LOGICS

Ali Assaf

September 28, 2015

École Polytechnique & Inria Paris

4-color theorem

4-color theorem

Coq

4-color theorem

Kepler conjecture

Coq

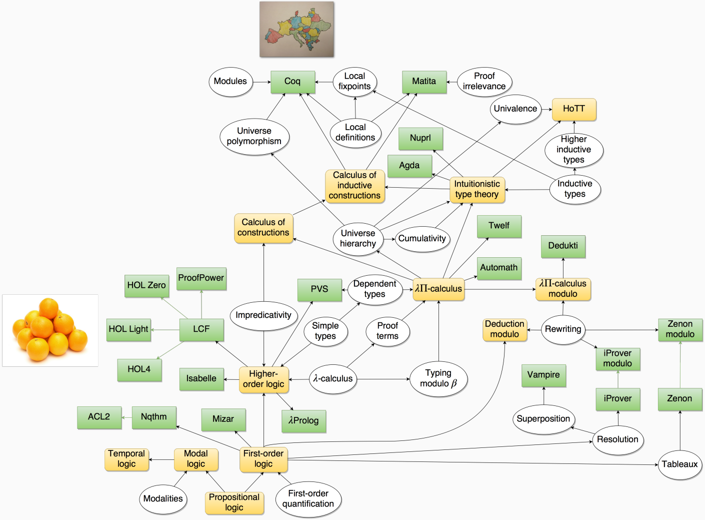
4-color theorem

Coq

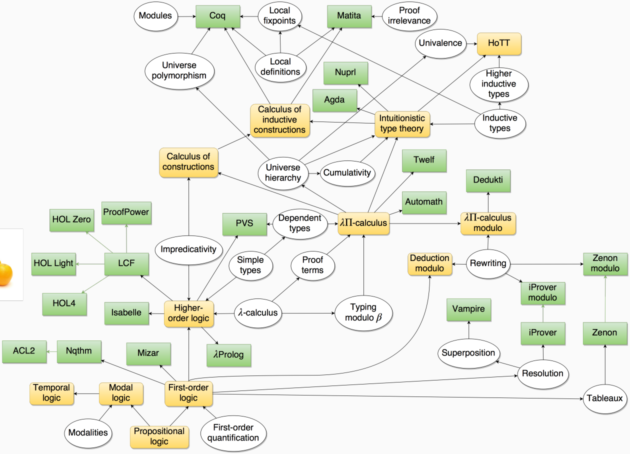
Kepler conjecture

HOL Light

PROOF SYSTEMS

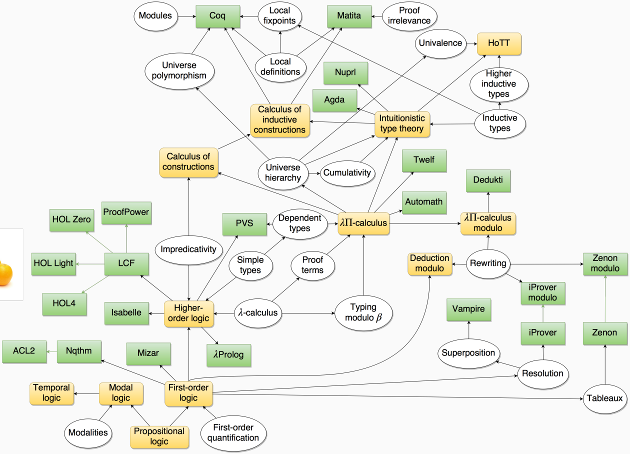


PROOF SYSTEMS



Independent proof checking?

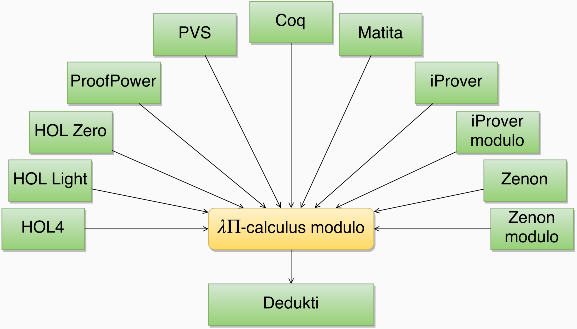
PROOF SYSTEMS



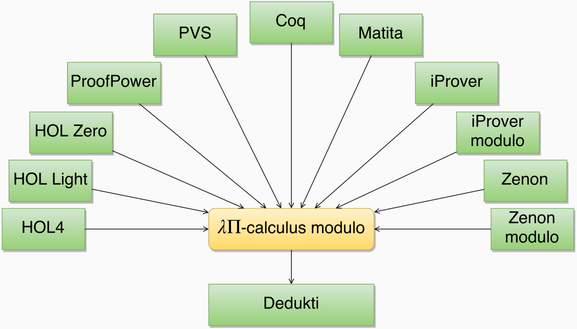
Independent proof checking?

Proof interoperability?

LOGICAL FRAMEWORK



LOGICAL FRAMEWORK



- Is it **trustworthy**?
- Is it **expressive**?
- Is it **efficient**?

$$\Gamma \vdash_{\mathcal{L}} A \iff \Sigma_{\mathcal{L}}, [\![\Gamma]\!] \vdash [\![A]\!]$$

- First-order logic (FOL) [Hilbert and Ackermann 1928]
  - $\Sigma = f : n_f, \ldots, p : n_p, \ldots, A, \ldots$

$$\Gamma \vdash_{\mathcal{L}} A \iff \Sigma_{\mathcal{L}}, [\![\Gamma]\!] \vdash [\![A]\!]$$

- **First-order logic** (FOL) [Hilbert and Ackermann 1928]
- **Higher-order logic** (HOL) [Church 1940]
  - Binders using **simply-typed λ-calculus**
  - $\Sigma = f : \tau, \ldots, A, \ldots$
  - Used in ISABELLE, λPROLOG

$$\Gamma \vdash_{\mathcal{L}} A \iff \Sigma_{\mathcal{L}}, [\![\Gamma]\!] \vdash M : [\![A]\!]$$

- **First-order logic** (FOL) [Hilbert and Ackermann 1928]
- **Higher-order logic** (HOL) [Church 1940]
- **$\lambda\Pi$-calculus** (LF, $\lambda\Pi$, λP) [Harper et al. 1987]
    - Proof terms using **dependently-typed λ-calculus**
    - $\Sigma = f : A, \ldots$
    - Used in AUTOMATH, TWELF

$$\Gamma \vdash_{\mathcal{L}} A \iff \Sigma_{\mathcal{L}}, [\![\Gamma]\!] \vdash M : [\![A]\!]$$

- **First-order logic** (FOL) [Hilbert and Ackermann 1928]
- **Higher-order logic** (HOL) [Church 1940]
- $\lambda\Pi$**-calculus** (LF, $\lambda\Pi$, λP) [Harper et al. 1987]
- $\lambda\Pi$**-calculus modulo** ($\lambda\Pi R$) [Cousineau and Dowek 2007]
    - Computation using **rewriting**
    - $\Sigma = f : A, \ldots, f\vec{M} \mapsto N, \ldots$
    - Used in DEDUKTI

THE $\lambda\Pi$-CALCULUS MODULO REWRITING

λ-calculus + dependent types + rewriting

$$x \mid \lambda x^A . M \mid M\, N \mid \mathrm{Type} \mid \Pi x^A . B$$

λ-calculus + dependent types + rewriting

$$x \mid \lambda x^A . M \mid M\, N \mid \mathrm{Type} \mid \Pi x^A . B$$

Evaluation: β-reduction and rewrite rules

Example

$$\mathcal{R} = \begin{cases} x + 0 & \longmapsto & x \\ x + \mathsf{S}\,(y) & \longmapsto & \mathsf{S}\,(x) + y \end{cases}$$

λ-calculus + dependent types + rewriting

$$x \mid \lambda x^A . M \mid M\, N \mid \mathrm{Type} \mid \Pi x^A . B$$

Evaluation: β-reduction and rewrite rules

Example

$$\mathcal{R} = \begin{cases} x + 0 & \longmapsto \quad x \\ x + \mathrm{S}\,(y) & \longmapsto \quad \mathrm{S}\,(x) + y \end{cases}$$

Typing: modulo β-reduction and rewrite rules

$$\frac{\Gamma \vdash M : A \qquad \dots \qquad A \equiv_{\beta\mathcal{R}} B}{\Gamma \vdash M : B}\ \textsc{Conv}$$

Smaller encodings

$$\dfrac{\dfrac{}{2+2=3+1} \qquad \dfrac{\dfrac{}{3+1=4+0} \qquad \dfrac{}{4+0=4}}{3+1=4}}{2+2=4} \qquad \text{vs} \qquad \dfrac{\dfrac{}{4=4}}{2+2=4}$$

Smaller encodings

$$\dfrac{\dfrac{}{2+2=3+1} \qquad \dfrac{\dfrac{}{3+1=4+0} \qquad \dfrac{}{4+0=4}}{3+1=4}}{2+2=4} \qquad \text{vs} \qquad \dfrac{\dfrac{}{4=4}}{2+2=4}$$

Encodings of **more powerful** theories

- Pure type systems [Cousineau and Dowek 2007]

$$\Gamma \vdash_{\mathcal{P}} M : A \implies \Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} [M] : [\![A]\!]$$

Calculus of constructions (CC), higher-order logic (HOL), …

Smaller encodings

$$\dfrac{\dfrac{}{2+2=3+1} \qquad \dfrac{\dfrac{}{3+1=4+0} \qquad \dfrac{}{4+0=4}}{3+1=4}}{2+2=4} \qquad \text{vs} \qquad \dfrac{\dfrac{}{4=4}}{2+2=4}$$

Encodings of **more powerful** theories

- Pure type systems [Cousineau and Dowek 2007]

  $$\Gamma \vdash_{\mathcal{P}} M : A \implies \Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} [M] : [\![A]\!]$$

  Calculus of constructions (CC), higher-order logic (HOL), ...
- Inductive types [Boespflug and Burel 2012]

Smaller encodings

$$\dfrac{\dfrac{}{2+2=3+1} \quad \dfrac{\dfrac{}{3+1=4+0} \quad \dfrac{}{4+0=4}}{3+1=4}}{2+2=4} \qquad \text{vs} \qquad \dfrac{\dfrac{}{4=4}}{2+2=4}$$

Encodings of **more powerful** theories

- Pure type systems [Cousineau and Dowek 2007]

  $$\Gamma \vdash_{\mathcal{P}} M : A \implies \Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} [M] : [\![A]\!]$$

  Calculus of constructions (CC), higher-order logic (HOL), ...

- Inductive types [Boespflug and Burel 2012]

HOL and Coq?

PROBLEM I

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

This is not enough!

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

This is not enough!

$$\exists M.\, \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\, \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]$$

PROBLEM I

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

$$\begin{aligned} [\![A]\!] &= \top \\ [M] &= \top\text{-intro} \end{aligned}$$

$$\exists M.\, \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\, \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]$$

$$\exists M.\, \Gamma \vdash M : A \implies \Sigma, [\![\Gamma]\!] \vdash \top\text{-intro} : \top \quad \checkmark$$

$$\begin{aligned} [\![A]\!] &= \top \\ [M] &= \top\text{-intro} \end{aligned}$$

$$\exists M.\, \Gamma \vdash M : A \impliedby \exists M'.\, \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]$$

PROBLEM I

$$\exists M.\, \Gamma \vdash M : A \quad \Longrightarrow \quad \Sigma, [\![\Gamma]\!] \vdash \top\text{-intro} : \top \quad ✓$$

$$[\![A]\!] \quad = \quad \top$$

$$[M] \quad = \quad \top\text{-intro}$$

$$\exists M.\, \Gamma \vdash M : A \quad \Longleftarrow \quad \Sigma, [\![\Gamma]\!] \vdash \top\text{-intro} : \top \quad ✗$$

Cumulative universes

- Intuitionistic type theory (ITT)
- Calculus of inductive constructions (CIC)

$$\frac{}{\Gamma \vdash \mathsf{Type}_i : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \in \mathsf{Type}_2 \in \mathsf{Type}_3 \in \cdots$$

Cumulative universes

- Intuitionistic type theory (ITT)
- Calculus of inductive constructions (CIC)

$$\frac{}{\Gamma \vdash \mathsf{Type}_i : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \in \mathsf{Type}_2 \in \mathsf{Type}_3 \in \cdots$$

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \subseteq \mathsf{Type}_2 \subseteq \mathsf{Type}_3 \subseteq \cdots$$

- **Prove** that the embedding is **conservative**

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

- **Extend** the embedding to **cumulative systems**

$$\mathsf{Type}_0 \subseteq \mathsf{Type}_1 \subseteq \mathsf{Type}_2 \subseteq \cdots$$

- **Implement** the translation of the proofs of **HOL**, **Coq**, and **Matita** into Dedukti

EMBEDDING PURE TYPE SYSTEMS

$$x \mid \lambda x^A . M \mid M\, N \mid s \mid \Pi x^A . B$$

PURE TYPE SYSTEMS

$$x \mid \lambda x^A \,.\, M \mid M\,N \mid s \mid \Pi x^A \,.\, B$$

Typing: parameterized by a specification $(\mathcal{S}, \mathcal{A}, \mathcal{R})$

- $\mathcal{S}$ set of *sorts* (a.k.a. *universes*)
- $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$
- $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$

$$\frac{(s_1, s_2) \in \mathcal{A}}{\vdash s_1 : s_2} \qquad \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathcal{R}}{\Gamma \vdash \Pi x^A \,.\, B : s_3}$$

$$x \mid \lambda x^A . M \mid M\, N \mid s \mid \Pi x^A . B$$

Typing: parameterized by a specification $(\mathcal{S}, \mathcal{A}, \mathcal{R})$

- $\mathcal{S}$ set of *sorts* (a.k.a. *universes*)
- $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$
- $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$

$$\frac{(s_1, s_2) \in \mathcal{A}}{\vdash s_1 : s_2} \qquad \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathcal{R}}{\Gamma \vdash \Pi x^A . B : s_3}$$

Example (Induction principle in the calculus of constructions)

$$\Pi p^{(\mathbb{N} \to \mathsf{Prop})} . p\, 0 \to \left(\Pi n^{\mathbb{N}} . p\, n \to p\, (\mathsf{S}\, n)\right) \to \Pi n^{\mathbb{N}} . p\, n$$

LIMITATIONS OF $\lambda\Pi$

Traditional embeddings in $\lambda\Pi$ do not preserve reduction:

$$M \longrightarrow M' \quad \not\Rightarrow \quad [M] \longrightarrow [M']$$

Traditional embeddings in $\lambda\Pi$ do not preserve reduction:

$$M \longrightarrow M' \quad \not\Longrightarrow \quad [M] \longrightarrow [M']$$

Example

$$\left[(\lambda x^{C} . x)\, y\right] \;=\; \mathrm{app}\,[C]\;[C]\;\left(\mathrm{lam}\,[C]\;[C]\;(\lambda x^{[\![C]\!]} . x)\right)\,y \;\not\longrightarrow\; y$$

LIMITATIONS OF $\lambda\Pi$

Traditional embeddings in $\lambda\Pi$ do not preserve equivalence:

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

Example

$$\left[(\lambda x^C . x)\, y\right] \;=\; \mathrm{app}\,[C]\;[C]\;\left(\mathrm{lam}\,[C]\;[C]\;(\lambda x^{[\![C]\!]} . x)\right)\; y \;\not\rightarrow\; y$$

LIMITATIONS OF $\lambda\Pi$

Traditional embeddings in $\lambda\Pi$ do not preserve equivalence:

$$M \equiv M' \quad \not\Longrightarrow \quad [M] \equiv [M']$$

Example

$$\left[(\lambda x^{C} . x)\, y\right] \;=\; \mathsf{app}\,[C]\;[C]\;\left(\mathsf{lam}\,[C]\;[C]\;(\lambda x^{[\![C]\!]} . x)\right)\; y \;\not\longrightarrow\; y$$

This is a problem for computational systems:

$$\frac{\Gamma \vdash A \qquad A \equiv B}{\Gamma \vdash B}\ \textsc{Conv}$$

LIMITATIONS OF $\lambda\Pi$

Traditional embeddings in $\lambda\Pi$ do not preserve equivalence:

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

Example

$$\left[(\lambda x^{C} . x)\, y\right] \;=\; \mathsf{app}\,[C]\;[C]\;\left(\mathsf{lam}\,[C]\;[C]\;(\lambda x^{[\![C]\!]} . x)\right)\, y \;\not\longrightarrow\; y$$

This is a problem for computational systems:

$$\frac{\Gamma \vdash A \qquad A \equiv B}{\Gamma \vdash B}\ \textsc{Conv}$$

- Calculus of constructions (CC) ✗
- Intuitionistic type theory (ITT) ✗
- Calculus of inductive constructions (CIC) ✗

using rewriting [Cousineau and Dowek 2007]

$$\begin{aligned}[x] &= x \\ [M\ N] &= [M]\ [N] \\ \left[\lambda x^{A} \, . \, M\right] &= \lambda x^{[\![A]\!]} \, . \, [M]\end{aligned}$$

$$\begin{aligned}[\![s]\!] &= \mathsf{U}_{s} \\ \left[\!\left[\Pi x^{A} \, . \, B\right]\!\right] &= \Pi x^{[\![A]\!]} \, . \, [\![B]\!]\end{aligned}$$

using rewriting [Cousineau and Dowek 2007]

$$
\begin{aligned}
[x] &= x \\
[M\,N] &= [M]\ [N] \\
\left[\lambda x^{A}.\,M\right] &= \lambda x^{[\![A]\!]}.\,[M] \\
[s] &= u_s \\
\left[\Pi x^{A}.\,B\right] &= \pi_{s_1,s_2}\ [A]\ (\lambda x : [\![A]\!].\,[B])
\end{aligned}
$$

$$
\begin{aligned}
[\![s]\!] &= U_s \\
\left[\!\left[\Pi x^{A}.\,B\right]\!\right] &= \Pi x^{[\![A]\!]}.\,[\![B]\!]
\end{aligned}
$$

USING REWRITING [COUSINEAU AND DOWEK 2007]

$$
\begin{aligned}
[x] &= x \\
[M\ N] &= [M]\ [N] \\
[\lambda x^A . M] &= \lambda x^{[\![A]\!]} . [M] \\
[s] &= \mathsf{u}_s \\
[\Pi x^A . B] &= \pi_{s_1,s_2}\ [A]\ (\lambda x : [\![A]\!] . [B])
\end{aligned}
$$

$$
\begin{aligned}
[\![s]\!] &= \mathsf{U}_s \\
[\![\Pi x^A . B]\!] &= \Pi x^{[\![A]\!]} . [\![B]\!] \\
[\![M]\!] &= \mathsf{T}_s\ [M]
\end{aligned}
$$

USING REWRITING [COUSINEAU AND DOWEK 2007]

$$
\begin{array}{lcl}
[x] & = & x \\
[M\,N] & = & [M]\ [N] \\
\left[\lambda x^{A} \,.\, M\right] & = & \lambda x^{[\![A]\!]} \,.\, [M] \\
[s] & = & \mathsf{u}_{s} \\
\left[\Pi x^{A} \,.\, B\right] & = & \pi_{s_1,s_2}\ [A]\ (\lambda x : [\![A]\!] \,.\, [B])
\end{array}
$$

$$
\begin{array}{lcl}
[\![s]\!] & = & \mathsf{U}_{s} \\
[\![\Pi x^{A} \,.\, B]\!] & = & \Pi x^{[\![A]\!]} \,.\, [\![B]\!] \\
[\![M]\!] & = & \mathsf{T}_{s}\ [M]
\end{array}
$$

$$
\begin{array}{lcl}
\mathsf{T}_{s_2}\ \mathsf{u}_{s_1} & \mapsto & \mathsf{U}_{s_1} \\
\mathsf{T}_{s_3}\ (\pi_{s_1,s_2}\ a\ b) & \mapsto & \Pi x^{\mathsf{T}_{s_1}\ a} \,.\, \mathsf{T}_{s_2}\ (b\ x)
\end{array}
$$

preservation of typing [cousineau and dowek 2007]

Theorem (Preservation of typing)

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

PRESERVATION OF TYPING [COUSINEAU AND DOWEK 2007]

Theorem (Preservation of equivalence)

$$M \equiv M' \implies [M] \equiv [M']$$

Theorem (Preservation of typing)

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

PRESERVATION OF TYPING [COUSINEAU AND DOWEK 2007]

Theorem (Preservation of reduction)

$$M \to^{+} M' \implies [M] \to^{+} [M']$$

Theorem (Preservation of equivalence)

$$M \equiv M' \implies [M] \equiv [M']$$

Theorem (Preservation of typing)

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

CONSERVATIVITY

TYPING IN THE EMBEDDING VS. IN THE PTS

Question:

$$\exists M.\ \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\ \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]\ ?$$

Question:

$$\exists M.\ \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\ \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]\ ?$$

Polymorphic identity function:

$$I = \lambda \alpha^{\mathsf{Type}} .\, \lambda x^{\alpha} .\, x$$

TYPING IN THE EMBEDDING VS. IN THE PTS

Question:

$$\exists M.\, \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\, \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!] \; ?$$

Polymorphic identity function:

$$I = \lambda \alpha^{\mathsf{Type}} . \lambda x^{\alpha} . x$$

$\nvdash_{\mathsf{HOL}} I : \_$

Question:

$$\exists M.\, \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\, \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]\ ?$$

Polymorphic identity function:

$$I = \lambda \alpha^{\mathsf{Type}} \,.\, \lambda x^{\alpha} \,.\, x$$

$\nvdash_{\mathsf{HOL}} I : \_$

$\vdash_{\mathsf{U}^-} I : \Pi \alpha^{\mathsf{Type}} \,.\, \alpha \to \alpha$

TYPING IN THE EMBEDDING VS. IN THE PTS

Question:

$$\exists M.\, \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\, \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!] \ ?$$

Polymorphic identity function:

$$I = \lambda \alpha^{\mathsf{Type}} \,.\, \lambda x^{\alpha} \,.\, x$$

$\nvdash_{\mathsf{HOL}} I : \_$

$\vdash_{\mathsf{U}-} I : \Pi \alpha^{\mathsf{Type}} \,.\, \alpha \to \alpha$

$\Sigma_{\mathsf{HOL}} \vdash_{\lambda\Pi R} [I] : \Pi \alpha^{\mathsf{U_{Type}}} \,.\, \mathsf{T}\,\alpha \to \mathsf{T}\,\alpha$

Question:

$$\exists M.\ \Gamma \vdash M : A \quad \Longleftarrow \quad \exists M'.\ \Sigma, [\![\Gamma]\!] \vdash M' : [\![A]\!]\ ?$$

Polymorphic identity function:

$$I = \lambda \alpha^{\mathsf{Type}} .\, \lambda x^{\alpha} .\, x$$

$$\nvdash_{\mathsf{HOL}} I : \_ \qquad\qquad \vdash_{\mathsf{U}-} I : \Pi \alpha^{\mathsf{Type}} .\, \alpha \to \alpha$$

$$\Sigma_{\mathsf{HOL}} \vdash_{\lambda\Pi\mathsf{R}} [\![I]\!] : \Pi \alpha^{\mathsf{U_{Type}}} .\, \mathsf{T}\,\alpha \to \mathsf{T}\,\alpha$$

ALERT

$$\exists M.\ \vdash_{U^-} M : \bot\ !$$

ALERT

$$\exists M.\ \vdash_{U^-} M : \bot\ !$$

$$\exists M.\ \Sigma_{\mathrm{HOL}} \vdash_{\lambda\Pi R} M : [\![\bot]\!]\ ?$$

Given: $\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi} M : [\![A]\!]$

Given: $\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi} M : [\![A]\!]$

1. Every well-typed term normalizes
2. Every normal term is the translation of a proof

Given: $\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi} M : [\![A]\!]$

1. Every well-typed term normalizes
2. Every normal term is the translation of a proof

Get: $\Gamma \vdash_{\mathcal{P}} N' : A$

Given: $\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} M : [\![A]\!]$

1. Every well-typed term normalizes
2. Every normal term is the translation of a proof

Get: $\Gamma \vdash_{\mathcal{P}} N' : A$

TRADITIONAL PROOFS OF CONSERVATIVITY

Given: $\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} M : [\![A]\!]$

1. ~~Every well-typed term normalizes~~ ?
2. Every normal term is the translation of a proof

Get: $\Gamma \vdash_{\mathcal{P}} N' : A$

Preservation of computation:

$$\begin{array}{ccccccc} M_1 & \twoheadrightarrow & M_2 & \twoheadrightarrow & M_3 & \twoheadrightarrow & \cdots \\ [M_1] & \twoheadrightarrow & [M_2] & \twoheadrightarrow & [M_3] & \twoheadrightarrow & \cdots \end{array}$$

Preservation of computation:

$$M_1 \twoheadrightarrow M_2 \twoheadrightarrow M_3 \twoheadrightarrow \cdots$$

$$[M_1] \twoheadrightarrow [M_2] \twoheadrightarrow [M_3] \twoheadrightarrow \cdots$$

Proving strong normalization for $[M]$ is at least as hard as for M!

- CC^∞? Brrr...

RELATIVE NORMALIZATION

Preservation of computation:

$$M_1 \twoheadrightarrow M_2 \twoheadrightarrow M_3 \twoheadrightarrow \cdots$$

$$[M_1] \twoheadrightarrow [M_2] \twoheadrightarrow [M_3] \twoheadrightarrow \cdots$$

Proving strong normalization for $[M]$ is at least as hard as for M!

- CC^{∞}? Brrr...

Idea: reduce only what is necessary

Relative normalization

If $\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash M : [\![A]\!]$ then $\exists M'$ such that $M \longrightarrow^* [M']$ and $\Gamma \vdash M' : A$.

Proof case:

$$\frac{\Sigma, [\![\Gamma]\!] \vdash M : \Pi x^{B} . C \qquad \Sigma, [\![\Gamma]\!] \vdash N : B}{\Sigma, [\![\Gamma]\!] \vdash M\,N : [\![A]\!]} \qquad \text{where } C\,\{x \backslash N\} = [\![A]\!]$$

Proof case:

$$\frac{\Sigma, \llbracket\Gamma\rrbracket \vdash M : \Pi x^B . C \qquad \Sigma, \llbracket\Gamma\rrbracket \vdash N : B}{\Sigma, \llbracket\Gamma\rrbracket \vdash M\,N : \llbracket A\rrbracket} \qquad \text{where } C\,\{x \backslash N\} = \llbracket A\rrbracket$$

Need a stronger induction!

Definition

Reducibility relation

$$\Vdash M : C$$

such that

$$\begin{aligned} \Vdash M : \llbracket A\rrbracket \quad &\Longleftrightarrow \quad \exists M'.\ \ M \longrightarrow^* [M'] \ \wedge \ M' : A \\ \Vdash M : \Pi x^B . C \quad &\Longleftrightarrow \quad \forall N.\ \Vdash N : B \implies \Vdash M\,N : C\,\{x \backslash N\} \end{aligned}$$

Theorem (Reducibility)

$$\Sigma_{\mathcal{P}}, \Delta \vdash_{\lambda\Pi\mathrm{R}} M : C \implies \Delta \Vdash_{\mathcal{P}} M : C$$

Theorem (Reducibility)

$$\Sigma_{\mathcal{P}}, \Delta \vdash_{\lambda\Pi\mathrm{R}} M : C \implies \Delta \Vdash_{\mathcal{P}} M : C$$

Theorem (Conservativity)

$$\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi\mathrm{R}} M : [\![A]\!] \implies \exists M'.\, M \longrightarrow^{*} [M'] \;\wedge\; \Gamma \vdash_{\mathcal{P}} M' : A$$

Theorem (Reducibility)

$$\Sigma_{\mathcal{P}}, \Delta \vdash_{\lambda\Pi R} M : C \implies \Delta \Vdash_{\mathcal{P}} M : C$$

Theorem (Conservativity)

$$\Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} M : [\![A]\!] \implies \exists M'.\ M \longrightarrow^{*} [M'] \ \wedge\ \Gamma \vdash_{\mathcal{P}} M' : A$$

- Equivalence of type inhabitation ✓

$$\exists M'.\ \Gamma \vdash_{\mathcal{P}} M' : A \iff \exists M'.\ \Sigma_{\mathcal{P}}, [\![\Gamma]\!] \vdash_{\lambda\Pi R} M : [\![A]\!]$$

- Soundness of the embedding ✓

$$\nexists M.\ \Sigma_{\text{HOL}}, \vdash_{\lambda\Pi R} M : [\![\bot]\!]$$

SUMMARY

- Have a **general embedding of pure type systems** in the $\lambda\Pi$-calculus modulo rewriting
- The embedding preserves **typing** and **computation**
- Proved that it is **conservative** using relative normalization

- Have a **general embedding of pure type systems** in the $\lambda\Pi$-calculus modulo rewriting
- The embedding preserves **typing** and **computation**
- Proved that it is **conservative** using relative normalization
    - Works for **all** normalizing systems: System F, CC, HOL, ...

- Have a **general embedding of pure type systems** in the $\lambda\Pi$-calculus modulo rewriting
- The embedding preserves **typing** and **computation**
- Proved that it is **conservative** using relative normalization
    - Works for **all** normalizing systems: System F, CC, HOL, ...
    - Works for all **non-normalizing** systems: U, U^{-}, $\lambda*$, ...

CUMULATIVITY

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \subseteq \mathsf{Type}_2 \subseteq \mathsf{Type}_3 \subseteq \cdots$$

CHALLENGES OF CUMULATIVITY

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \subseteq \mathsf{Type}_2 \subseteq \mathsf{Type}_3 \subseteq \cdots$$

- **No uniqueness** of types

Example

$$\vdash \mathsf{Type}_0 : \mathsf{Type}_1 \quad \text{and} \quad \vdash \mathsf{Type}_0 : \mathsf{Type}_2$$

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \subseteq \mathsf{Type}_2 \subseteq \mathsf{Type}_3 \subseteq \cdots$$

- **No uniqueness** of types

Example

$$\vdash \mathsf{Type}_0 : \mathsf{Type}_1 \quad \text{and} \quad \vdash \mathsf{Type}_0 : \mathsf{Type}_2$$

- Principal types...

CHALLENGES OF CUMULATIVITY

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \mathsf{Type}_1 \subseteq \mathsf{Type}_2 \subseteq \mathsf{Type}_3 \subseteq \cdots$$

- **No uniqueness** of types

Example

$$\vdash \mathsf{Type}_0 : \mathsf{Type}_1 \quad \text{and} \quad \vdash \mathsf{Type}_0 : \mathsf{Type}_2$$

- Principal types **not preserved by reduction**

Example

$$\begin{array}{ccc} \vdash & (\lambda x^{\mathsf{Type}_2} . x)\ \mathsf{Type}_0 & : & \mathsf{Type}_2 \\ & \downarrow_\beta & & \\ \vdash & \mathsf{Type}_0 & : & \mathsf{Type}_1 \end{array}$$

EMBEDDING CUMULATIVITY

Idea: use explicit casts

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash {\uparrow_i} A : \mathsf{Type}_{i+1}}$$

EMBEDDING CUMULATIVITY

Idea: use explicit casts

$$\frac{\Gamma \vdash \quad A : \mathsf{Type}_i}{\Gamma \vdash \uparrow_i A : \mathsf{Type}_{i+1}}$$

$$\begin{array}{lcl} \mathsf{u}_i & : & \mathsf{U}_{i+1} \\ \pi_i & : & \Pi a^{\mathsf{U}_i} . \Pi b^{(\mathsf{T}_i\, a \to \mathsf{U}_i)} . \mathsf{U}_i \end{array}$$

$$\begin{array}{lcl} \mathsf{T}_{i+1}\, \mathsf{u}_i & \longmapsto & \mathsf{U}_i \\ \mathsf{T}_i\, (\pi_i\, a\, b) & \longmapsto & \Pi x^{\mathsf{T}_i\, a} . \mathsf{T}_i\, (b\, x) \end{array}$$

Idea: use explicit casts

$$\frac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash \uparrow_i A : \mathsf{Type}_{i+1}}$$

$$\begin{array}{lcl} \mathsf{u}_i & : & \mathsf{U}_{i+1} \\ \pi_i & : & \Pi a^{\mathsf{U}_i} . \Pi b^{(\mathsf{T}_i\, a \to \mathsf{U}_i)} . \mathsf{U}_i \\ \uparrow_i & : & \mathsf{U}_i \to \mathsf{U}_{i+1} \end{array}$$

$$\begin{array}{lcl} \mathsf{T}_{i+1}\, \mathsf{u}_i & \longmapsto & \mathsf{U}_i \\ \mathsf{T}_i\, (\pi_i\, a\, b) & \longmapsto & \Pi x^{\mathsf{T}_i\, a} . \mathsf{T}_i\, (b\, x) \end{array}$$

Idea: use explicit casts

$$\frac{\Gamma \vdash \quad A : \mathsf{Type}_i}{\Gamma \vdash \uparrow_i\ A : \mathsf{Type}_{i+1}}$$

$$\begin{array}{lll} \mathsf{u}_i & : & \mathsf{U}_{i+1} \\ \pi_i & : & \Pi a^{\mathsf{U}_i} . \Pi b^{(\mathsf{T}_i\, a \to \mathsf{U}_i)} . \mathsf{U}_i \\ \uparrow_i & : & \mathsf{U}_i \to \mathsf{U}_{i+1} \end{array}$$

$$\begin{array}{lll} \mathsf{T}_{i+1}\, \mathsf{u}_i & \longmapsto & \mathsf{U}_i \\ \mathsf{T}_i\, (\pi_i\, a\, b) & \longmapsto & \Pi x^{\mathsf{T}_i\, a} . \mathsf{T}_i\, (b\, x) \\ \mathsf{T}_{i+1}\, (\uparrow_i\ a) & \longmapsto & \mathsf{T}_i\, a \end{array}$$

Idea: use explicit casts (Martin-Löf 1984)

$$\frac{\Gamma \vdash \quad A : \mathsf{Type}_i}{\Gamma \vdash \uparrow_i \; A : \mathsf{Type}_{i+1}}$$

$$\begin{aligned} \mathsf{u}_i &: \mathsf{U}_{i+1} \\ \pi_i &: \Pi a^{\mathsf{U}_i} \,.\, \Pi b^{(\mathsf{T}_i\, a \to \mathsf{U}_i)} \,.\, \mathsf{U}_i \\ \uparrow_i &: \mathsf{U}_i \to \mathsf{U}_{i+1} \end{aligned}$$

$$\begin{aligned} \mathsf{T}_{i+1}\, \mathsf{u}_i &\longmapsto \mathsf{U}_i \\ \mathsf{T}_i\,(\pi_i\, a\, b) &\longmapsto \Pi x^{\mathsf{T}_i\, a} \,.\, \mathsf{T}_i\,(b\, x) \\ \mathsf{T}_{i+1}\,(\uparrow_i \; a) &\longmapsto \mathsf{T}_i\, a \end{aligned}$$

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i \qquad \Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_i}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}}$$

MULTIPLICITY OF DERIVATIONS

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i \qquad \Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_i}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}}$$

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \dfrac{\Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma, x : A \vdash B : \mathsf{Type}_{i+1}}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}}$$

MULTIPLICITY OF DERIVATIONS

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i \qquad \Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_i}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}} \qquad \Uparrow_i \; (\pi_i \, A \, (\lambda x \,.\, B))$$

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \dfrac{\Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma, x : A \vdash B : \mathsf{Type}_{i+1}}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}} \qquad \pi_{i+1} \, (\Uparrow_i \; A) \; (\lambda x \,.\, \Uparrow_i \; B)$$

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i \qquad \Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_i}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}} \qquad \Uparrow_i\ (\pi_i\, A\, (\lambda x \,.\, B))$$

$$\not\equiv$$

$$\dfrac{\dfrac{\Gamma \vdash A : \mathsf{Type}_i}{\Gamma \vdash A : \mathsf{Type}_{i+1}} \qquad \dfrac{\Gamma, x : A \vdash B : \mathsf{Type}_i}{\Gamma, x : A \vdash B : \mathsf{Type}_{i+1}}}{\Gamma \vdash \Pi x^A \,.\, B : \mathsf{Type}_{i+1}} \qquad \pi_{i+1}\, (\Uparrow_i\ A)\ (\lambda x \,.\, \Uparrow_i\ B)$$

Counter-example

In the context

$$\begin{aligned} p &: \mathsf{Type}_1 \to \mathsf{Type}_1, \\ f &: \Pi c^{\mathsf{Type}_0} . \, p\, c \to \bot, \\ g &: \Pi a^{\mathsf{Type}_1} . \, \Pi b^{\mathsf{Type}_1} . \, p\, (\Pi x^a . \, b) \\ a, b &: \mathsf{Type}_0, \end{aligned}$$

we have

$$f\, (\Pi x^a . \, b) \; (g\, a\, b) \quad : \quad \bot$$

Counter-example

In the context

$$\begin{array}{rcl} p & : & \mathsf{U}_{\mathsf{Type}_1} \to \mathsf{U}_{\mathsf{Type}_1}, \\ f & : & \Pi c^{\mathsf{U}_{\mathsf{Type}_0}}\,.\,p\,c \to \bot, \\ g & : & \Pi a^{\mathsf{U}_{\mathsf{Type}_1}}\,.\,\Pi b^{\mathsf{U}_{\mathsf{Type}_1}}\,.\,p\left(\pi_{\mathsf{Type}_1}\,a\,(\lambda x\,.\,b)\right) \\ a, b & : & \mathsf{U}_{\mathsf{Type}_1}, \end{array}$$

we have

$$f\,(\pi_0\,a\,(\lambda x\,.\,b))\;(g\,(\uparrow_0\;a)\;(\uparrow_0\;b)) \quad \not: \quad \bot \quad ✗$$

Counter-example

In the context

$$\begin{aligned} p &: \mathsf{U}_{\mathsf{Type}_1} \to \mathsf{U}_{\mathsf{Type}_1}, \\ f &: \Pi c^{\mathsf{U}_{\mathsf{Type}_0}} . p\, c \to \bot, \\ g &: \Pi a^{\mathsf{U}_{\mathsf{Type}_1}} . \Pi b^{\mathsf{U}_{\mathsf{Type}_1}} . p\left(\pi_{\mathsf{Type}_1}\, a\, (\lambda x . b)\right) \\ a, b &: \mathsf{U}_{\mathsf{Type}_1}, \end{aligned}$$

we have

$$\begin{aligned} f\,(\pi_0\, a\,(\lambda x . b)) &: \mathsf{T}_1\,(p\,(\uparrow_0\ (\pi_0\, a\,(\lambda x . b)))) \to \bot \\ &\neq \\ g\,(\uparrow_0\ a)\ (\uparrow_0\ b) &: \mathsf{T}_1\,(p\,(\pi_1\,(\uparrow_0\ a)\ (\lambda x . \uparrow_0\ b))) \end{aligned}$$

Need: uniqueness of the representation of types as terms

Need: uniqueness of the representation of types as terms

Solution: add equations

$$\pi_{i+1}\,(\uparrow_i\; a)\;(\lambda x\,.\uparrow_i\;(b\,x)) \quad\equiv\quad \uparrow_i\;(\pi_i\,a\,(\lambda x\,.\,b\,x))$$

Need: uniqueness of the representation of types as terms

Solution: add rewrite rules

$$\pi_{i+1}\,(\uparrow_i\ a)\ (\lambda x\,.\uparrow_i\ (b\,x)) \quad \longmapsto \quad \uparrow_i\ (\pi_i\, a\,(\lambda x\,.\,b\,x))$$

Need: uniqueness of the representation of types as terms

Solution: add rewrite rules

$$\pi_{i+1}\,(\uparrow_i\; a)\;(\lambda x\,.\uparrow_i\;(b\,x)) \quad\longmapsto\quad \uparrow_i\;(\pi_i\,a\,(\lambda x\,.\,b\,x))$$

Requires higher-order rewriting [Saillard 2015]

Theorem (Preservation of substitution)

$$[M\{x\backslash N\}] \equiv [M]\{x\backslash [N]\}$$

Theorem (Preservation of substitution)

$$[M\{x \backslash N\}] \equiv [M]\{x \backslash [N]\}$$

Theorem (Preservation of equivalence)

$$M \equiv N \implies [M] \equiv [N]$$

Theorem (Preservation of substitution)

$$[M\{x\backslash N\}] \equiv [M]\{x\backslash [N]\}$$

Theorem (Preservation of equivalence)

$$M \equiv N \implies [M] \equiv [N]$$

Theorem (Preservation of typing)

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

- Extended the embedding to systems with cumulativity by using **explicit casts**
- Added equations to guarantee **uniqueness of term representation**
- Can be adapted to **impredicative universes** (Prop)

IMPLEMENTATIONS

Holide

https://www.rocq.inria.fr/deducteam/Holide/

- **HOL In DE**dukti

HOL4
HOL Light
HOL Zero
ProofPower
Isabelle/HOL

Holide

https://www.rocq.inria.fr/deducteam/Holide/

- **HOL I**n **DE**dukti
- Using the `OpenTheory` format

Holide

https://www.rocq.inria.fr/deducteam/Holide/

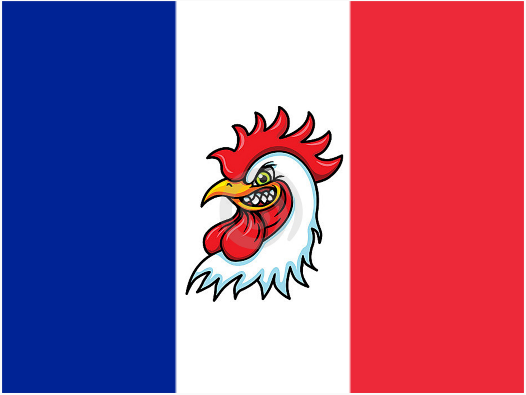
- HOL In DEdukti
- Using the `OpenTheory` format
- Translation of the standard library

	Compressed size (KB)		Time (s)	
	OpenTheory	Dedukti	Translation	Verification
Total	1702	4877	40	22

Coqine

https://gforge.inria.fr/projects/coqine/

- **COQ IN** d**E**dukti



Coqine

https://gforge.inria.fr/projects/coqine/

- **COQ IN** d**E**dukti
- Version **1.0** by Boespflug and Burel (2012)
    - Inductive types, modules ✓
    - Type : Type ✗

Coqine

https://gforge.inria.fr/projects/coqine/

- **COQ IN** d**E**dukti
- Version **1.0** by Boespflug and Burel (2012)
  - Inductive types, modules ✓
  - Type : Type ✗
- Version **2.0**
  - $\text{Type}_i : \text{Type}_{i+1}$ ✓

Coqine

https://gforge.inria.fr/projects/coqine/

- **COQ IN** d**E**dukti
- Version **1.0** by Boespflug and Burel (2012)
  - Inductive types, modules ✓
  - Type : Type ✗
- Version **2.0**
  - $\text{Type}_i : \text{Type}_{i+1}$ ✓
  - Universe polymorphism ✗
  - Anonymous fixpoints ✗
  - Functors ✗

Coqine

https://gforge.inria.fr/projects/coqine/

- **COQ IN** d**E**dukti
- Version **1.0** by Boespflug and Burel (2012)
  - Inductive types, modules ✓
  - Type : Type ✗
- Version **2.0**
  - $\text{Type}_i : \text{Type}_{i+1}$ ✓
  - Universe polymorphism ✗
  - Anonymous fixpoints ✗
  - Functors ✗
- Interoperability with HOL (A. & Cauderlier 2015)

Krajono

https://gforge.inria.fr/projects/krajono/

- Matita in Dedukti

Krajono

https://gforge.inria.fr/projects/krajono/

- Matita in Dedukti
- Features:
  - No universe polymorphism ✓
  - No anonymous fixpoints ✓
  - No modules ✓

Krajono

https://gforge.inria.fr/projects/krajono/

- Matita in Dedukti
- Features:
  - No universe polymorphism ✓
  - No anonymous fixpoints ✓
  - No modules ✓
  - Proof irrelevance ✗

Krajono

https://gforge.inria.fr/projects/krajono/

- Matita in Dedukti
- Features:
  - No universe polymorphism ✓
  - No anonymous fixpoints ✓
  - No modules ✓
  - Proof irrelevance ✗
- Translation of the `arithmetics` library

	Compiled size (KB)		Time (s)	
	Matita	Dedukti	Matita	Dedukti
Total	3120	11955	438	1412

- There is a wide **gap between theory and practice**
- It can be very hard to obtain **usable proof objects**
- We need support for **well-specified proof formats**
    - OpenTheory [Hurd 2011]
    - L∃∀N [De Moura 2015]

CONCLUSION

- Using the $\lambda\Pi$-calculus modulo as a logical framework for **independent proof checking** and **proof interoperability**
- Embedding of computational higher-order logics that is **sound** and **complete**
- Implementation of automated proof translations:
  HOL, Coq, and Matita

Translations

- Functors (Coq)
- Local fixpoints (Coq)
- Universe polymorphism (Coq)
- Proof irrelevance (Matita)
- Intersection type systems?

Translations

- Functors (Coq)
- Local fixpoints (Coq)
- Universe polymorphism (Coq)
- Proof irrelevance (Matita)
- Intersection type systems?

Interoperability

Gotta catch ém all !

Gotta catch ém all !

Thank you!

Gotta catch 'em all!

Ala, Andrea, Brigitte, Caterina, Catherine, Dale, the Deducteammates, Erik, Esmeralda, Felicity, Francesca, Gilles, Guillaume, Hanane, Herman, Hugo, John, Julia, Olivier, the Parsifalians, Riquet, Roger, the Stockholm University Logicians, Ygritte, Zak

Ali Assaf and Guillaume Burel.
Translating HOL to Dedukti.
In *Proceedings of PxTP*, 2015.

Ali Assaf and Raphaël Cauderlier.
Mixing HOL and Coq in Dedukti (Extended Abstract).
In *Proceedings of PxTP*, 2015.

Ali Assaf.
A calculus of constructions with explicit subtyping.
In *Proceedings of TYPES*, 2014.

Ali Assaf.
Conservativity of embeddings in the lambda-Pi calculus modulo rewriting.
In *Proceedings of TLCA*, 2015.

BIBLIOGRAPHY II

- Henk Barendregt.
  Lambda calculi with types.
  Oxford University Press, 1992.

- Denis Cousineau and Gilles Dowek.
  Embedding pure type systems in the $\lambda\Pi$-calculus modulo.
  In *Proceedings of TLCA*, 2007.

- Robert Harper, Furio Honsell, and Gordon Plotkin.
  A framework for defining logics.
  Journal of the ACM, 1993.

- Per Martin-Löf and Giovanni Sambin.
  Intuitionistic type theory.
  Bibliopolis Naples, 1984.