Paramcoq-iff: Parametricity and Uniformity of Propositions

Abhishek Anand and Greg Morrisett

Cornell University

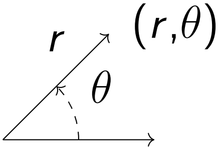
(paper and slides are on the wiki)

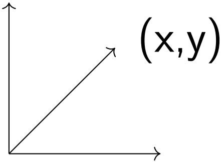
Parametricity: Reynold's vision

Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

Parametricity: Reynold's vision

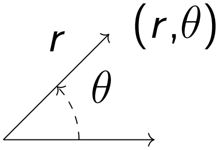
Complex Variables

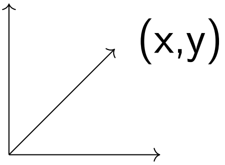




Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

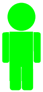
Complex Variables



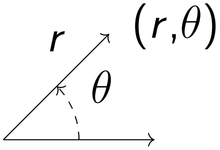


Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

Parametricity: Reynold's vision



Complex Variables

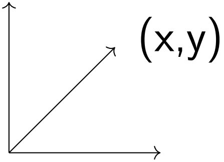


$+$

$\times$

$\| \quad \|$

$\vdots$



Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

Parametricity: Reynold's vision

Complex Variables

Cl :={

$\mathbb{C}$:Type

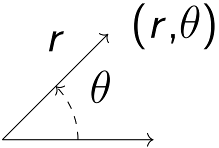
$+: \ \mathbb{C} \to \mathbb{C} \to \mathbb{C}$

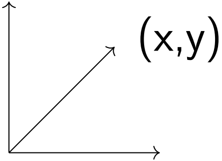
$\times: \ \mathbb{C} \to \mathbb{C} \to \mathbb{C}$

$|| \ ||: \ \mathbb{C} \to \mathbb{R}$

$\vdots$

}

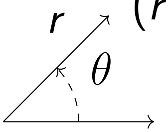




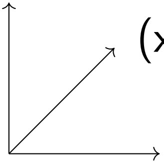
Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

Complex Variables

$$
\begin{aligned}
&\mathsf{CI} := \{ \\
&\mathbb{C} : \texttt{Type} \\
&+ : \ \mathbb{C} \to \mathbb{C} \to \mathbb{C} \\
&\times : \ \mathbb{C} \to \mathbb{C} \to \mathbb{C} \\
&|| \ || : \ \mathbb{C} \to \mathbb{R} \\
&\vdots \\
&\}
\end{aligned}
$$



(r,θ)

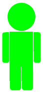


(x,y)

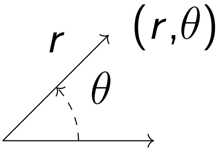
2^{nd} lecture onwards : $\mathsf{CI} \to \ \ldots$

Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

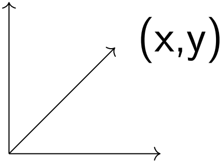
Parametricity: Reynold's vision



Complex Variables



$$
\begin{aligned}
&\mathrm{Cl} := \{\\
&\mathbb{C} : \texttt{Type}\\
&+ : \ \mathbb{C} \to \mathbb{C} \to \mathbb{C}\\
&\times : \ \mathbb{C} \to \mathbb{C} \to \mathbb{C}\\
&|| \ || : \ \mathbb{C} \to \mathbb{R}\\
&\vdots\\
&\}
\end{aligned}
$$



behaves uniformly

2^{nd} lecture onwards : $\mathrm{Cl} \to \ldots$

Reynolds, "Types, Abstraction and Parametric Polymorphism",1983

Progress towards Reynold's vision

System F

Progress towards Reynold's vision

System F : ~~proofs~~

Progress towards Reynold's vision

System F : ~~proofs~~

$\vdots$

Progress towards Reynold's vision

System F : ~~proofs~~

⋮

Coq/Agda

Bernardy 2011, Keller and Lasson 2012

Progress towards Reynold's vision

System F : ~~proofs~~

$\vdots$

Coq/Agda : complex analysis course

Bernardy 2011, Keller and Lasson 2012

System F : ~~proofs~~

⋮

Coq/Agda : complex analysis course

Bernardy 2011, Keller and Lasson 2012

program translation

Progress towards Reynold's vision

System F : ~~proofs~~

⋮

Coq/Agda : complex analysis course

Bernardy 2011, Keller and Lasson 2012

program translation

computations✓

Progress towards Reynold's vision

System F : ~~proofs~~

⋮

Coq/Agda : complex analysis course

Bernardy 2011, Keller and Lasson 2012

program translation

computations✓

logic?

Missing: Uniformity of Coq's logic

$$f : \forall\ T{:}\mathtt{Type},\ T \to \mathsf{bool}$$

Missing: Uniformity of Coq's logic

$$f : \forall\ T{:}\texttt{Type},\ T \to \mathsf{bool}$$

$[\![f]\!]$

Missing: Uniformity of Coq's logic

$$f : \forall\ T{:}\mathtt{Type},\ T \to \mathsf{bool}$$

$$\begin{aligned}\llbracket f \rrbracket :\ & \forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r : T_1 \to T_2 \to \mathtt{Type}) \\ & (ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \to \\ & f\ T_1\ ta_1 = \quad f\ T_2\ ta_2\end{aligned}$$

Missing: Uniformity of Coq's logic

constant

$$f : \forall\ T{:}\mathtt{Type},\ T \to \mathsf{bool}$$

$$\begin{aligned}[\![f]\!] : \forall\ &(T_1\ T_2 : \mathtt{Type})\ (T_r : T_1 \to T_2 \to \mathtt{Type})\\ &(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \to\\ &f\ T_1\ ta_1 = \ f\ T_2\ ta_2\end{aligned}$$

Missing: Uniformity of Coq's logic

constant

$$f : \forall\ T{:}\mathtt{Type},\ T \to \mathsf{bool}$$

$\lambda\_\_,$ True

$$[\![f]\!] : \forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r : T_1 \to T_2 \to \mathtt{Type})$$
$$(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \to$$
$$f\ T_1\ ta_1 = f\ T_2\ ta_2$$

Missing: Uniformity of Coq's logic

$$f : \forall\ T{:}\texttt{Type},\ T \rightarrow\ \cancel{\texttt{bool}}\texttt{Prop}$$

$$\begin{aligned}[\![f]\!] : \forall\ & (T_1\ T_2 : \texttt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \texttt{Type}) \\ & (ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow \\ & f\ T_1\ ta_1 = \ \ f\ T_2\ ta_2\end{aligned}$$

Missing: Uniformity of Coq's logic

undecidable predicates

$$f : \forall\ T{:}\mathtt{Type},\ T \rightarrow\ \cancel{\mathtt{bool}}\mathtt{Prop}$$

$$\begin{aligned}[\![f]\!] : \forall\ & (T_1\ T_2 : \mathtt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \mathtt{Type}) \\ & (ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow \\ & f\ T_1\ ta_1 = \ f\ T_2\ ta_2\end{aligned}$$

Missing: Uniformity of Coq's logic

undecidable predicates

$$f : \forall\ T{:}\texttt{Type},\ T \rightarrow\ \cancel{\texttt{bool}}\texttt{Prop}$$

$$\llbracket f \rrbracket : \forall\ (T_1\ T_2 : \texttt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \texttt{Type})$$
$$(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$\boxed{f\ T_1\ ta_1 = f\ T_2\ ta_2}$$

Missing: Uniformity of Coq's logic

undecidable predicates

$$f : \forall\ T{:}\texttt{Type},\ T \rightarrow\ \cancel{\texttt{bool}}\texttt{Prop}$$

$$\begin{aligned}\llbracket f \rrbracket : \forall\ &(T_1\ T_2 : \texttt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \texttt{Type})\\ &(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow\\ &f\ T_1\ ta_1 \quad \rightarrow f\ T_2\ ta_2 \rightarrow \texttt{Prop}\end{aligned}$$

Missing: Uniformity of Coq's logic

undecidable predicates

$$f\colon \forall\ T{:}\texttt{Type},\ T \rightarrow\ \cancel{\texttt{bool}}\texttt{Prop}$$

$$\llbracket f \rrbracket\colon \forall\ (T_1\ T_2 : \texttt{Type})\ (T_r\colon\ T_1 \rightarrow T_2 \rightarrow \texttt{Type})$$
$$(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$\boxed{f\ T_1\ ta_1} \rightarrow \boxed{f\ T_2\ ta_2} \rightarrow \texttt{Prop}$$

Missing: Uniformity of Coq's logic

undecidable predicates

$$f : \forall\ T{:}\mathtt{Type},\ T \rightarrow\ \cancel{\mathtt{bool}}\mathtt{Prop}$$

$$\begin{aligned}[\![f]\!] : \forall\ &(T_1\ T_2 : \mathtt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \mathtt{Type})\\ &(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow\\ &\boxed{f\ T_1\ ta_1} \rightarrow \boxed{f\ T_2\ ta_2} \rightarrow \mathtt{Prop}\end{aligned}$$

$\lambda\_\_,$ True

Missing: Uniformity of Coq's logic

undecidable predicates

$$f : \forall\ T\text{:}\mathtt{Type},\ T \rightarrow\ \cancel{\mathtt{bool}}\mathtt{Prop}$$

utterly useless

$$[\![f]\!] : \forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \mathtt{Type})$$
$$(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$\boxed{f\ T_1\ ta_1} \rightarrow \boxed{f\ T_2\ ta_2} \rightarrow \mathtt{Prop}$$

$\lambda \_ \_,$ True

Missing: Uniformity of Coq's logic

$$f\colon \forall\ T{:}\texttt{Type},\ T \to\ \cancel{\mathsf{bool}}\texttt{Prop}$$

$$\begin{aligned}[\![f]\!]\colon \forall\ & (T_1\ T_2 : \texttt{Type})\ (T_r\colon\ T_1 \to\ T_2 \to \texttt{Type})\\ & (ta_1 :\ T_1)\ (ta_2 :\ T_2),\ T_r\ ta_1\ ta_2 \to\\ & f\ \ T_1\ ta_1 \quad \to f\ \ T_2\ ta_2 \to \texttt{Prop}\end{aligned}$$

$$[\![\mathsf{bool}]\!] := \lambda\ (b_1\ b_2\colon\ \mathsf{bool})\ ,\ b_1 = b_2$$

$$[\![\texttt{Prop}]\!] := \lambda\ (b_1\ b_2\colon\ \texttt{Prop}),\ b_1 \to\ b_2 \to \texttt{Prop}$$

Missing: Uniformity of Coq's logic

$$f\colon \forall\ T{:}\texttt{Type},\ T \rightarrow\ \texttt{Prop}$$

$$\begin{aligned}[\![f]\!]\colon \forall\ &(T_1\ T_2 : \texttt{Type})\ (T_r\colon\ T_1 \rightarrow\ T_2 \rightarrow \texttt{Type})\\ &(ta_1 :\ T_1)\ (ta_2 :\ T_2),\ T_r\ ta_1\ ta_2 \rightarrow\\ &f\ T_1\ ta_1 \quad \rightarrow f\ \ T_2\ ta_2 \rightarrow \texttt{Prop}\end{aligned}$$

$$[\![\texttt{Prop}]\!] := \lambda\ (b_1\ b_2\colon\ \texttt{Prop}),\ b_1 \rightarrow\ b_2 \rightarrow \texttt{Prop}$$

Missing: Uniformity of Coq's logic

$$f : \forall\ T\text{:}\texttt{Type},\ T \rightarrow\ \texttt{Prop}$$

$$\begin{aligned}[\![f]\!] : \forall\ &(T_1\ T_2 : \texttt{Type})\ (T_r : T_1 \rightarrow T_2 \rightarrow \texttt{Type})\\ &(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow\\ &f\ T_1\ ta_1 \leftrightarrow f\ T_2\ ta_2\end{aligned}$$

$$[\![\texttt{Prop}]\!] := \lambda\ (b_1\ b_2 : \texttt{Prop}),\ b_1 \rightarrow b_2 \rightarrow \texttt{Prop}$$

Missing: Uniformity of Coq's logic

$$f \colon \forall\ T{:}\texttt{Type},\ T \to\ \texttt{Prop}$$

$$\begin{aligned}[\![f]\!] \colon \forall\ &(T_1\ T_2 : \texttt{Type})\ (T_r \colon\ T_1 \to\ T_2 \to \texttt{Type})\\ &(ta_1\ :\ T_1)\ (ta_2\ :\ T_2),\ T_r\ ta_1\ ta_2 \to\\ &f\ T_1\ ta_1\ \leftrightarrow\ f\ T_2\ ta_2\end{aligned}$$

$$\begin{aligned}[\![\texttt{Prop}]\!] := \lambda\ (b_1\ b_2 \colon\ \texttt{Prop}),\ &b_1 \to b_2 \to \texttt{Prop}\\ &\times\ b_1 \leftrightarrow b_2\end{aligned}$$

$$f\colon \forall\ T{:}\mathtt{Type},\ T \to\ \mathtt{Prop}$$

$$\begin{aligned}[\![f]\!]\colon \forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r\colon\ T_1 \to\ T_2 \to \mathtt{Type})\\ (ta_1 :\ T_1)\ (ta_2 :\ T_2),\ T_r\ ta_1\ ta_2 \to\\ f\ \ T_1\ ta_1\ \ \leftrightarrow\ \ f\ \ T_2\ ta_2\end{aligned}$$

$$\begin{aligned}[\![\mathtt{Prop}]\!] := \lambda\ (b_1\ b_2\colon\ \mathtt{Prop}),\ b_1 \to\ b_2 \to \mathtt{Prop}\\ \times\ b_1 \leftrightarrow b_2\end{aligned}$$

Impossibility

?

$$f : \forall\ T{:}\mathtt{Type},\ T \rightarrow\ \mathtt{Prop}$$

$$\forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r{:}\ T_1 \rightarrow T_2 \rightarrow \mathtt{Type})$$
$$(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$f\ T_1\ ta_1\ \leftrightarrow\ f\ T_2\ ta_2$$

Impossibility

$\lambda\ (T{:}\texttt{Type})\ (\_{:}T),\ \forall\ (a\ b{:}T),\ a{=}b$

$$f{:}\ \forall\ T{:}\texttt{Type},\ T\ \rightarrow\ \texttt{Prop}$$

$$\forall\ (T_1\ T_2\ :\ \texttt{Type})\ (T_r{:}\ T_1 \rightarrow T_2 \rightarrow \texttt{Type})$$
$$(ta_1\ :\ T_1)\ (ta_2\ :\ T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$f\ T_1\ ta_1\ \leftrightarrow\ f\ T_2\ ta_2$$

Impossibility

$\lambda\ (T{:}\mathtt{Type})\ (\_{:}T),\ \forall\ (a\ b{:}T),\ a{=}b$

$$f : \forall\ T{:}\mathtt{Type},\ T \to\ \mathtt{Prop}$$

$$\begin{array}{l} \forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r{:}\ T_1 \to T_2 \to \mathtt{Type}) \\ \quad (ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \to \\ \quad f\ T_1\ ta_1 \leftrightarrow f\ T_2\ ta_2 \end{array}$$

unit

bool

Parametricity is too liberal

$\lambda\ (T{:}\mathtt{Type})\ (\_{:}T),\ \forall\ (a\ b{:}T),\ a{=}b$

$$f : \forall\ T{:}\mathtt{Type},\ T \rightarrow\ \mathtt{Prop}$$

$\lambda\_\_,\ \mathsf{True}$

$$\forall\ (T_1\ T_2 : \mathtt{Type})\ (T_r{:}\ T_1 \rightarrow T_2 \rightarrow \mathtt{Type})$$
$$(ta_1 : T_1)\ (ta_2 : T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$f\ T_1\ ta_1\ \leftrightarrow\ f\ T_2\ ta_2$$

unit

bool

Parametricity is too liberal

$\lambda\ (T{:}\mathtt{Type})\ (\_{:}T),\ \forall\ (a\ b{:}T),\ a{=}b$

$$f:\ \forall\ T{:}\mathtt{Type},\ T\ \rightarrow\ \mathtt{Prop}$$

Minimally restrict parametricity relations

$\lambda \_\,\_,\ \mathsf{True}$

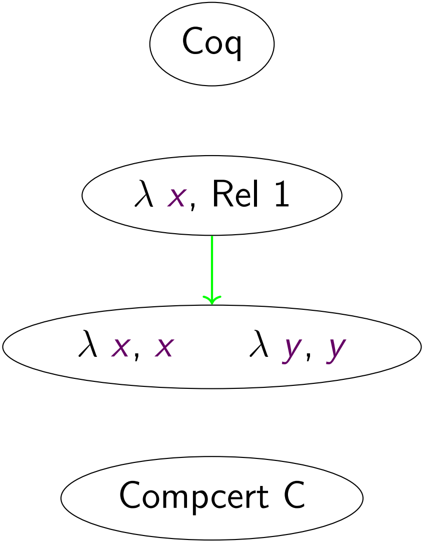
$$\forall\ (T_1\ T_2\ :\ \mathtt{Type})\ (T_r{:}\ T_1 \rightarrow T_2 \rightarrow \mathtt{Type})$$
$$(ta_1\ :\ T_1)\ (ta_2\ :\ T_2),\ T_r\ ta_1\ ta_2 \rightarrow$$
$$f\ T_1\ ta_1\ \leftrightarrow\ f\ T_2\ ta_2$$

unit

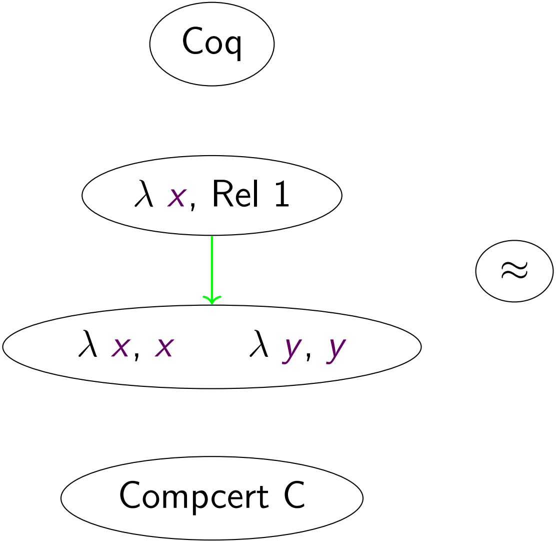
bool

Undecidable proposition: example from CertiCoq

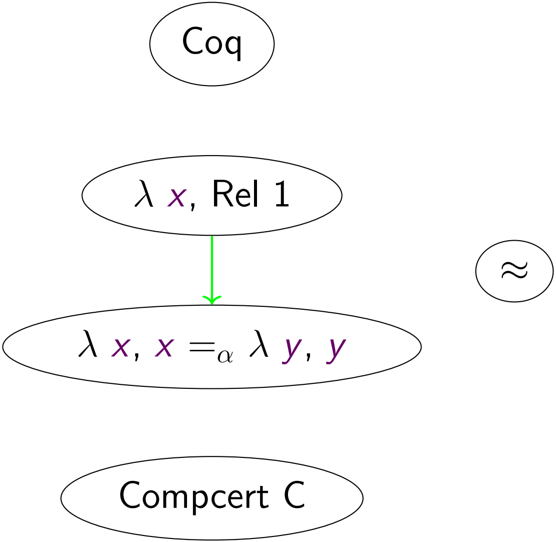
Undecidable proposition: example from CertiCoq



Undecidable proposition: example from CertiCoq



Undecidable proposition: example from CertiCoq



AnyRel translation (Keller and Lasson 2012)

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

AnyRel translation (Keller and Lasson 2012)

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}[\![\forall x{:}A.B]\!] := \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2),\\ \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!]x\, x_2),\\ [\![B]\!](f\, x)(f_2\, x_2)\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}[\![\forall x : A.B]\!] := \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2),\\ \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!] x\, x_2),\\ [\![B]\!](f\, x)(f_2\, x_2)\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}[\![\forall x{:}A.B]\!] := \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2),\\ \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!]\, x\, x_2),\\ [\![B]\!](f\, x)(f_2\, x_2)\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}[\![\forall x{:}A.B]\!] := \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2),\\ \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!]x\, x_2),\\ [\![B]\!](f\, x)(f_2\, x_2)\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$\Gamma \vdash a\colon\ T \qquad \Rightarrow \qquad [\![\Gamma]\!] \vdash [\![a]\!]\colon\ ([\![T]\!]\ a\ a_2)$$

$$[\![s]\!] := \lambda(x\ x_2 : \mathtt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}[\![\forall x\colon A.B]\!] := \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2),\\ \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!] x\, x_2),\\ [\![B]\!](f\, x)(f_2\, x_2)\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$\Gamma \vdash a \colon T \qquad \Rightarrow \qquad [\![\Gamma]\!] \vdash [\![a]\!] \colon ([\![T]\!]\ a\ a_2)$$

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}[\![\forall x \colon A.B]\!] := \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2),\\ \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!] x\, x_2),\\ [\![B]\!](f\, x)(f_2\, x_2)\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$\Gamma \vdash a\colon\ T \qquad \Rightarrow \qquad [\![\Gamma]\!] \vdash [\![a]\!]\colon\ ([\![T]\!]\ a\ a_2)$$

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}
[\![\forall x\colon A.B]\!] :=\ & \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2), \\
& \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!] x\, x_2), \\
& [\![B]\!](f\, x)(f_2\, x_2) \\
[\![x]\!] :=\ & x_r \\
[\![\lambda x : A, B]\!] :=\ & \lambda(x : A)(x_2 : A_2)(x_r : [\![A]\!] x\, x_2), [\![B]\!] \\
[\![(A\, B)]\!] :=\ & ([\![A]\!]\ B\ B_2\ [\![B]\!])
\end{aligned}$$

AnyRel translation (Keller and Lasson 2012)

$$\Gamma \vdash a\colon\ T \qquad \Rightarrow \qquad [\![\Gamma]\!] \vdash [\![a]\!]\colon\ ([\![T]\!]\ a\ a_2)$$

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

$$\begin{aligned}
[\![\forall x{:}A.B]\!] &:= \lambda(f : \forall x : A.B)(f_2 : \forall x_2 : A_2.B_2), \\
&\qquad \forall(x : A)(x_2 : A_2)(x_r : [\![A]\!]x\, x_2), \\
&\qquad\ [\![B]\!](f\, x)(f_2\, x_2) \\
[\![x]\!] &:= x_r \\
[\![\lambda x : A, B]\!] &:= \lambda(x : A)(x_2 : A_2)(x_r : [\![A]\!]x\, x_2), [\![B]\!] \\
[\![(A\, B)]\!] &:= ([\![A]\!]\ B\ B_2\ [\![B]\!])
\end{aligned}$$

Our translation

$$[\![s]\!] := \lambda(x\, x_2 : \texttt{Prop}), x \to x_2 \to s$$

Our translation

$$[\![\texttt{Prop}]\!]_{iso} := \lambda(x\, x_2 : \texttt{Prop}), x \rightarrow x_2 \rightarrow s \quad \times\, x \leftrightarrow x_2$$

Our translation

$$[\![\mathtt{Prop}]\!]_{iso} := \lambda(x\ x_2 : \mathtt{Prop}), x \to x_2 \to s \quad \times x \leftrightarrow x_2$$
$$[\![\mathtt{Type}_i]\!]_{iso} := \lambda(x\ x_2 : \mathtt{Type}_i), \{R : x \to x_2 \to s \& P\ R\}$$

Our translation

$$[\![\texttt{Prop}]\!]_{iso} := \lambda(x\ x_2 : \texttt{Prop}), x \to x_2 \to s \quad \times\ x \leftrightarrow x_2$$

$$[\![\texttt{Type}_i]\!]_{iso} := \lambda(x\ x_2 : \texttt{Type}_i), \{R : x \to x_2 \to s \& P\ R\}$$

weakest condition

Our translation

$$[\![\texttt{Prop}]\!]_{iso} := \lambda(x\, x_2 : \texttt{Prop}),\, x \to x_2 \to s \quad \times\ x \leftrightarrow x_2$$

$$[\![\texttt{Type}_i]\!]_{iso} := \lambda(x\, x_2 : \texttt{Type}_i),\, \{R : x \to x_2 \to s \& P\ R\}$$

weakest condition

up next: weakest conditions for relations of types occuring in propositions

Propositions in Coq

- $\forall$
- Inductives

A:`Type` B:$A \to$ `Prop`

$(\forall\ (x{:}A).\ B\ x)$ `Prop`

A:`Type` $\qquad$ $B : A \rightarrow$ `Prop`

$$\Big\updownarrow [\![A]\!]$$

A_2:`Type`

$(\forall\ (x{:}A).\ B\ x)$ `Prop`

$$\begin{array}{ccc} A{:}\texttt{Type} & & B{:}A \to \texttt{Prop} \\ \Big\updownarrow [\![A]\!] & & \Big\updownarrow [\![B]\!] \\ A_2{:}\texttt{Type} & & B_2{:}A_2 \to \texttt{Prop} \end{array}$$

$(\forall\ (x{:}A).\ B\ x)\ \texttt{Prop}$

$$\begin{array}{ccc}
A{:}\texttt{Type} & \qquad & B{:}A \to \texttt{Prop} \\
\Big\updownarrow [\![A]\!] & & \Big\updownarrow [\![B]\!] \\
A_2{:}\texttt{Type} & & B_2{:}A_2 \to \texttt{Prop}
\end{array}$$

$$[\![B]\!] \Rightarrow \forall\ (a{:}A)\ (a_2{:}A_2),\ [\![A]\!]\ a\ a_2 \to (B\ a \leftrightarrow B_2\ a_2)$$

$(\forall\ (x{:}A).\ B\ x)$

$$\begin{array}{ccc} A{:}\texttt{Type} & & B{:}A \to \texttt{Prop} \\ \updownarrow [\![A]\!] & & \updownarrow [\![B]\!] \\ A_2{:}\texttt{Type} & & B_2{:}A_2 \to \texttt{Prop} \end{array}$$

$$[\![B]\!] \Rightarrow \forall\ (a{:}A)\ (a_2{:}A_2),\ [\![A]\!]\ a\ a_2 \to (B\ a \leftrightarrow B_2\ a_2)$$

$$(\forall\ (x{:}A).\ B\ x) \qquad \leftrightarrow \qquad (\forall\ (x_2{:}A_2).\ B_2\ x_2)$$

$$
\begin{array}{ll}
A := \mathrm{True},\ A_2 := \mathrm{False} & B := B_2 := \lambda\_, \mathrm{False} \\
A : \texttt{Type} & B : A \to \texttt{Prop} \\
\updownarrow [\![A]\!] & \updownarrow [\![B]\!] \\
A_2 : \texttt{Type} & B_2 : A_2 \to \texttt{Prop}
\end{array}
$$

$$[\![B]\!] \Rightarrow \forall\ (a{:}A)\ (a_2{:}A_2),\ [\![A]\!]\ a\ a_2 \to (B\ a \leftrightarrow B_2\ a_2)$$

$$(\forall\ (x{:}A).\ B\ x) \qquad \leftrightarrow \qquad (\forall\ (x_2{:}A_2).\ B_2\ x_2)$$

$$\begin{array}{ccc} \mathrm{A}{:}\texttt{Type} & \qquad & \mathrm{B}{:}\mathrm{A} \to \texttt{Prop} \\ \big\updownarrow [\![\mathrm{A}]\!] & & \big\updownarrow [\![\mathrm{B}]\!] \\ \mathrm{A}_2{:}\texttt{Type} & & \mathrm{B}_2{:}\mathrm{A}_2 \to \texttt{Prop} \end{array}$$

$$[\![\mathrm{B}]\!] \Rightarrow \forall\ (a{:}\mathrm{A})\ (a_2{:}\mathrm{A}_2),\ [\![\mathrm{A}]\!]\ a\ a_2 \to (\mathrm{B}\ a \leftrightarrow \mathrm{B}_2\ a_2)$$

$$\begin{aligned} \text{Total}\ [\![\mathrm{A}]\!] := {} & \forall\ (a{:}A),\ \{a_2 : A_2\ \&\ [\![A]\!]\ a\ a_2\} \\ & \wedge \forall\ (a{:}A),\ \{a_2 : A_2\ \&\ [\![A]\!]\ a\ a_2\} \end{aligned}$$

$$(\forall\ (x{:}\mathrm{A}).\ \mathrm{B}\ x) \qquad \leftrightarrow \qquad (\forall\ (x_2{:}\mathrm{A}_2).\ \mathrm{B}_2\ x_2)$$

Propositions in Coq

- $\forall$
- Inductives
  - equality (Coq.Init.Logic.eq)

    $\vdots$

Propositions in Coq

- $\forall$ ✓
- Inductives
  - equality (Coq.Init.Logic.eq)

    $\vdots$

A:`Type` x:A y:A

(eq A x y) `Prop`

Coq.Init.Logic.eq

$A{:}\mathtt{Type}$ $\quad$ $x{:}A$ $\quad$ $y{:}A$

$$\begin{array}{l} A{:}\mathtt{Type} \\ \updownarrow [\![A]\!] \\ A_2{:}\mathtt{Type} \end{array}$$

$(\mathbf{eq}\ A\ x\ y)\ \mathtt{Prop}$

Coq.Init.Logic.eq

$$\begin{array}{ccc}
\mathsf{A}{:}\mathtt{Type} & \mathsf{x}{:}\mathsf{A} & \mathsf{y}{:}\mathsf{A} \\
\Big\updownarrow [\![\mathsf{A}]\!] & \Big\updownarrow [\![\mathsf{x}]\!]{:}[\![\mathsf{A}]\!]\ x\ x_2 & \Big\updownarrow [\![\mathsf{y}]\!]{:}[\![\mathsf{A}]\!]\ y\ y_2 \\
\mathsf{A}_2{:}\mathtt{Type} & \mathsf{x}_2{:}\mathsf{A}_2 & \mathsf{y}_2{:}\mathsf{A}_2
\end{array}$$

$(\mathsf{eq}\ \mathsf{A}\ x\ y)\ \mathtt{Prop}$

Coq.Init.Logic.eq

$$
\begin{array}{ccc}
A{:}\mathtt{Type} & x{:}A & y{:}A \\
\updownarrow [\![A]\!] & \updownarrow [\![x]\!]{:}[\![A]\!]\ x\ x_2 & \updownarrow [\![y]\!]{:}[\![A]\!]\ y\ y_2 \\
A_2{:}\mathtt{Type} & x_2{:}A_2 & y_2{:}A_2
\end{array}
$$

$$(\mathrm{eq}\ A\ x\ y) \quad \leftrightarrow \quad (\mathrm{eq}\ A_2\ x_2\ y_2)$$

Coq.Init.Logic.eq

$$\begin{array}{ccc} \mathsf{A}{:}\mathtt{Type} & \mathsf{x}{:}\mathsf{A} & \mathsf{y}{:}\mathsf{A} \\ \updownarrow [\![\mathsf{A}]\!] & \updownarrow [\![\mathsf{x}]\!]{:}[\![\mathsf{A}]\!]\ x\ x_2 & \updownarrow [\![\mathsf{y}]\!]{:}[\![\mathsf{A}]\!]\ y\ y_2 \\ \mathsf{A}_2{:}\mathtt{Type} & \mathsf{x}_2{:}\mathsf{A}_2 & \mathsf{y}_2{:}\mathsf{A}_2 \end{array}$$

$$\mathrm{OneToOne}\ [\![\mathsf{A}]\!] := \forall\ (x\ y{:}\mathsf{A})\ (x_2\ y_2{:}\mathsf{A}_2), \\ [\![\mathsf{A}]\!]\ x\ x_2 \to [\![\mathsf{A}]\!]\ y\ y_2 \to (\mathsf{eq}\ \mathsf{A}\ x\ y \leftrightarrow \mathsf{eq}\ \mathsf{A}_2\ x_2\ y_2)$$

$$(\mathsf{eq}\ \mathsf{A}\ x\ y) \qquad \leftrightarrow \qquad (\mathsf{eq}\ \mathsf{A}_2\ x_2\ y_2)$$

General Inductives

W props

$\texttt{Inductive}\ \mathsf{IWP}\ (I\ A : \texttt{Type})\ (B : A \to \texttt{Type})$
$\quad (AI : A \to I)\ (BI : \forall\ (a : A),\ B\ a \to I)$
$\quad : \forall\ (i{:}I),\ \texttt{Prop} :=$
$\mathsf{node} : \forall\ (a : A)$
$\quad (brs : \forall\ b : B\ a,\ \mathsf{IWP}\ I\ A\ B\ AI\ BI\ (BI\ a\ b)),$
$\quad \mathsf{IWP}\ I\ A\ B\ AI\ BI\ (AI\ a).$

W props

OneToOne

Total Total

```
Inductive IWP (I A : Type) (B : A → Type)
  (AI : A → I) (BI : ∀ (a : A), B a → I)
  : ∀ (i:I), Prop :=
node : ∀ (a : A)
  (brs : ∀ b : B a, IWP I A B AI BI (BI a b)),
  IWP I A B AI BI (AI a).
```

W props

OneToOne

Total Total

```
Inductive IWP (I A : Type) (B : A → Type)
   (AI : A → I) (BI : ∀ (a : A), B a → I)
   : ∀ (i:I), Prop :=
node : ∀ (a : A)
   (brs : ∀ b : B a, IWP I A B AI BI (BI a b)),
   IWP I A B AI BI (AI a).
```

$\text{IWP A B AI BI i} \leftrightarrow \text{IWP } \mathrm{A_2\ B_2\ AI_2\ BI_2\ i_2}$

W props

OneToOne only

Total Total

$$\texttt{Inductive}\ \mathsf{IWP}\ (I\ A : \texttt{Type})\ (B : A \to \texttt{Type})$$
$$(AI : A \to I)\ (BI : \forall\ (a : A), B\ a \to I)$$
$$: \forall\ (i{:}I), \texttt{Prop} :=$$
$$\mathsf{node} : \forall\ (a : A)$$
$$(brs : \forall\ b : B\ a, \mathsf{IWP}\ I\ A\ B\ AI\ BI\ (BI\ a\ b)),$$
$$\mathsf{IWP}\ I\ A\ B\ AI\ BI\ (AI\ a).$$

$$\mathsf{IWP}\ \mathrm{A\ B\ AI\ BI\ i} \leftrightarrow \mathsf{IWP}\ \mathrm{A_2\ B_2\ AI_2\ BI_2\ i_2}$$

Recap: Propositions in Coq

- $\forall$
- Inductives

Recap: Propositions in Coq

- $\forall$✓
- Inductives✓

Recap: Propositions in Coq

- $\forall$✓
- Inductives✓

Summary:

Recap: Propositions in Coq

- $\forall$ ✓
- Inductives ✓

Summary:

- Total and OneToOne for types mentioned in Props $\Rightarrow$ uniformity of props

Recap: Propositions in Coq

- $\forall$ ✓
- Inductives ✓

Summary:

- Total and OneToOne for types mentioned in Props $\Rightarrow$ uniformity of props
- One or both may not be needed, depending on where the type occurs

Total and OneToOne for types

$$[\![\lambda\ (A{:}\mathtt{Type}),\ (\forall\ x{:}A \qquad \ldots){:}\mathtt{Prop}]\!]$$

Total and OneToOne for types

$$[\![\lambda\ (A{:}\mathtt{Type}),\ (\forall\ \colorbox{yellow}{$x{:}A$}\qquad \ldots){:}\mathtt{Prop}]\!] :=$$

$$\lambda\ (A{:}\mathtt{Type})\ (A_2{:}\mathtt{Type})$$
$$\quad (A_r{:}\{A_r{:}\ A \to A_2 \to \mathtt{Type} \mid \colorbox{yellow}{$\mathrm{Total}\ A_r$}\}),\ \ldots$$

Total and OneToOne for types

$[\![\lambda\ (A{:}\mathtt{Type}),\ (\forall\ \boxed{x{:}A}\qquad \ldots){:}\mathtt{Prop}]\!]$

Total and OneToOne for types

$$[\![\lambda\ (A{:}\mathtt{Type}),\ (\forall\ x{:}A \times A,\ \ldots){:}\mathtt{Prop}]\!]$$

Total and OneToOne for types

$[\![\lambda\ (A{:}\mathtt{Type}),\ (\forall\ x{:}A\times A,\ \ldots){:}\mathtt{Prop}]\!]$

Total and OneToOne for composite types

Canonical types in Coq

- $\forall$
- inductives
- universes

$$
\begin{aligned}
&[\![\forall x{:}\, A.B]\!] := \\
&\quad \lambda(f{:}\forall x{:}A.B)(f_2{:}\forall x_2{:}A_2.B_2), \\
&\qquad \forall(x{:}A)(x_2{:}A_2)\ (x_r{:}[\![A]\!]\ x\ x_2), \quad [\![B]\!](f\ x)(f_2\ x_2)
\end{aligned}
$$

$\forall$: Total and OneToOne

$$[\![\forall x\!: \mathrm{A}.\mathrm{B}]\!] :=$$

$$\lambda(f\!:\!\forall x\!:\!\mathrm{A}.\mathrm{B})(f_2\!:\!\forall x_2\!:\!\mathrm{A}_2.\mathrm{B}_2),$$

$$\forall(x\!:\!\mathrm{A})(x_2\!:\!\mathrm{A}_2)\ (x_r\!:\![\![\mathrm{A}]\!]\ x\ x_2),\quad [\![\mathrm{B}]\!](f\ x)(f_2\ x_2)$$

$\forall$: Total and OneToOne

$$
\begin{aligned}
&[\![\forall x{:}\, A.B]\!] := \\
&\quad \lambda(f{:}\forall x{:}A.B)(f_2{:}\forall x_2{:}A_2.B_2), \\
&\qquad \forall(x{:}A)(x_2{:}A_2)\ (x_r{:}[\![A]\!]\ x\ x_2), \quad [\![B]\!](f\ x)(f_2\ x_2)
\end{aligned}
$$

Total

Total, OneToOne, proof irrelevant

$\forall$: Total and OneToOne

$$
\begin{aligned}
&[\![\forall x\!: A.B]\!] := \\
&\quad \lambda(f{:}\forall x{:}A.B)(f_2{:}\forall x_2{:}A_2.B_2), \\
&\qquad \forall(x{:}A)(x_2{:}A_2)\ (x_r{:}\underbrace{[\![A]\!]}_{\text{Total}}\ x\ x_2),\quad \underbrace{[\![B]\!]}_{\text{OneToOne}}(f\ x)(f_2\ x_2)
\end{aligned}
$$

function extensionality

W types: Total and OneToOne

```
Inductive IWT (I A : Type) (B : A → Type)
  (AI : A → I) (BI : ∀ (a : A), B a → I)
  : ∀ (i:I), Type :=
node : ∀ (a : A)
  (brs : ∀ b : B a, IWT I A B AI BI (BI a b)),
  IWT I A B AI BI (AI a).
```

W types: Total and OneToOne

OneToOne, irr

Total

Total, OneToOne, irr

```
Inductive IWT (I A : Type) (B : A → Type)
   (AI : A → I) (BI : ∀ (a : A), B a → I)
   : ∀ (i:I), Type :=
node : ∀ (a : A)
   (brs : ∀ b : B a, IWT I A B AI BI (BI a b)),
   IWT I A B AI BI (AI a).
```

W types: Total and OneToOne

OneToOne, irr

Total

$$
\begin{aligned}
&\texttt{Inductive}\ \mathsf{IWT}\ (I\ A : \texttt{Type})\ (B : A \to \texttt{Type})\\
&\quad (AI : A \to I)\ (BI : \forall\,(a : A),\ B\ a \to I)\\
&\quad : \forall\,(i{:}I),\ \texttt{Type} :=\\
&\mathsf{node} : \forall\,(a : A)\\
&\quad (brs : \forall\ b : B\ a,\ \mathsf{IWT}\ I\ A\ B\ AI\ BI\ (BI\ a\ b)),\\
&\quad \mathsf{IWT}\ I\ A\ B\ AI\ BI\ (AI\ a).
\end{aligned}
$$

function extensionality, proof irrelevance

OneToOne of $[\![\texttt{Type}_i]\!]$ is not provable

OneToOne of $[\![\texttt{Type}_i]\!]$ is not provable

$$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2:\texttt{Type}_j),$$
$$\{R\colon x \to x_2 \to \texttt{Type}_j\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$$

OneToOne of $[\![\texttt{Type}_i]\!]$ is not provable

$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),$
$\{R{:}\ x \to x_2 \to \texttt{Type}_j\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$

$\texttt{Type}_0{:}\texttt{Type}_1$

OneToOne of $[\![\texttt{Type}_i]\!]$ is not provable

$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),$
$\{R\colon x \to x_2 \to \texttt{Type}_j\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$

$\texttt{Type}_0{:}\texttt{Type}_1$

$[\![\texttt{Type}_0]\!]_{iso}$ nat nat
$[\![\texttt{Type}_0]\!]_{iso}$ nat (list unit)

OneToOne of $[\![\mathtt{Type}_i]\!]$ is not provable

$[\![\mathtt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\mathtt{Type}_j),$
$\{R{:}\ x \to x_2 \to \mathtt{Type}_j\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$

$\mathtt{Type}_0{:}\mathtt{Type}_1$

$[\![\mathtt{Type}_0]\!]_{iso}$ nat nat
$[\![\mathtt{Type}_0]\!]_{iso}$ nat (list unit)

Fail on Proposition mentioning higher universes

$$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$$
$$\{R{:}\ x \to x_2 \to \texttt{Set}\ \ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$$

$$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),\ x \to x_2 \to \texttt{Type}_j$$

Fail on Proposition mentioning higher universes

$$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$$
$$\{R{:}\ x \to x_2 \to \texttt{Set}\ \ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$$

$$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),\ x \to x_2 \to \texttt{Type}_j$$

Fail on Proposition mentioning higher universes

$$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$$
$$\{R{:}\ x \to x_2 \to\ \texttt{Prop}\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$$

$$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),\ x \to x_2 \to \texttt{Type}_j$$

Fail on Proposition mentioning higher universes

$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2\text{:}\texttt{Set}),$
$\{R\text{: } x \to x_2 \to\ \texttt{Prop}\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$

$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2\text{:}\texttt{Type}_j),\ x \to x_2 \to \texttt{Type}_j$

Implicity subtyping:
$(\lambda\ (x\text{:}\texttt{Type}_j)\text{,b})$ nat

Fail on Proposition mentioning higher universes

$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$
$\{R{:}\ x \to x_2 \to\ \texttt{Prop}\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$

$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),\ x \to x_2 \to \texttt{Type}_j$

Implicity subtyping:
$(\lambda\ (x{:}\texttt{Type}_j),\mathrm{b})\ \mathsf{nat}$
$(\lambda\ (x\ x_2{:}\texttt{Type}_j)\ x_r,\mathrm{b})\ \mathsf{nat}\ \mathsf{nat}\ [\![\mathsf{nat}]\!]_{iso}$

Fail on Proposition mentioning higher universes

$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$
$\{R{:}\ x \to x_2 \to\ \texttt{Prop}\ \&\ \mathrm{Total}\ R\ \&\ \mathrm{OneToOne}\ R\}$

$[\![\texttt{Type}_j]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Type}_j),\ x \to x_2 \to \texttt{Type}_j$

Implicity subtyping:
$(\lambda\ (x{:}\texttt{Type}_j),\mathrm{b})\ \mathsf{nat}$
$(\lambda\ (x\ x_2{:}\texttt{Type}_j)\ x_r,\mathrm{b})\ \mathsf{nat}\ \mathsf{nat}\ [\![\mathsf{nat}]\!]_{iso}$

Recap: Total and OneToOne for types

- $\forall$ ✓
- inductives ✓
- universes ✓
  - `Set` ✓
  - `Type` ✓

IsoRel translation

IsoRel translation

$$\llbracket \lambda x : A, B \rrbracket_{iso} :=$$
$$\lambda(x : A)(x_2 : A_2)(x_r : \quad \llbracket A \rrbracket x\, x_2), \llbracket B \rrbracket$$

IsoRel translation

$$\llbracket \lambda x : A, B \rrbracket_{iso} :=$$
$$\lambda (x : A)(x_2 : A_2)(x_r : \pi_1 \llbracket A \rrbracket x \, x_2), \llbracket B \rrbracket$$

MinIsoRel translation

$$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$$
$$\{R{:}\ x \to x_2 \to \texttt{Prop}\ \&\qquad\qquad\&\qquad\qquad\}$$

MinIsoRel translation

$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$
$\{R{:}\ x \to x_2 \to \texttt{Prop}\ \&\ \qquad\qquad \&\ \mathrm{OneToOne}\ R\}$

$\texttt{Definition}\ \mathrm{POne} :=$
$\lambda\ \texttt{(}T{:}\texttt{Set}\texttt{)}\ (f{:}\mathsf{nat}{\to}T\texttt{)}\texttt{,}\ f\ \mathsf{O} = f\ (\mathsf{S}\ \mathsf{O}).$

MinIsoRel translation

$$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$$
$$\{R{:}\ x \to x_2 \to \texttt{Prop}\ \&\ \mathrm{Total}\ R\ \& \qquad\qquad \}$$

$$\texttt{Definition}\ \mathrm{PTot} :=$$
$$\lambda\ \texttt{(}T{:}\mathsf{Set})\ (f{:}T{\to}\mathsf{nat}\texttt{)}\texttt{,}\ \forall\ (t{:}T),\ f\ t\ \texttt{=}\ \mathsf{O}.$$

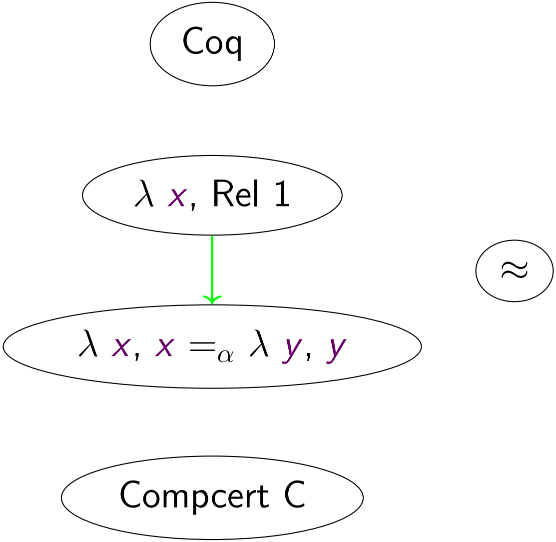
MinIsoRel translation

$$[\![\texttt{Set}]\!]_{iso} := \lambda\ (x\ x_2{:}\texttt{Set}),$$
$$\{R{:}\ x \to x_2 \to \texttt{Prop}\ \&\ \qquad\qquad \&\ \qquad\qquad\qquad \}$$

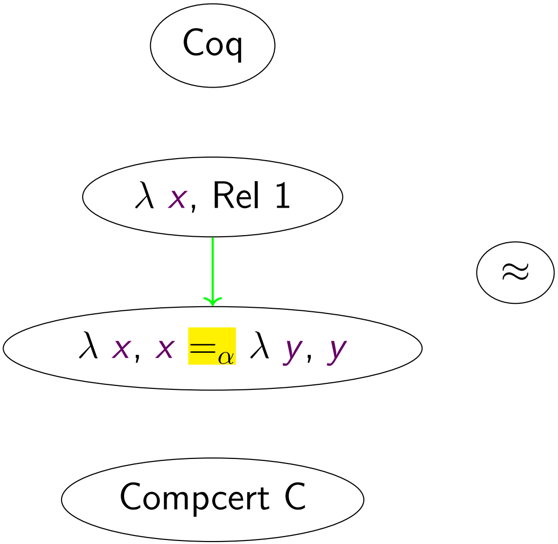
$\texttt{Definition}$ PNone :=

λ $\texttt{(}T{:}\texttt{Set}\texttt{)}$ $(f{:}T{\to}\mathsf{nat})$ $(a\ b\ {:}T\texttt{)}\ \texttt{,}\ \texttt{(}f\ a\ \texttt{=}\ f\ b\texttt{)}.$

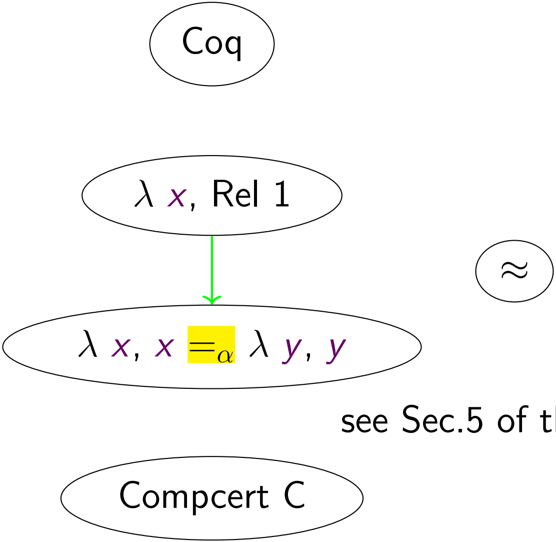
The example from CertiCoq, again



The example from CertiCoq, again



The example from CertiCoq, again



see Sec.5 of the paper

universe polymorphic inductives

$$\texttt{Inductive}\ \mathsf{list@}\{i\}\ (A : \mathit{Type@}\{i\}) : \mathit{Type@}\{i\}$$
$$:= \mathsf{nil} : \mathsf{list}\ A \mid \mathsf{cons} : A \to \mathsf{list}\ A \to \mathsf{list}\ A.$$

```
Inductive list@{i} (A : Type@{i}) : Type@{i}
:= nil : list A | cons : A → list A → list A.

 Fixpoint list_RF@{i} (A A₂: Type@{i})
   (A_R : A → A₂ → Type@{i} )
   (l : list@{i} A) (l₂ : list@{i} A₂): Type@{i} :=
match l, l₂ with
| nil,nil ⇒ True
| cons h tl, cons h₂ tl₂ ⇒
   (A_R h h₂ × list_RF A A₂ A_R tl tl₂)%type
| _,_ ⇒ False
end.
```

universe polymorphic inductives

```
Inductive list@{i} (A : Type@{i}) : Type@{i}
:= nil : list A | cons : A → list A → list A.

 Fixpoint list_RF@{i} (A A₂: Type@{i})
   (A_R : A → A₂ → Type@{i} )
   (l : list@{i} A) (l₂ : list@{i} A₂): Type@{i} :=
match l, l₂ with
| nil,nil ⇒ True
| cons h tl, cons h₂ tl₂ ⇒
   (A_R h h₂ × list_RF A A₂ A_R tl tl₂)%type
| _,_ ⇒ False
end.
```

```
Inductive list@{i} (A : Type@{i}) : Type@{i}
:= nil : list A | cons : A → list A → list A.

 Fixpoint list_RF@{i} (A A₂: Type@{i})
   (A_R : A → A₂ → if i=Set then Prop else i )
   (l : list@{i} A) (l₂ : list@{i} A₂): Type@{i} :=
match l, l₂ with
| nil,nil ⇒ True
| cons h tl, cons h₂ tl₂ ⇒
   (A_R h h₂ × list_RF A A₂ A_R tl tl₂)%type
| _,_ ⇒ False
end.
```

Wishlist/Questions

- explicit universe subtyping markers
- more expressive syntax for universe polymorphism
- registering translations
- better support for nested fixes