Strength of Weak Type Theory – Dutch Categories and Types Seminar

Effective conservativity of extensional type theory over weak type theory

Théo Winterhalter

joint work with Simon Boulier

Equality in type theory

Definitional

Objects are identified on the nose:

`vec A (2 + 3) ≡ vec A 5`

Proof simplification / witness property

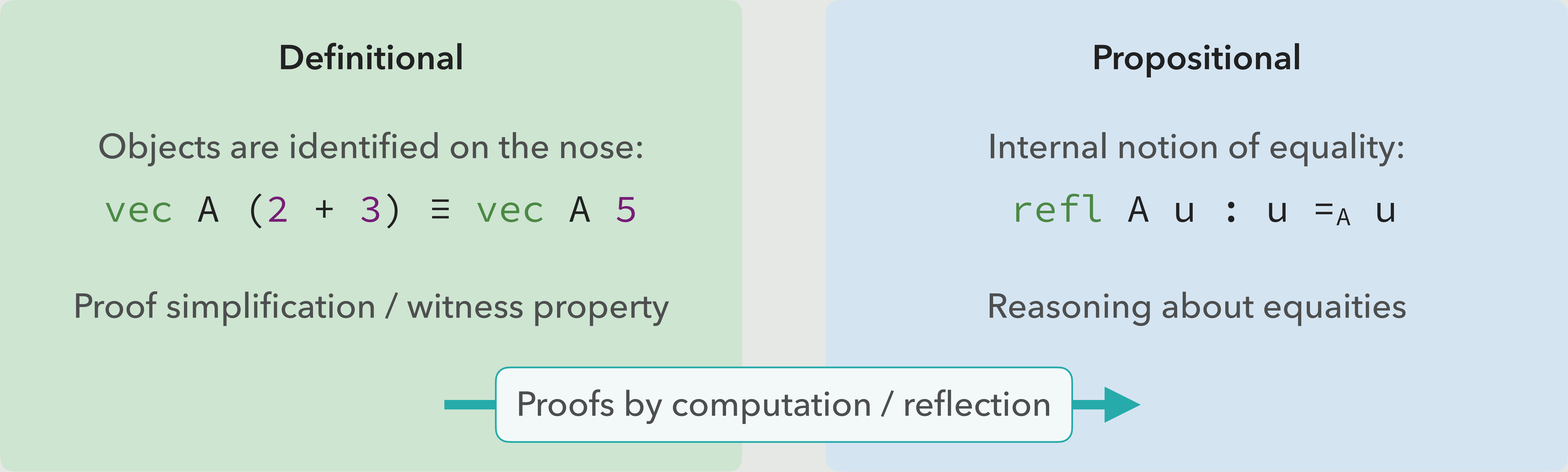
Propositional

Internal notion of equality:

$\texttt{refl A u : u} =_{A} \texttt{u}$

Reasoning about equaities

Equality in type theory



Definitional

Objects are identified on the nose:

$$\mathtt{vec}\ A\ (2 + 3) \equiv \mathtt{vec}\ A\ 5$$

Proof simplification / witness property

Propositional

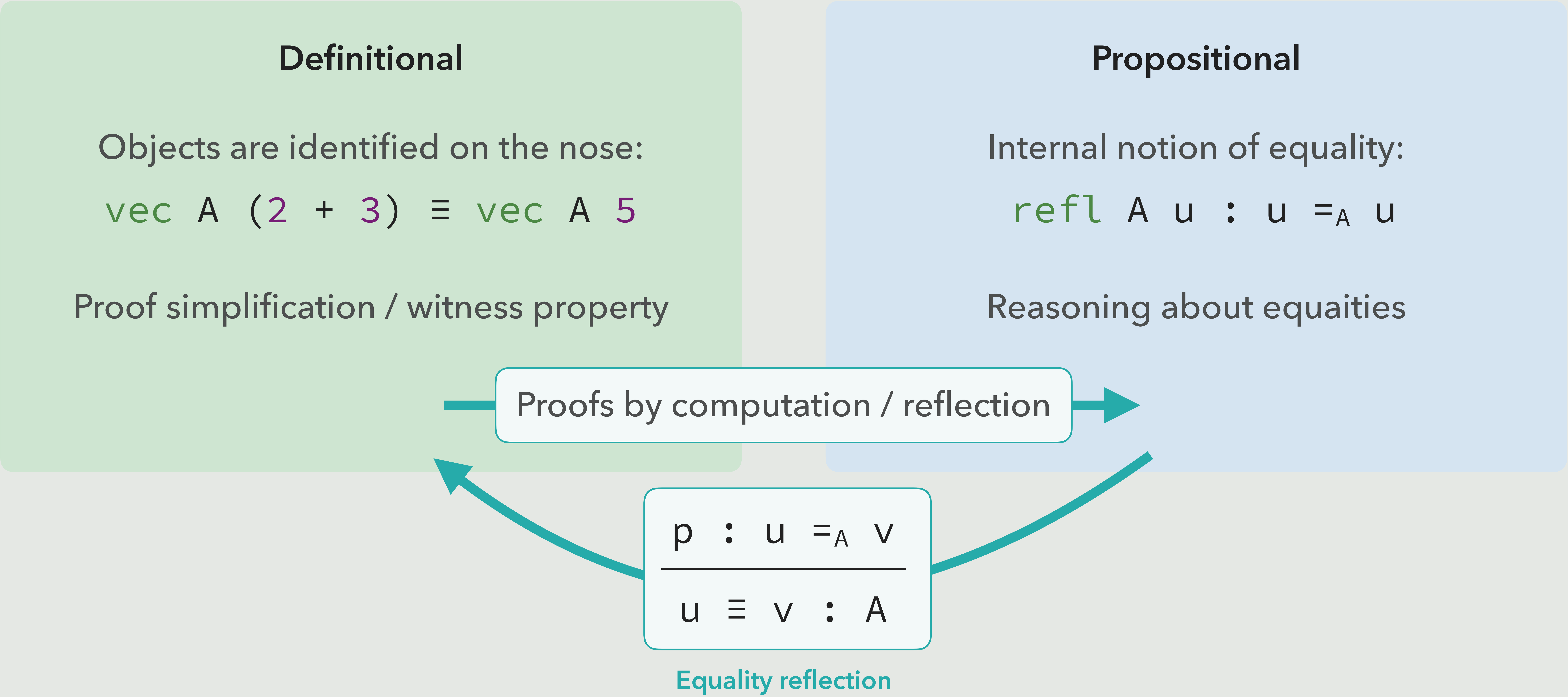
Internal notion of equality:

$$\mathtt{refl}\ A\ u : u =_A u$$

Reasoning about equaities

Proofs by computation / reflection

Equality in type theory



Definitional

Objects are identified on the nose:

`vec A (2 + 3) ≡ vec A 5`

Proof simplification / witness property

Propositional

Internal notion of equality:

$\mathtt{refl}\ A\ u : u =_A u$

Reasoning about equaities

Proofs by computation / reflection

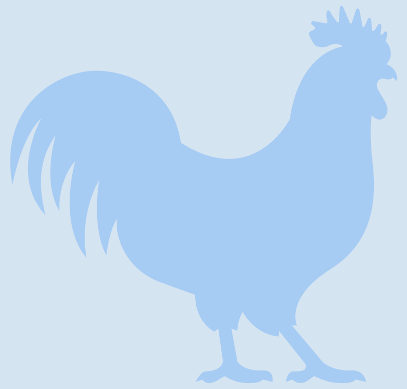
$$\frac{p : u =_A v}{u \equiv v : A}$$

Equality reflection

ETT

= ITT ⊕ Reflection

ITT



ETT

= ITT ⊕ Reflection

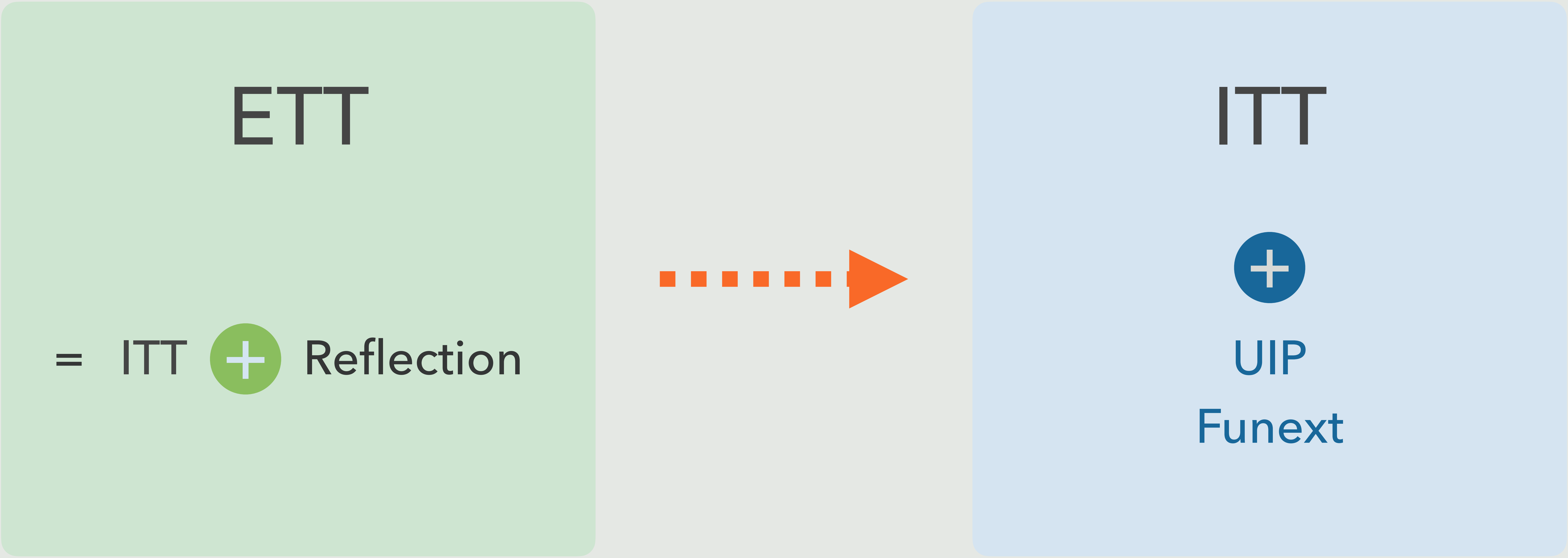
ITT

⊕

UIP

Funext

Martin Hofmann (1995): ETT is conservative over ITT (categorically)



ETT

= ITT + Reflection

ITT

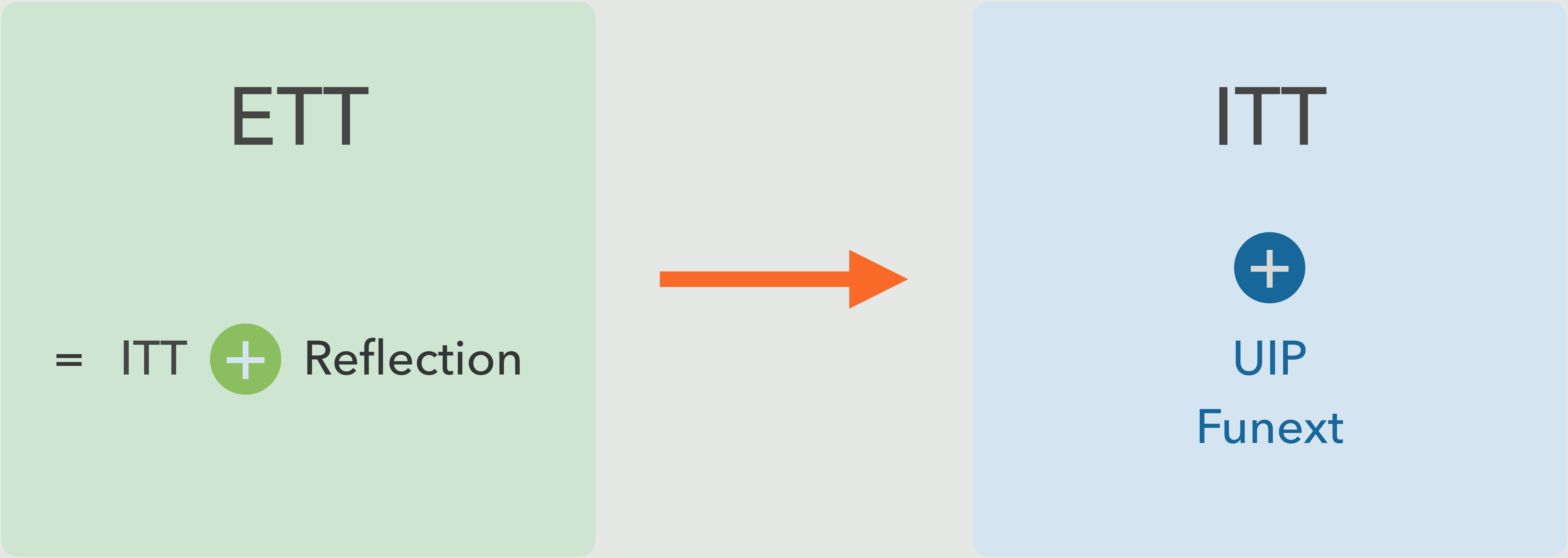
+

UIP

Funext

Nicolas Oury (2005): conservative translation (on paper)

⚠ ITT + congruence of application for heterogenous equality



ETT

= ITT + Reflection

ITT

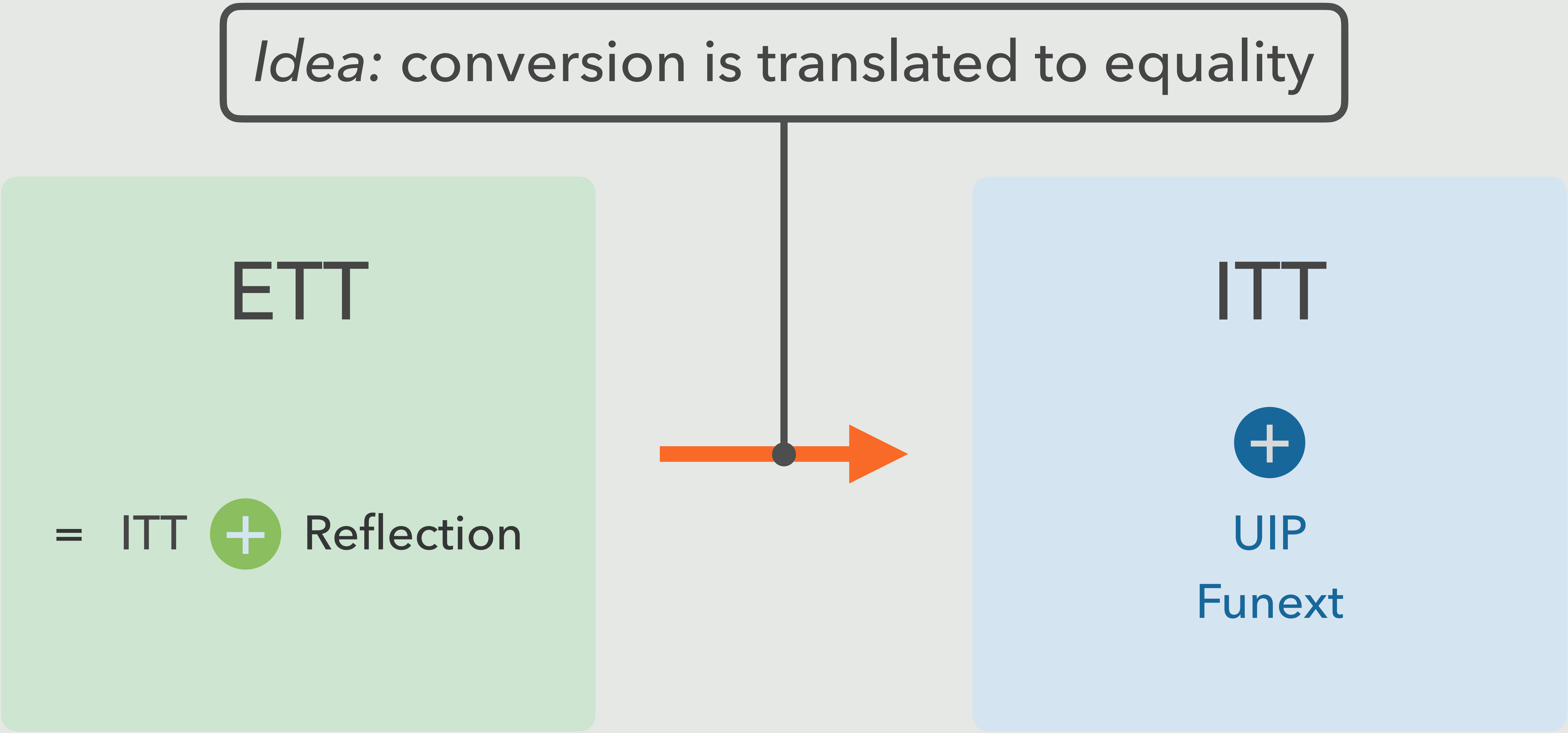
+

UIP

Funext

with Nicolas Tabareau and Matthieu Sozeau (2019):
conservative translation in Coq

no extra axiom needed!



Idea: conversion is translated to equality

ETT

= ITT + Reflection

ITT

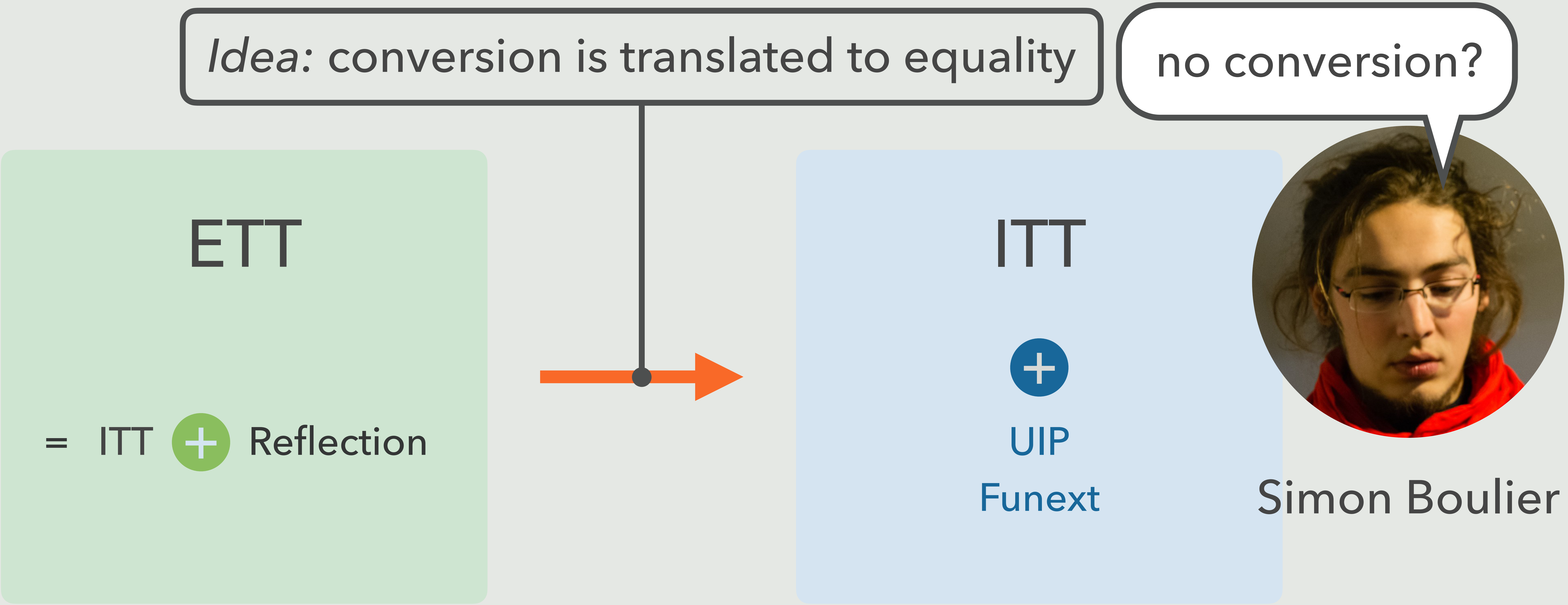
+

UIP

Funext

with Nicolas Tabareau and Matthieu Sozeau (2019): conservative translation in Coq

no extra axiom needed!



Idea: conversion is translated to equality

no conversion?

ETT

= ITT + Reflection

ITT

+

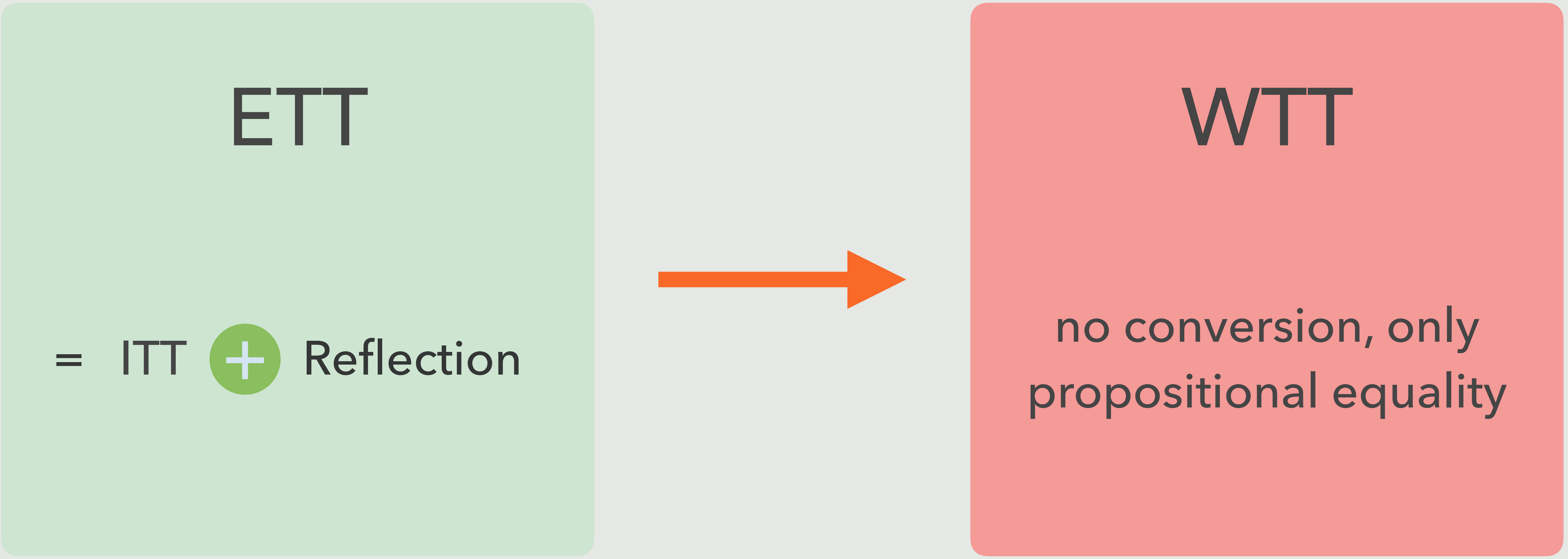
UIP

Funext

Simon Boulier

with Nicolas Tabareau and Matthieu Sozeau (2019): conservative translation in Coq

no extra axiom needed!



ETT

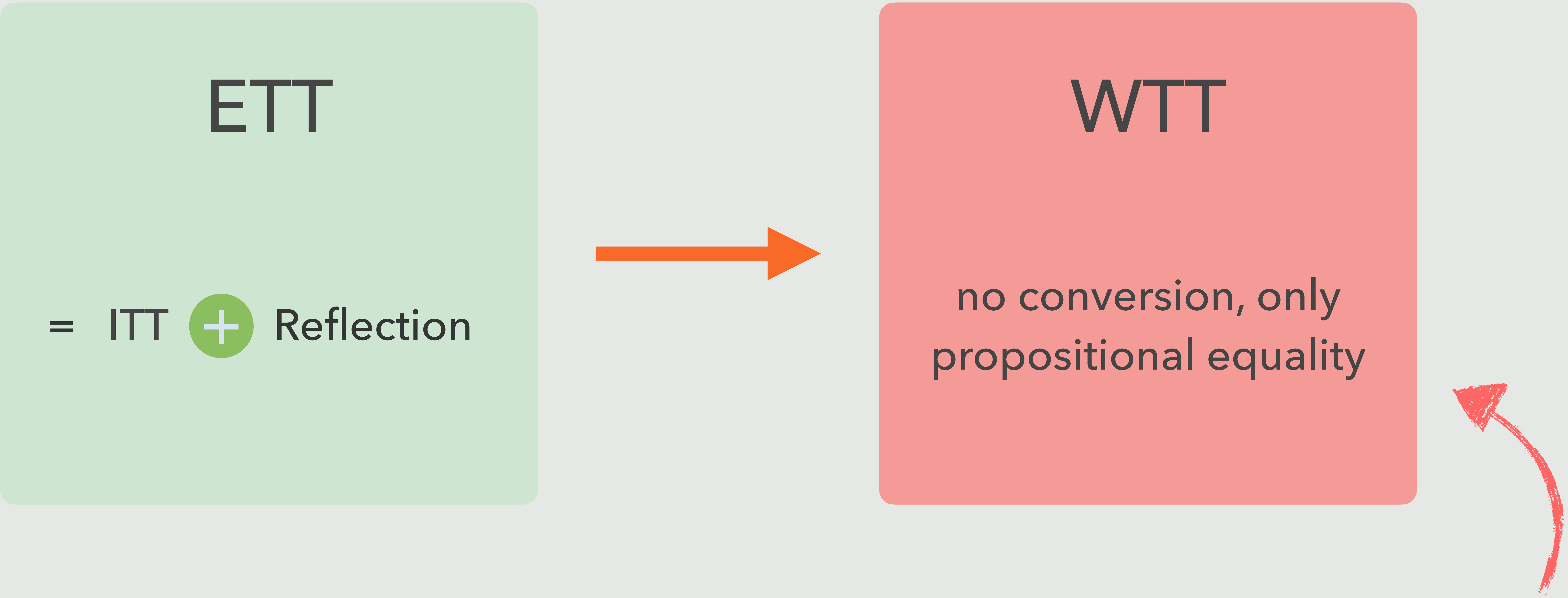
= ITT + Reflection

WTT

no conversion, only propositional equality

with Simon Boulier (2019):
conservative translation over WTT in Coq

Coq proof becomes much simpler!



ETT

= ITT + Reflection

WTT

no conversion, only propositional equality

what is the correct design?

with Simon Boulier (2019):
conservative translation over WTT in Coq

Coq proof becomes much simpler!

Idea of the translation

We want $[\![.]\!]$ such that

$\Gamma \vdash t : A$ implies $[\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!]$

$\Gamma \vdash u \equiv v : A$ implies $[\![\Gamma]\!] \vdash p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$

for some p

Idea of the translation

We want $[\![.]\!]$ such that

$$\Gamma \vdash_{x} t : A \quad \text{implies} \quad [\![\Gamma]\!] \vdash_{w} [\![t]\!] : [\![A]\!]$$

$$\Gamma \vdash_{x} u \equiv v : A \quad \text{implies} \quad [\![\Gamma]\!] \vdash_{w} p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$$

Idea of the translation

We want $[\![.]\!]$ such that

$\Gamma \vdash_{\mathtt{x}} t : A$ implies $[\![\Gamma]\!] \vdash_{\mathtt{w}} [\![t]\!] : [\![A]\!]$

$\Gamma \vdash_{\mathtt{x}} u \equiv v : A$ implies $[\![\Gamma]\!] \vdash_{\mathtt{w}} p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$

Let's look at the *conversion* rule

$$\frac{\Gamma \vdash_{x} t : A \qquad \Gamma \vdash_{x} A \equiv B}{\Gamma \vdash_{x} t : B}$$

Idea of the translation

We want $[\![.]\!]$ such that

$\Gamma \vdash_{\mathtt{x}} t : A$ implies $[\![\Gamma]\!] \vdash_{\mathtt{w}} [\![t]\!] : [\![A]\!]$

$\Gamma \vdash_{\mathtt{x}} u \equiv v : A$ implies $[\![\Gamma]\!] \vdash_{\mathtt{w}} p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$

Let's look at the *conversion* rule

$$\frac{\Gamma \vdash_{\mathtt{x}} t : A \qquad \Gamma \vdash_{\mathtt{x}} A \equiv B}{\Gamma \vdash_{\mathtt{x}} t : B} \dashrightarrow \qquad [\![\Gamma]\!] \vdash_{\mathtt{w}} [\![t]\!] : [\![A]\!]$$

Idea of the translation

We want $\llbracket . \rrbracket$ such that

$$\begin{array}{lll} \Gamma \vdash_{\mathbf{x}} t : A & \text{implies} & \llbracket \Gamma \rrbracket \vdash_{\mathbf{w}} \llbracket t \rrbracket : \llbracket A \rrbracket \\ \Gamma \vdash_{\mathbf{x}} u \equiv v : A & \text{implies} & \llbracket \Gamma \rrbracket \vdash_{\mathbf{w}} p : \llbracket u \rrbracket =_{\llbracket A \rrbracket} \llbracket v \rrbracket \end{array}$$

Let's look at the *conversion* rule

$$\frac{\Gamma \vdash_{x} t : A \qquad \Gamma \vdash_{x} A \equiv B}{\Gamma \vdash_{x} t : B} \quad \dashrightarrow \quad \begin{array}{l} \llbracket \Gamma \rrbracket \vdash_{w} \llbracket t \rrbracket : \llbracket A \rrbracket \\ \llbracket \Gamma \rrbracket \vdash_{w} p : \llbracket A \rrbracket =_{\llbracket \mathtt{Type} \rrbracket} \llbracket B \rrbracket \end{array}$$

Idea of the translation

We want $[\![.]\!]$ such that

$\Gamma \vdash_x t : A$ implies $[\![\Gamma]\!] \vdash_w [\![t]\!] : [\![A]\!]$

$\Gamma \vdash_x u \equiv v : A$ implies $[\![\Gamma]\!] \vdash_w p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$

Let's look at the *conversion* rule

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B} \quad \dashrightarrow \quad \begin{array}{l} [\![\Gamma]\!] \vdash_w [\![t]\!] : [\![A]\!] \\ [\![\Gamma]\!] \vdash_w p : [\![A]\!] =_{[\![\mathtt{Type}]\!]} [\![B]\!] \end{array}$$

so $[\![\Gamma]\!] \vdash_w \mathtt{transp}(p, [\![t]\!]) : [\![B]\!]$

Idea of the translation

We want $[\![\,.\,]\!]$ such that

$\Gamma \vdash_x t : A$ implies $[\![\Gamma]\!] \vdash_w [\![t]\!] : [\![A]\!]$

$\Gamma \vdash_x u \equiv v : A$ implies $[\![\Gamma]\!] \vdash_w p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$

Let's look at the *conversion* rule

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B}$$

⇢

$[\![\Gamma]\!] \vdash_w [\![t]\!] : [\![A]\!]$

$[\![\Gamma]\!] \vdash_w p : [\![A]\!] =_{[\![\mathtt{Type}]\!]} [\![B]\!]$

so $[\![\Gamma]\!] \vdash_w \underline{\mathtt{transp}(p, [\![t]\!])} : [\![B]\!]$

We wanted $[\![t]\!]$!

Idea of the translation

We want $[\![\,.\,]\!]$ such that

~~$\Gamma \vdash_x t : A$ implies $[\![\Gamma]\!] \vdash_w [\![t]\!] : [\![A]\!]$~~

~~$\Gamma \vdash_x u \equiv v : A$ implies $[\![\Gamma]\!] \vdash_w p : [\![u]\!] =_{[\![A]\!]} [\![v]\!]$~~

💡 $[\![t]\!]$ should be a *class of terms* with

$t' \in [\![t]\!]$ implies $\mathtt{transp}(p,t') \in [\![t]\!]$

Idea of the translation

We want $[\![\,.\,]\!]$ such that

$\Gamma \vdash_x t : A$ implies $\Gamma' \vdash_w t' : A'$

$\Gamma \vdash_x u \equiv v : A$ implies $\Gamma' \vdash_w p : u' =_{A'} v'$

for $\Gamma' \in [\![\Gamma]\!]$, $A' \in [\![A]\!]$, …

💡 $[\![t]\!]$ should be a *class of terms* with

$t' \in [\![t]\!]$ implies $\mathtt{transp}(p,t') \in [\![t]\!]$

Idea of the translation

We want $[\![\,.\,]\!]$ such that

$$\Gamma \vdash_x t : A \qquad \text{implies} \qquad \Gamma' \vdash_w t' : A'$$

$$\Gamma \vdash_x u \equiv v : A \qquad \text{implies} \qquad \Gamma' \vdash_w p : u' =_{A'} v'$$

for $\Gamma' \in [\![\Gamma]\!]$, $A' \in [\![A]\!]$, …

conversion rule again:

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B} \quad \dashrightarrow \quad \begin{array}{l} \underline{\Gamma'} \vdash_w t' : \underline{A'} \\ \underline{\Gamma''} \vdash_w p : \underline{A''} =_{T'} B' \end{array}$$

Idea of the translation

We want $[\![\,.\,]\!]$ such that

$$\Gamma \vdash_x t : A \qquad \text{implies} \qquad \Gamma' \vdash_w t' : A'$$

$$\Gamma \vdash_x u \equiv v : A \qquad \text{implies} \qquad \Gamma' \vdash_w p : u' =_{A'} v'$$

for $\Gamma' \in [\![\Gamma]\!]$, $A' \in [\![A]\!]$, …

conversion rule again:

$T' \in [\![\text{Type}]\!]$

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B} \quad \dashrightarrow \quad \begin{array}{l} \underline{\Gamma'} \vdash_w t' : \underline{A'} \\ \underline{\Gamma''} \vdash_w p : \underline{A''} =_{T'} B' \end{array}$$

Idea of the translation

We want $\llbracket . \rrbracket$ such that

$$\Gamma \vdash_x t : A \quad \text{implies} \quad \Gamma' \vdash_w t' : A'$$

$$\Gamma \vdash_x u \equiv v : A \quad \text{implies} \quad \Gamma' \vdash_w p : u' =_{A'} v'$$

for $\Gamma' \in \llbracket \Gamma \rrbracket$, $A' \in \llbracket A \rrbracket$, …

conversion rule again:

$T' \in \llbracket \mathtt{Type} \rrbracket$

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B} \quad \dashrightarrow \quad \begin{array}{l} \underline{\Gamma'} \vdash_w t' : \underline{A'} \\ \underline{\Gamma''} \vdash_w p : \underline{A''} =_{T'} B' \end{array}$$

we need to relate two translations of the same object (at possibly two different types)

Heterogenous equality

Axiomatically

$$\frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B}{\Gamma \vdash u \ {}_{A}{=}_{B}\ v : \mathsf{Type}}$$

Heterogenous equality

Axiomatically

$$\frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B}{\Gamma \vdash u \ {}_{A}{=}_{B}\ v : \mathsf{Type}}$$

with some constructions

$$\mathtt{hrefl}\ A\ u : u\ {}_{A}{=}_{A}\ u$$

Heterogenous equality

Axiomatically

$$\frac{\Gamma \vdash u : A \qquad\qquad \Gamma \vdash v : B}{\Gamma \vdash u \; {}_{A}{=}_{B} \; v : \mathtt{Type}}$$

with some constructions

$$\mathtt{hrefl}\ A\ u : u \; {}_{A}{=}_{A} \; u$$

$$\mathtt{eq\_to\_heq}\ (p : u =_{A} v) : u \; {}_{A}{=}_{A} \; v$$

Heterogenous equality

Axiomatically

$$\frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B}{\Gamma \vdash u \ {}_{A}{=}_{B}\ v : \mathsf{Type}}$$

with some constructions

$\mathtt{hrefl}\ A\ u : u\ {}_{A}{=}_{A}\ u$

$\mathtt{eq\_to\_heq}\ (p : u =_{A} v) : u\ {}_{A}{=}_{A}\ v$

$\mathtt{heq\_to\_eq}\ (p : u\ {}_{A}{=}_{A}\ v) : u =_{A} v$

Heterogenous equality

Axiomatically

$$\frac{\Gamma \vdash u : A \qquad\qquad \Gamma \vdash v : B}{\Gamma \vdash u \;{}_{A}{=}_{B}\; v : \mathtt{Type}}$$

with some constructions

```
hrefl A u : u A=A u
```

$\mathtt{hrefl}\ A\ u : u\ {}_{A}{=}_{A}\ u$

$\mathtt{eq\_to\_heq}\ (p : u =_{A} v) : u\ {}_{A}{=}_{A}\ v$

$\mathtt{heq\_to\_eq}\ (p : u\ {}_{A}{=}_{A}\ v) : u =_{A} v$

$\mathtt{heq\_transp}\ (p : A = B)\ (t : A) : t\ {}_{A}{=}_{B}\ \mathtt{transp}(p,t)$

Heterogenous equality

ITT realisation

$$a \; {}_{A}{=}_{B} \; b \coloneqq \sum (p : A =_{\mathtt{Type}} B).\ \mathtt{transp}(p,a) =_{B} b$$

with some **provable** constructions

$$\mathtt{hrefl}\ A\ u : u \; {}_{A}{=}_{A} \; u$$

$$\mathtt{eq\_to\_heq}\ (p : u =_{A} v) : u \; {}_{A}{=}_{A} \; v$$

$$\mathtt{heq\_to\_eq}\ (p : u \; {}_{A}{=}_{A} \; v) : u =_{A} v$$

$$\mathtt{heq\_transp}\ (p : A = B)\ (t : A) : t \; {}_{A}{=}_{B} \; \mathtt{transp}(p,t)$$

Heterogenous equality

ITT realisation

$$a \;{}_{A}{=}_{B}\; b \coloneqq \sum (p : A =_{\mathsf{Type}} B).\ \mathtt{transp}(p,a) =_{B} b$$

with some **provable** constructions

`hrefl A u : u` $_{A}{=}_{A}$ `u`

`eq_to_heq (p : u` $=_{A}$ `v) : u` $_{A}{=}_{A}$ `v`

using UIP!

`heq_to_eq (p : u` $_{A}{=}_{A}$ `v) : u` $=_{A}$ `v`

`heq_transp (p : A = B) (t : A) : t` $_{A}{=}_{B}$ `transp(p,t)`

Heterogenous equality

ITT realisation

$$a \;{}_{A}{=}_{B}\; b \coloneqq \sum\,(p : A =_{\mathrm{Type}} B).\ \mathtt{transp}(p,a) =_{B} b$$

with some **provable** constructions

$\mathtt{hrefl}\ A\ u : u \;{}_{A}{=}_{A}\; u$

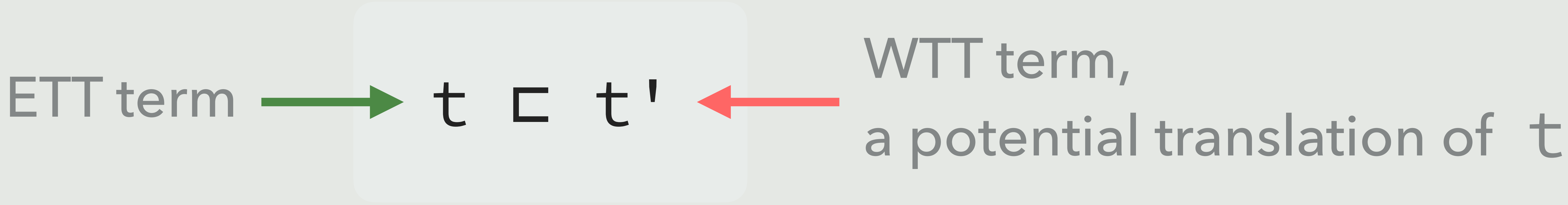
$\mathtt{eq\_to\_heq}\ (p : u =_{A} v) : u \;{}_{A}{=}_{A}\; v$

using UIP!

$\mathtt{heq\_to\_eq}\ (p : u \;{}_{A}{=}_{A}\; v) : u =_{A} v$

$\mathtt{heq\_transp}\ (p : A = B)\ (t : A) : t \;{}_{A}{=}_{B}\; \mathtt{transp}(p,t)$

Terms up to transport



ETT term

`t ⊏ t'`

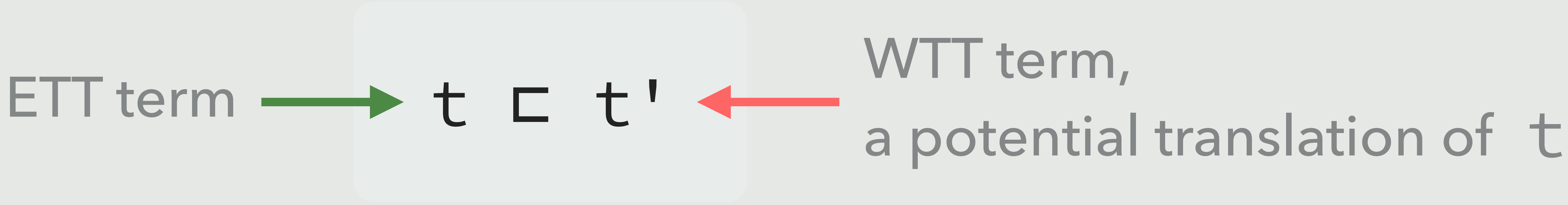
WTT term,
a potential translation of `t`

Terms up to transport

ETT term → `t ⊏ t'` ← WTT term, a potential translation of `t`

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$$

Terms up to transport



ETT term

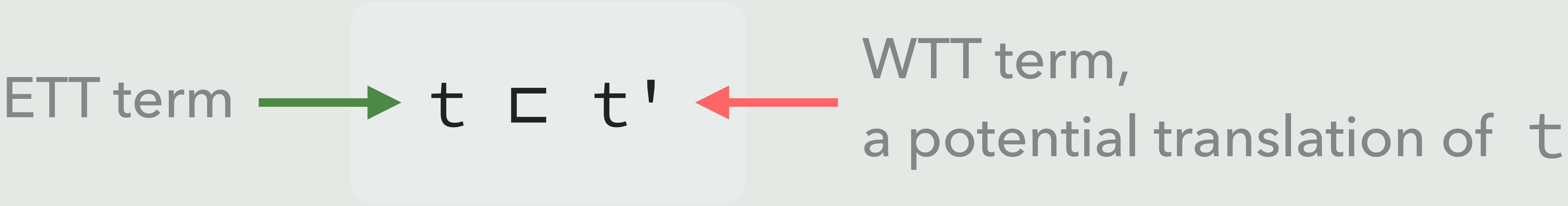
$t \sqsubset t'$

WTT term,
a potential translation of t

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p, t')}$$

$$\frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u\ @^{(x:A).B}\ v \sqsubset u'\ @^{(x:A').B'}\ v}$$

Terms up to transport



ETT term

$t \sqsubset t'$

WTT term,
a potential translation of t

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p, t')}$$

$$\frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u \ @^{(x:A).B}\ v \sqsubset u' \ @^{(x:A').B'}\ v}$$

...

Terms up to transport

ETT term → $t \sqsubset t'$ ← WTT term, a potential translation of t

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p, t')} \qquad \frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u \; @^{(x:A).B} \; v \sqsubset u' \; @^{(x:A').B'} \; v} \qquad \ldots$$

Fundamental lemma

Given Γ and $t_0 \sqsupset\!.\!\sqsubset t_1$, there exists a term p such that if $\Gamma \vdash_w t_0 : A$ and $\Gamma \vdash_w t_1 : B$ then $\Gamma \vdash_w p : t_0 \; {}_{A}{=}_{B} \; t_1$.

Terms up to transport

using
heq_transp

$$\frac{\mathtt{t} \sqsubset \mathtt{t'}}{\mathtt{t} \sqsubset \mathtt{transp(p,t')}}$$

$$\frac{\mathtt{u} \sqsubset \mathtt{u'} \quad \mathtt{A} \sqsubset \mathtt{A'} \quad \mathtt{B} \sqsubset \mathtt{B'} \quad \mathtt{v} \sqsubset \mathtt{v'}}{\mathtt{u}\ @^{\mathtt{(x:A).B}}\ \mathtt{v} \sqsubset \mathtt{u'}\ @^{\mathtt{(x:A').B'}}\ \mathtt{v}}$$

...

Fundamental lemma

Given Γ and $t_0 \sqsupset.\!\sqsubset t_1$, there exists a term p such that
if Γ $\vdash_w t_0 : A$ and Γ $\vdash_w t_1 : B$ then Γ $\vdash_w p : t_0 \ {}_A{=}_B \ t_1$.

Terms up to transport

using
heq_transp

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$$

using
cong_app

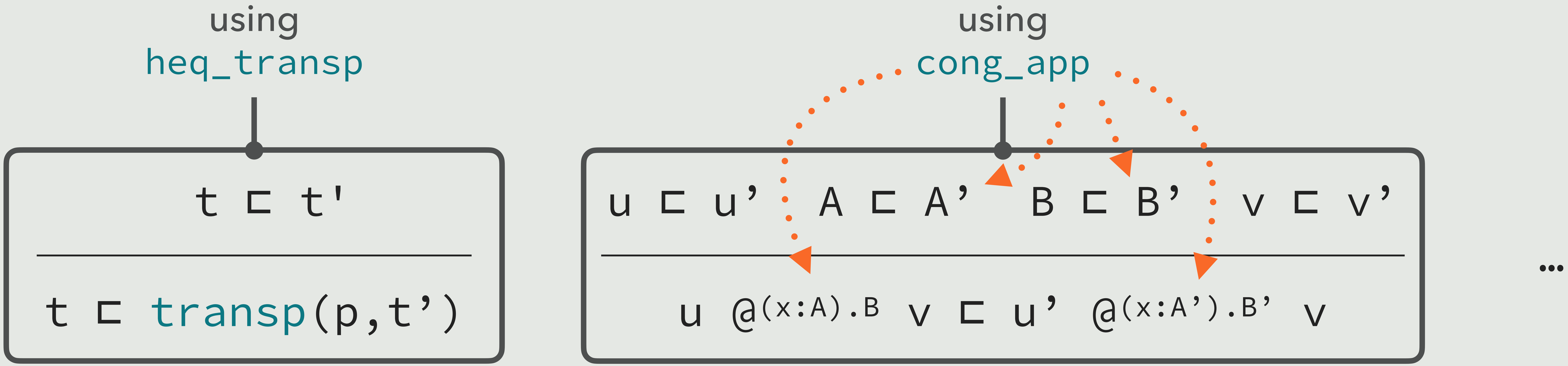
$$\frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u \ @^{(x:A).B}\ v \sqsubset u' \ @^{(x:A').B'}\ v}$$

…

Fundamental lemma

Given Γ and $t_0 \sqsupset.\!\sqsubset t_1$, there exists a term p such that if $\Gamma \vdash_w t_0 : A$ and $\Gamma \vdash_w t_1 : B$ then $\Gamma \vdash_w p : t_0 \ {}_A{=}_B \ t_1$.

Terms up to transport



using heq_transp

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$$

using cong_app

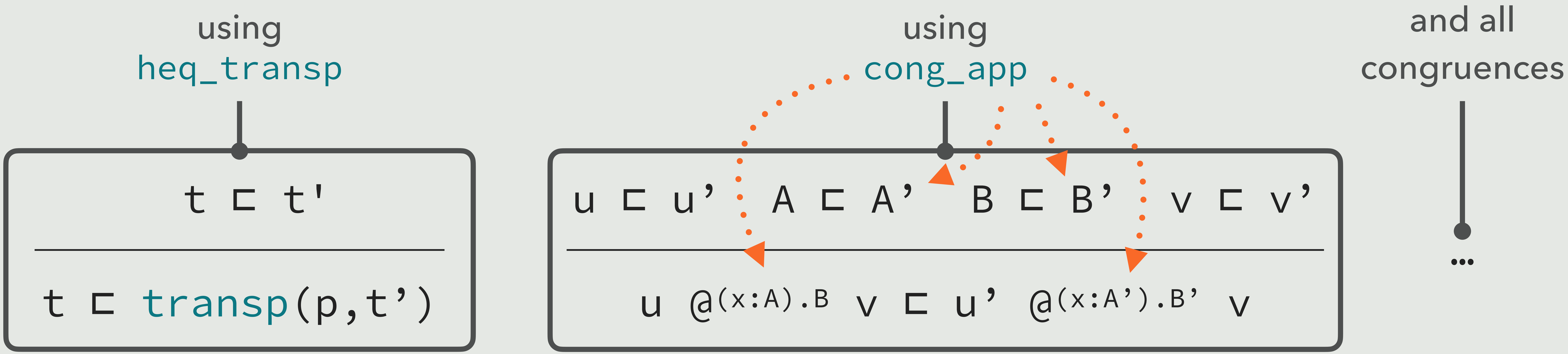
$$\frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u \; @^{(x:A).B} \; v \sqsubset u' \; @^{(x:A').B'} \; v}$$

...

Fundamental lemma

Given Γ and $t_0 \sqsupset.\sqsubset t_1$, there exists a term p such that if $\Gamma \vdash_w t_0 : A$ and $\Gamma \vdash_w t_1 : B$ then $\Gamma \vdash_w p : t_0 \; {}_A{=}_B \; t_1$.

Terms up to transport



using heq_transp

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$$

using cong_app

$$\frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u \ @^{(x:A).B}\ v \sqsubset u' \ @^{(x:A').B'}\ v}$$

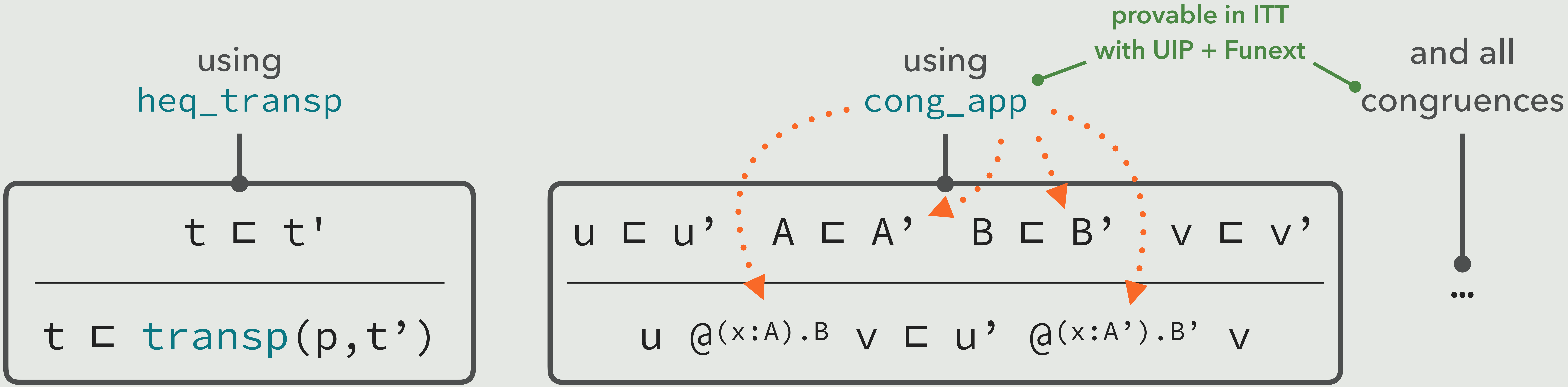
and all congruences

...

Fundamental lemma

Given Γ and $t_0 \sqsupset.\sqsubset t_1$, there exists a term p such that if $\Gamma \vdash_w t_0 : A$ and $\Gamma \vdash_w t_1 : B$ then $\Gamma \vdash_w p : t_0 \ {}_A{=}_B\ t_1$.

Terms up to transport



using heq_transp

$$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$$

using cong_app

provable in ITT with UIP + Funext

$$\frac{u \sqsubset u' \quad A \sqsubset A' \quad B \sqsubset B' \quad v \sqsubset v'}{u \ @^{(x:A).B}\ v \sqsubset u' \ @^{(x:A').B'}\ v}$$

and all congruences

...

Fundamental lemma

Given Γ and $t_0 \sqsupset.\sqsubset t_1$, there exists a term p such that if $\Gamma \vdash_w t_0 : A$ and $\Gamma \vdash_w t_1 : B$ then $\Gamma \vdash_w p : t_0 \ {}_A{=}_B\ t_1$.

Translations of judgments

Sets of valid judgements (with derivations)

$\vdash_{w}\ \Gamma' \ \in\ [\![\ \vdash_{x}\ \Gamma\ ]\!]$ **when** $\Gamma\ \sqsubset\ \Gamma'$

Translations of judgments

Sets of valid judgements (with derivations)

$\vdash_w \Gamma' \in \llbracket \vdash_x \Gamma \rrbracket$ when $\Gamma \sqsubset \Gamma'$

$\Gamma' \vdash_w t' : A' \in \llbracket \Gamma \vdash_x t : A \rrbracket$ when $\Gamma \sqsubset \Gamma', t \sqsubset t', A \sqsubset A'$

Translations of judgments

Sets of valid judgements (with derivations)

$\vdash_w \Gamma' \in \llbracket \vdash_x \Gamma \rrbracket$ when $\Gamma \sqsubset \Gamma'$

$\Gamma' \vdash_w t' : A' \in \llbracket \Gamma \vdash_x t : A \rrbracket$ when $\Gamma \sqsubset \Gamma', t \sqsubset t', A \sqsubset A'$

$\Gamma' \vdash_w p : u' {}_{A'=A''} v' \in$
$\llbracket \Gamma \vdash_x u \equiv v : A \rrbracket$ when $\Gamma \sqsubset \Gamma', A \sqsubset A', A \sqsubset A'',$
$u \sqsubset u', v \sqsubset v'$

Translations of judgments

Sets of valid judgements (with derivations)

$\vdash_w \Gamma' \in [\![\vdash_x \Gamma]\!]$ when $\Gamma \sqsubset \Gamma'$

$\Gamma' \vdash_w t' : A' \in [\![\Gamma \vdash_x t : A]\!]$ when $\Gamma \sqsubset \Gamma', t \sqsubset t', A \sqsubset A'$

$\Gamma' \vdash_w p : u' \,{}_{A'=A''}\, v' \in [\![\Gamma \vdash_x u \equiv v : A]\!]$ when $\Gamma \sqsubset \Gamma', A \sqsubset A', A \sqsubset A'', u \sqsubset u', v \sqsubset v'$

Translation theorem

If $\vdash_x \Gamma$ then there exists $\vdash_w \Gamma' \in [\![\vdash_x \Gamma]\!]$

Translations of judgments

Sets of valid judgements (with derivations)

$\vdash_{w} \Gamma' \in [\![\vdash_{x} \Gamma]\!]$ when $\Gamma \sqsubset \Gamma'$

$\Gamma' \vdash_{w} t' : A' \in [\![\Gamma \vdash_{x} t : A]\!]$ when $\Gamma \sqsubset \Gamma',\ t \sqsubset t',\ A \sqsubset A'$

$\Gamma' \vdash_{w} p : u' \ {}_{A'=A''}\ v' \in [\![\Gamma \vdash_{x} u \equiv v : A]\!]$ when $\Gamma \sqsubset \Gamma',\ A \sqsubset A',\ A \sqsubset A'',\ u \sqsubset u',\ v \sqsubset v'$

Translation theorem

If $\vdash_{x} \Gamma$ then there exists $\vdash_{w} \Gamma' \in [\![\vdash_{x} \Gamma]\!]$

If $\Gamma \vdash_{x} t : A$ then for any $\vdash_{w} \Gamma' \in [\![\vdash_{x} \Gamma]\!]$,
there exists $\Gamma' \vdash_{w} t' : A' \in [\![\Gamma \vdash_{x} t : A]\!]$

Translations of judgments

Sets of valid judgements (with derivations)

$$\vdash_w \Gamma' \in [\![\vdash_x \Gamma]\!] \quad \text{when} \quad \Gamma \sqsubset \Gamma'$$

$$\Gamma' \vdash_w t' : A' \in [\![\Gamma \vdash_x t : A]\!] \quad \text{when} \quad \Gamma \sqsubset \Gamma',\ t \sqsubset t',\ A \sqsubset A'$$

$$\Gamma' \vdash_w p : u' \,{}_{A'}{=}_{A''}\, v' \in [\![\Gamma \vdash_x u \equiv v : A]\!] \quad \text{when} \quad \Gamma \sqsubset \Gamma',\ A \sqsubset A',\ A \sqsubset A'',\ u \sqsubset u',\ v \sqsubset v'$$

Translation theorem

If $\vdash_x \Gamma$ then there exists $\vdash_w \Gamma' \in [\![\vdash_x \Gamma]\!]$

If $\Gamma \vdash_x t : A$ then for any $\vdash_w \Gamma' \in [\![\vdash_x \Gamma]\!]$,
there exists $\Gamma' \vdash_w t' : A' \in [\![\Gamma \vdash_x t : A]\!]$

If $\Gamma \vdash_x u \equiv v : A$ then for any $\vdash_w \Gamma' \in [\![\vdash_x \Gamma]\!]$,
there exists $\Gamma' \vdash_w p : u' \,{}_{A'}{=}_{A''}\, v' \in [\![\Gamma \vdash_x u \equiv v : A]\!]$

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_w t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_w t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_w t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_w t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Proof

$A \to B \sqsubset C'$

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_w t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_w t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Proof

$A \to B \sqsubset C'$ is obtained from a certain number of applications of $$\frac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$$

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_w t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_w t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Proof

$A \to B \sqsubset C'$ is obtained from a certain number of applications of $\dfrac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$

followed by $\dfrac{A \sqsubset A' \qquad B \sqsubset B'}{A \to B \sqsubset A' \to B'}$

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_{w} t' : C' \in [\![\Gamma \vdash_{x} t : A \to B]\!]$,
there exists $\Gamma' \vdash_{w} t'' : A' \to B' \in [\![\Gamma \vdash_{x} t : A \to B]\!]$

* actually applies to any type former

Proof

$A \to B \sqsubset C'$ is obtained from a certain number of applications of $\dfrac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$

followed by $\dfrac{A \sqsubset A' \qquad B \sqsubset B'}{A \to B \sqsubset A' \to B'}$ we thus have $A' \to B' \sqsupset A \to B \sqsubset C'$

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_w t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_w t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Proof

$A \to B \sqsubset C'$ is obtained from a certain number of applications of $\dfrac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$

followed by $\dfrac{A \sqsubset A' \qquad B \sqsubset B'}{A \to B \sqsubset A' \to B'}$ we thus have $A' \to B' \sqsupset A \to B \sqsubset C'$

by the fundamental lemma, they are heterogeneously equal, and by heq_to_eq they are equal: $e : C' = A' \to B'$

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_w t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_w t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Proof

$A \to B \sqsubset C'$ is obtained from a certain number of applications of $\dfrac{t \sqsubset t'}{t \sqsubset \mathtt{transp}(p,t')}$

followed by $\dfrac{A \sqsubset A' \qquad B \sqsubset B'}{A \to B \sqsubset A' \to B'}$ we thus have $A' \to B' \sqsupset A \to B \sqsubset C'$

by the fundamental lemma, they are heterogeneously equal, and by heq_to_eq they are equal: $e : C' = A' \to B'$

so $\Gamma' \vdash_w \mathtt{transp}(e,t') : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

□

Two useful lemmas

Function type lemma*

Given $\Gamma' \vdash_i t' : C' \in [\![\Gamma \vdash_x t : A \to B]\!]$,
there exists $\Gamma' \vdash_i t'' : A' \to B' \in [\![\Gamma \vdash_x t : A \to B]\!]$

* actually applies to any type former

Choice of type lemma

Given $\Gamma' \vdash_w t' : A' \in [\![\Gamma \vdash_x t : A]\!]$,
and $\Gamma' \vdash_w A'' : \mathsf{Type} \in [\![\Gamma \vdash_x A : \mathsf{Type}]\!]$,
there exists $\Gamma' \vdash_w t'' : A'' \in [\![\Gamma \vdash_x t : A]\!]$

Proof sketch of the main theorem

Conversion rule

$$\frac{\Gamma \vdash_{x} t : A \qquad \Gamma \vdash_{x} A \equiv B}{\Gamma \vdash_{x} t : B}$$

⇢

$$\Gamma' \vdash_{w} p : A' \;{}_{\texttt{Type}}{=}_{\texttt{Type}}\; B' \in \llbracket\, \Gamma \vdash_{x} A \equiv B : \texttt{Type} \,\rrbracket$$

Proof sketch of the main theorem

Conversion rule

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B} \quad \dashrightarrow \quad \begin{array}{l} \Gamma' \vdash_w p : A' \ {}_{\mathtt{Type}}{=}_{\mathtt{Type}} \ B' \\ A \sqsubset A',\ B \sqsubset B' \end{array}$$

Proof sketch of the main theorem

Conversion rule

$$\frac{\Gamma \vdash_x t : A \qquad\qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B} \quad \dashrightarrow \quad \begin{array}{l} \Gamma' \vdash_w p : A' =_{\mathrm{Type}} B' \\ A \sqsubset A',\ B \sqsubset B' \end{array}$$

Proof sketch of the main theorem

Conversion rule

$$\frac{\Gamma \vdash_{x} t : A \qquad \Gamma \vdash_{x} A \equiv B}{\Gamma \vdash_{x} t : B} \quad \dashrightarrow \quad \begin{array}{l} \Gamma' \vdash_{w} p : A' =_{\mathtt{Type}} B' \\ A \sqsubset A',\ B \sqsubset B' \end{array}$$

from the choice of type lemma and the other IH: $\Gamma' \vdash_{w} t' : A' \in [\![\Gamma \vdash_{x} t : A]\!]$

Proof sketch of the main theorem

Conversion rule

$$\frac{\Gamma \vdash_x t : A \qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B}$$

⇢

$$\Gamma' \vdash_w p : A' =_{\mathtt{Type}} B'$$

$$A \sqsubset A',\ B \sqsubset B'$$

from the choice of type lemma and the other IH: $\Gamma' \vdash_w t' : A' \in [\![\ \Gamma \vdash_x t : A\ ]\!]$

so: $\Gamma' \vdash_w \mathtt{transp}(p,t') : B' \in [\![\ \Gamma \vdash_x t : B\ ]\!]$

Proof sketch of the main theorem

Conversion rule

$$\frac{\Gamma \vdash_x t : A \qquad\qquad \Gamma \vdash_x A \equiv B}{\Gamma \vdash_x t : B}$$

⇢

$$\Gamma' \vdash_w p : A' =_{\mathtt{Type}} B'$$

$$A \sqsubset A',\ B \sqsubset B'$$

from the choice of type lemma and the other IH: $\Gamma' \vdash_w t' : A' \in [\![\, \Gamma \vdash_x t : A \,]\!]$

so: $\Gamma' \vdash_w \mathtt{transp}(p,t') : B' \in [\![\, \Gamma \vdash_x t : B \,]\!]$

Other typing rules similar, for application rule we also use the function type lemma

Proof sketch of the main theorem

β-reduction rule

$$\frac{\Gamma,\ x : A \vdash_{x} t : B \qquad\qquad \Gamma \vdash_{x} u : A}{\Gamma \vdash_{x} (\lambda\ (x : A).\ B.\ t)\ @^{(x:A).B}\ u \equiv t\{\ x := u\ \} : B\{\ x := u\}}$$

Proof sketch of the main theorem

β-reduction rule

$$\frac{\Gamma,\ x : A \vdash_x t : B \qquad \Gamma \vdash_x u : A}{\Gamma \vdash_x (\lambda\ (x : A).\ B.\ t)\ @^{(x:A).B}\ u \equiv t\{\ x := u\ \} : B\{\ x := u\}}$$

$$\Gamma' \vdash_w u' : A' \in [\![\ \Gamma \vdash_x u : A\ ]\!]$$

Proof sketch of the main theorem

β-reduction rule

$$\frac{\Gamma,\ x : A \vdash_x t : B \qquad\qquad \Gamma \vdash_x u : A}{\Gamma \vdash_x (\lambda\ (x : A).\ B.\ t)\ @^{(x:A).B}\ u \equiv t\{\ x := u\ \} : B\{\ x := u\}}$$

$$\Gamma' \vdash_w u' : A' \in [\![\ \Gamma \vdash_x u : A\ ]\!]$$

$$\Gamma',\ x : A' \vdash_w t' : B' \in [\![\ \Gamma,\ x : A \vdash_x t : B\ ]\!]$$

Proof sketch of the main theorem

β-reduction rule

$$\frac{\Gamma, x : A \vdash_x t : B \qquad\qquad \Gamma \vdash_x u : A}{\Gamma \vdash_x (\lambda\ (x : A).\ B.\ t)\ @^{(x:A).B}\ u \equiv t\{ x := u \} : B\{ x := u\}}$$

$$\Gamma' \vdash_w u' : A' \in [\![\Gamma \vdash_x u : A]\!]$$

$$\Gamma', x : A' \vdash_w t' : B' \in [\![\Gamma, x : A \vdash_x t : B]\!]$$

$$\Gamma' \vdash_w \beta\ t'\ u' : (\lambda\ (x : A').\ B'.\ t')\ @^{(x:A').B'}\ u' = t'\{ x := u' \}$$

Proof sketch of the main theorem

β-reduction rule

$$\frac{\Gamma,\ x : A \vdash_{x} t : B \qquad\qquad \Gamma \vdash_{x} u : A}{\Gamma \vdash_{x} (\lambda\ (x : A).\ B.\ t)\ @^{(x:A).B}\ u \equiv t\{\ x := u\ \} : B\{\ x := u\}}$$

$$\Gamma' \vdash_{w} u' : A' \in [\![\ \Gamma \vdash_{x} u : A\ ]\!]$$

$$\Gamma',\ x : A' \vdash_{w} t' : B' \in [\![\ \Gamma,\ x : A \vdash_{x} t : B\ ]\!]$$

$$\Gamma' \vdash_{w} \beta\ t'\ u' : (\lambda\ (x : A').\ B'.\ t')\ @^{(x:A').B'}\ u' = t'\{\ x := u'\ \}$$

we conclude using eq_to_heq

Consequences of the translation

Conservativity

If $\vdash_w$ `A : Type` and $\vdash_x$ `t : A`

then there exists $\vdash_w$ `t’ : A` $\in [\![\vdash_x$ `t : A` $]\!]$

Consequences of the translation

Conservativity

If $\vdash_w \mathtt{A : Type}$ and $\vdash_x \mathtt{t : A}$
then there exists $\vdash_w \mathtt{t' : A} \in [\![\vdash_x \mathtt{t : A}]\!]$

Proof using the choice of type lemma

Consequences of the translation

Conservativity

If $\vdash_{w}$ `A : Type` and $\vdash_{x}$ `t : A`
then there exists $\vdash_{w}$ `t' : A` $\in$ $[\![\vdash_{x}$ `t : A` $]\!]$

Proof using the choice of type lemma

Relative consistency

If $\vdash_{x}$ `t :` $\bot$
then there exists $\vdash_{w}$ `t' :` $\bot$

Consequences of the translation

Conservativity

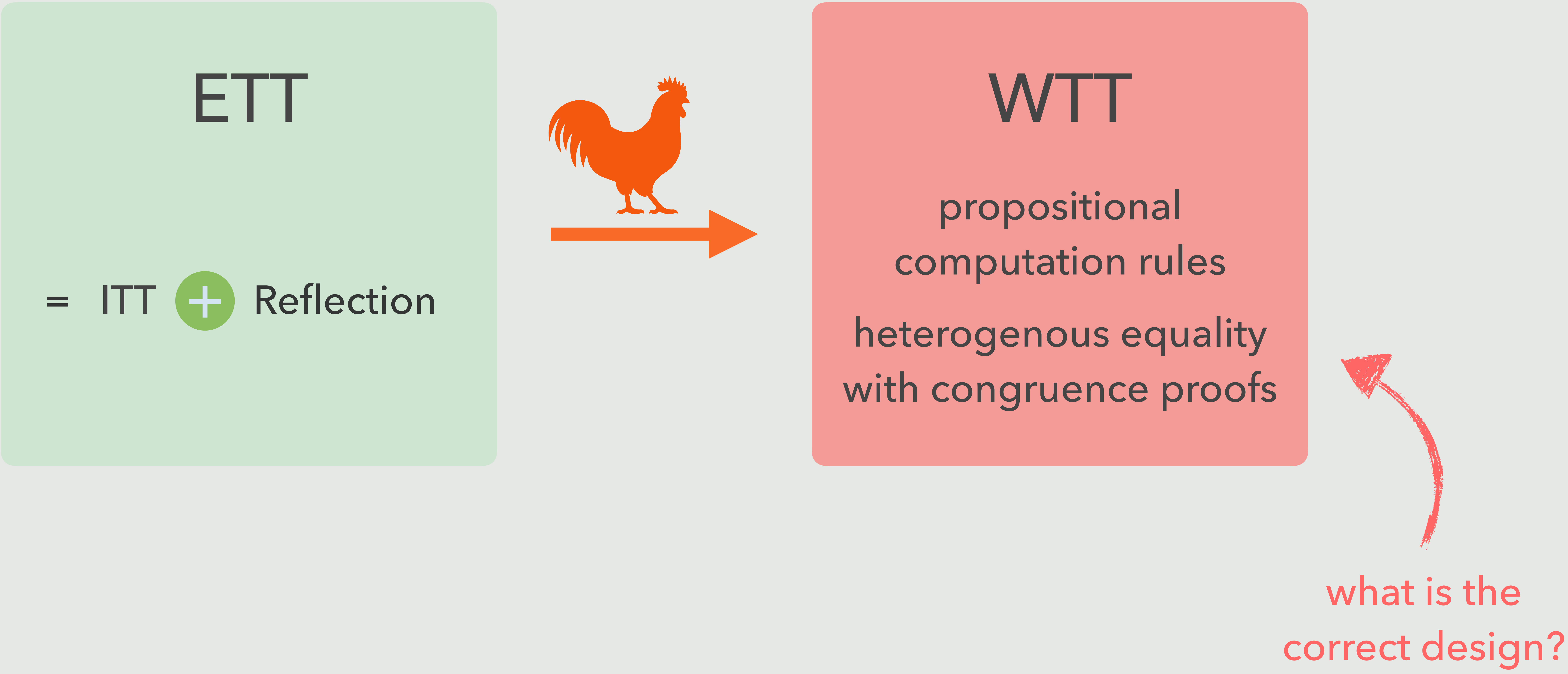
If $\vdash_w \mathtt{A : Type}$ and $\vdash_x \mathtt{t : A}$
then there exists $\vdash_w \mathtt{t' : A} \in [\![\vdash_x \mathtt{t : A}]\!]$

Proof using the choice of type lemma

Relative consistency

If $\vdash_x \mathtt{t} : \bot$
then there exists $\vdash_w \mathtt{t'} : \bot$

Proof using conservativity



ETT

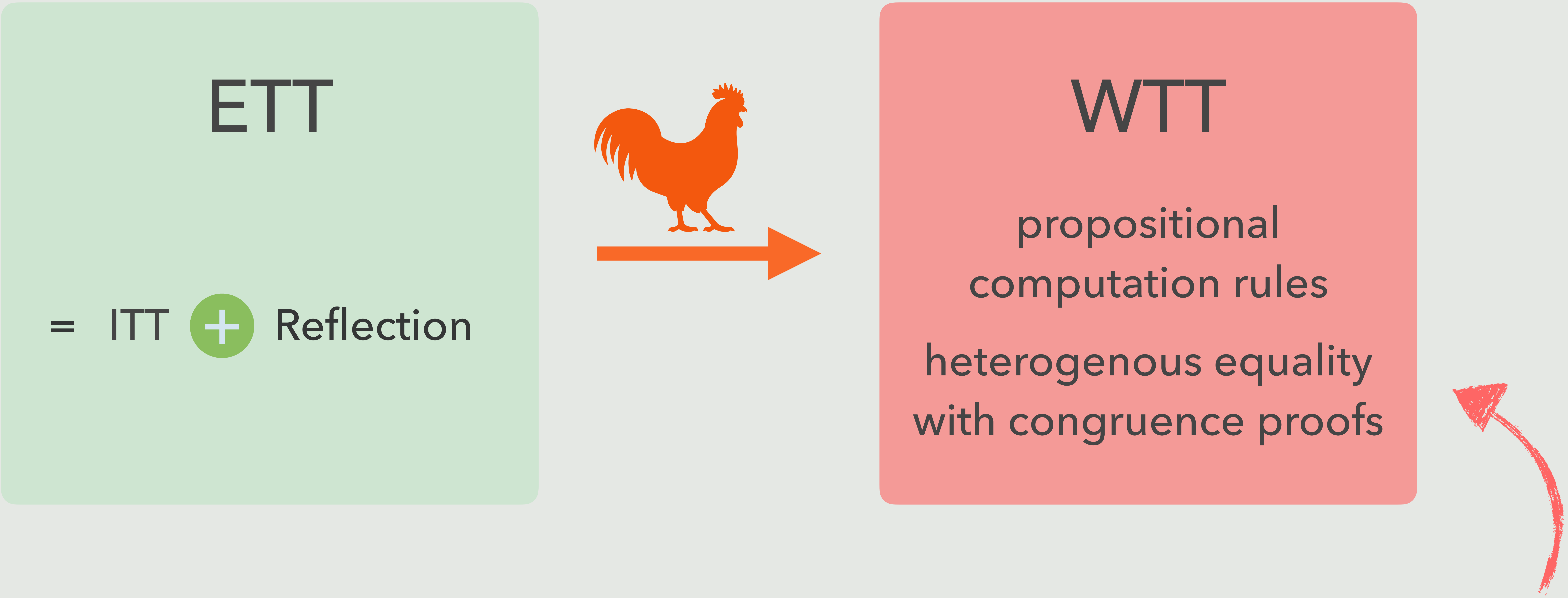
= ITT + Reflection

WTT

propositional
computation rules

heterogenous equality
with congruence proofs

what is the
correct design?



ETT

= ITT ⊕ Reflection

WTT

propositional computation rules

heterogenous equality with congruence proofs

what is the correct design?

Ideally, heterogenous equality should be interpreted just like in ITT

Congruence and binders

$$\dfrac{\begin{array}{c}\Gamma \vdash_{w} \mathtt{pA} : A_1 = A_2 \\ \Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_{w} \mathtt{pB} : B_1[x := \mathtt{Proj}_1\ x] = B_2[x := \mathtt{Proj}_2\ x] \\ \Gamma \vdash_{w} \mathtt{pu} : u_1\ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_{w} \mathtt{pv} : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} pA : A_1 = A_2 \\ \Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_{w} pB : B_1[x := \mathtt{Proj}_1\ x] = B_2[x := \mathtt{Proj}_2\ x] \\ \Gamma \vdash_{w} pu : u_1\ {}_{\Pi(x:A_1).B_1 = \Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_{w} pv : v_1\ {}_{A_1 = A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x := v_1] = B_2[x := v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\frac{\Gamma \vdash_{w} A_1, A_2 : \mathtt{Type}}{\Gamma \vdash_{w} \mathtt{Pack}\ A_1\ A_2 : \mathtt{Type}}$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_w pA : A_1 = A_2 \\ \Gamma, x : \mathsf{Pack}\ A_1\ A_2 \vdash_w pB : B_1[x := \mathsf{Proj}_1\ x] = B_2[x := \mathsf{Proj}_2\ x] \\ \Gamma \vdash_w pu : u_1\ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_w pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_w \mathsf{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\frac{\Gamma \vdash_w A_1, A_2 : \mathsf{Type}}{\Gamma \vdash_w \mathsf{Pack}\ A_1\ A_2 : \mathsf{Type}} \qquad \frac{\Gamma \vdash_w p : \mathsf{Pack}\ A_1\ A_2}{\Gamma \vdash_w \mathsf{Proj}_1\ p : A_1}$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} \mathtt{pA} : A_1 = A_2 \\ \Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_{w} \mathtt{pB} : B_1[x := \mathtt{Proj}_1\ x] = B_2[x := \mathtt{Proj}_2\ x] \\ \Gamma \vdash_{w} \mathtt{pu} : u_1\ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_{w} \mathtt{pv} : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\frac{\Gamma \vdash_{w} A_1, A_2 : \mathtt{Type}}{\Gamma \vdash_{w} \mathtt{Pack}\ A_1\ A_2 : \mathtt{Type}} \qquad \frac{\Gamma \vdash_{w} p : \mathtt{Pack}\ A_1\ A_2}{\Gamma \vdash_{w} \mathtt{Proj}_1\ p : A_1} \qquad \frac{\Gamma \vdash_{w} p : \mathtt{Pack}\ A_1\ A_2}{\Gamma \vdash_{w} \mathtt{Proj}_2\ p : A_2}$$

Congruence and binders

$$\dfrac{\begin{array}{c}\Gamma \vdash_{w} \mathtt{pA} : A_1 = A_2 \\ \Gamma,\ x : \mathtt{Pack}\ A_1\ A_2 \vdash_{w} \mathtt{pB} : B_1[x := \mathtt{Proj}_1\ x] = B_2[x := \mathtt{Proj}_2\ x] \\ \Gamma \vdash_{w} \mathtt{pu} : u_1\ {}_{\Pi(x:A_1).B_1 = \Pi(x:A_2).B_2}\ u_2 \qquad \Gamma \vdash_{w} \mathtt{pv} : v_1\ {}_{A_1 = A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x := v_1] = B_2[x := v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\dfrac{\Gamma \vdash_{w} A_1,\ A_2 : \mathtt{Type}}{\Gamma \vdash_{w} \mathtt{Pack}\ A_1\ A_2 : \mathtt{Type}} \qquad \dfrac{\Gamma \vdash_{w} p : \mathtt{Pack}\ A_1\ A_2}{\Gamma \vdash_{w} \mathtt{Proj}_1\ p : A_1} \qquad \dfrac{\Gamma \vdash_{w} p : \mathtt{Pack}\ A_1\ A_2}{\Gamma \vdash_{w} \mathtt{Proj}_2\ p : A_2}$$

$$\dfrac{\Gamma \vdash_{w} p : \mathtt{Pack}\ A_1\ A_2}{\Gamma \vdash_{w} \mathtt{Proj}_e\ p : \mathtt{Proj}_1\ p\ {}_{A_1 = A_2}\ \mathtt{Proj}_2\ p}$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_w \mathtt{pA} : A_1 = A_2 \\ \Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_w \mathtt{pB} : B_1[x := \mathtt{Proj}_1\ x] = B_2[x := \mathtt{Proj}_2\ x] \\ \Gamma \vdash_w \mathtt{pu} : u_1 \ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_w \mathtt{pv} : v_1 \ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1 \ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1 \ {}_{A_1}{=}_{A_2}\ x_2$$

$$\mathtt{Proj}_1\ p := p.1 \qquad \mathtt{Proj}_2\ p := p.2.1 \qquad \mathtt{Proj}_e\ p := p.2.2$$

Congruence and binders

$$\frac{\begin{array}{c} \Gamma \vdash_w pA : A_1 = A_2 \\ \Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_w pB : B_1[x := x.1] = B_2[x := x.2.1] \\ \Gamma \vdash_w pu : u_1\ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_w pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2 \end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x := v_1]}{=}_{B_2[x := v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$\mathtt{Proj}_1\ p := p.1 \qquad \mathtt{Proj}_2\ p := p.2.1 \qquad \mathtt{Proj}_e\ p := p.2.2$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} \mathtt{pA} : A_1 = A_2 \\ \Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_{w} \mathtt{pB} : B_1[x := x.1] = B_2[x := x.2.1] \\ \Gamma \vdash_{w} \mathtt{pu} : u_1 \ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_{w} \mathtt{pv} : v_1 \ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1 \ @^{(x:A_1).B_1}\ v_1 \ {}_{B_1[x := v_1]}{=}_{B_2[x := v_2]}\ u_2 \ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1 \ {}_{A_1}{=}_{A_2}\ x_2$$

$$a \ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

Congruence and binders

we want to use J on $\boxed{\Gamma \vdash_w pA : A_1 = A_2}$ abstracting the rest

$$\frac{\begin{array}{c}\Gamma, x : \mathtt{Pack}\ A_1\ A_2 \vdash_w pB : B_1[x := x.1] = B_2[x := x.2.1] \\ \Gamma \vdash_w pu : u_1\ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_w pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x := v_1]}{=}_{B_2[x := v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

Congruence and binders

we want to use J on $\boxed{\Gamma \vdash_w \mathtt{pA} : A_1 = A_2}$ abstracting the rest

$$\frac{\begin{array}{c}\Gamma,\ x : \mathtt{Pack}\ A_1\ A_2 \vdash_w \mathtt{pB} : B_1[x := x.1] = B_2[x := x.2.1] \\ \Gamma \vdash_w \mathtt{pu} : u_1\ {}_{\Pi(x:A_1).B_1=\Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_w \mathtt{pv} : v_1\ {}_{A_1=A_2}\ v_2\end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x := v_1]=B_2[x := v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1=A_2}\ x_2$$

$$a\ {}_{A=B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} \mathtt{pA} : A_1 = A_2 \\ \text{using } \beta \text{ and } \mathtt{eq\_trans} \quad \Gamma,\ x : \mathtt{Pack}\ A_1\ A_2 \vdash_{w} \mathtt{pB} : B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} \mathtt{pu} : u_1 \ {}_{\Pi(x:A_1).B_1 = \Pi(x:A_2).B_2}\ u_2 \qquad\qquad \Gamma \vdash_{w} \mathtt{pv} : v_1 \ {}_{A_1 = A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1 \ {}_{B_1[x\ :=\ v_1] = B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1 \ {}_{A_1 = A_2}\ x_2$$

$$a \ {}_{A = B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e, a) = b$$

$$B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} pA : A_1 = A_2 \\ \Gamma \vdash_{w} pB' : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} pu : u_1\ {}_{\Pi(x:A_1).B_1}{=}_{\Pi(x:A_2).B_2}\ u_2 \qquad \Gamma \vdash_{w} pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c} \Gamma \vdash_w pA : A_1 = A_2 \\ \Gamma \vdash_w pB' : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_w pu : u_1\ {}_{\boxed{\Pi(x:A_1).B_1}}{=}_{\boxed{\Pi(x:A_2).B_2}}\ u_2 \qquad\qquad \Gamma \vdash_w pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2 \end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x := v_1]}{=}_{B_2[x := v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\Gamma \vdash_w \mathtt{pA} : A_1 = A_2 \qquad \Gamma \vdash_w \mathtt{pB'} : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \qquad \Gamma \vdash_w \mathtt{pu} : u_1\ {}_{\boxed{\Pi(x:A_1).B_1}}\!=_{\boxed{\Pi(x:A_2).B_2}}\ u_2 \qquad \Gamma \vdash_w \mathtt{pv} : v_1\ {}_{A_1}\!=_{A_2}\ v_2}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}\!=_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

to abstract over B_1’ and B_2’ we need extensionality of Π!

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}\!=_{A_2}\ x_2$$

$$a\ {}_A\!=_B\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$\mathtt{pB'} := \lambda x.\mathtt{pB} \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\dfrac{\begin{array}{c}\Gamma \vdash_w \mathtt{pA} : A_1 = A_2 \\ \Gamma \vdash_w \mathtt{pB'} : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_w \mathtt{pu} : u_1\ {}_{\Pi(x:A_1).B_1'\ x} {=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_w \mathtt{pv} : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$\mathtt{pB'} := \lambda x.\mathtt{pB} \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c} \Gamma \vdash_w pA : A_1 = A_2 \\ \Gamma \vdash_w pB' : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_w pu : u_1\ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_w pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2 \end{array}}{\Gamma \vdash_w \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} pA : A_1 = A_2 \\ \Gamma \vdash_{w} pB' : \Pi\ (x : \mathsf{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} pu : u_1\ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_{w} pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} \mathsf{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ @^{(x:A_1).B_1}\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ @^{(x:A_2).B_2}\ v_2}$$

→ annotations were useful for the proof, now we can drop them!

$$\mathsf{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathsf{transp}(e,a) = b$$

$$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\dfrac{\begin{array}{c} \Gamma \vdash_{w} pA : A_1 = A_2 \\ \Gamma \vdash_{w} pB' : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} pu : u_1\ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_{w} pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2 \end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ pu\ pA\ pB\ pv : u_1\ v_1\ {}_{B_1[x\ :=\ v_1]}{=}_{B_2[x\ :=\ v_2]}\ u_2\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c} \Gamma \vdash_{w} \mathtt{pA} : A_1 = A_2 \\ \Gamma \vdash_{w} \mathtt{pB'} : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} \mathtt{pu} : u_1 \ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_{w} \mathtt{pv} : v_1 \ {}_{A_1}{=}_{A_2}\ v_2 \end{array}}{\Gamma \vdash_{w} \mathtt{cong\_app}\ B_1\ B_2\ \mathtt{pu}\ \mathtt{pA}\ \mathtt{pB}\ \mathtt{pv} : u_1\ v_1 \ {}_{B_1'\ v_1}{=}_{B_2'\ v_2}\ u_2\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1 \ {}_{A_1}{=}_{A_2}\ x_2$$

$$a \ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

$$\mathtt{pB'} := \lambda x.\mathtt{pB} \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} pA : A_1 = A_2 \\ \Gamma \vdash_{w} pB' : \Pi\ (x : \mathrm{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} pu : u_1\ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_{w} pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} ?e : u_1\ v_1\ {}_{B_1'\ v_1}{=}_{B_2'\ v_2}\ u_2\ v_2}$$

$$\mathrm{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathrm{transp}(e,a) = b$$

$$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$$

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} pA : A_1 = A_2 \\ \Gamma \vdash_{w} pB' : \Pi\ (x : \mathsf{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} pu : u_1 \ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_{w} pv : v_1 \ {}_{A_1}{=}_{A_2}\ v_2\end{array}}{\Gamma \vdash_{w} ?e : u_1\ v_1 \ {}_{B_1'\ v_1}{=}_{B_2'\ v_2}\ u_2\ v_2}$$

$$\mathsf{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1 \ {}_{A_1}{=}_{A_2}\ x_2$$

$$a \ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathsf{transp}(e,a) = b$$

~~$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$~~

Congruence and binders

we can now use J on $\boxed{\Gamma \vdash_w pA : A_1 = A_2}$ abstracting the rest

$$\Gamma \vdash_w pB' : \Pi\ (x : \mathtt{Pack}\ A_1\ A_2).\ B_1'\ x.1 = B_2'\ x.2.1$$

$$\frac{\Gamma \vdash_w pu : u_1\ {}_{\Pi(x:A_1).B_1'\ x}{=}_{\Pi(x:A_2).B_2'\ x}\ u_2 \qquad \Gamma \vdash_w pv : v_1\ {}_{A_1}{=}_{A_2}\ v_2}{\Gamma \vdash_w ?e : u_1\ v_1\ {}_{B_1'\ v_1}{=}_{B_2'\ v_2}\ u_2\ v_2}$$

$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$

$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$

~~$pB' := \lambda x.pB \qquad B_1' := \lambda x.B_1 \qquad B_2' := \lambda x.B_2$~~

Congruence and binders

$$\frac{\begin{array}{c}\Gamma \vdash_{w} \mathtt{pB'} : \Pi\ (x : \mathtt{Pack}\ A\ A).\ B_1'\ x.1 = B_2'\ x.2.1 \\ \Gamma \vdash_{w} \mathtt{pu} : u_1\ {}_{\Pi(x:A).B_1'\ x}{=}_{\Pi(x:A).B_2'\ x}\ u_2 \qquad\qquad \Gamma \vdash_{w} \mathtt{pv} : v_1\ {}_{A}{=}_{A}\ v_2\end{array}}{\Gamma \vdash_{w} ?e : u_1\ v_1\ {}_{B_1'\ v_1}{=}_{B_2'\ v_2}\ u_2\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

Congruence and binders

from this and computation equalities for sums we prove $\Pi\ (x : A).\ B_1'\ x = B_2'\ x$

$$\boxed{\Gamma \vdash_w pB' : \Pi\ (x : \mathtt{Pack}\ A\ A).\ B_1'\ x.1 = B_2'\ x.2.1}$$

$$\frac{\Gamma \vdash_w pu : u_1\ {}_{\Pi(x:A).B_1'\,x}{=}_{\Pi(x:A).B_2'\,x}\ u_2 \qquad\qquad \Gamma \vdash_w pv : v_1\ {}_{A}{=}_{A}\ v_2}{\Gamma \vdash_w ?e : u_1\ v_1\ {}_{B_1'\,v_1}{=}_{B_2'\,v_2}\ u_2\ v_2}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$

Congruence and binders

from this and computation equalities for sums, and funext we prove $B_1' = B_2'$

$$\boxed{\Gamma \vdash_w pB' : \Pi\ (x : \mathsf{Pack}\ A\ A).\ B_1'\ x.1 = B_2'\ x.2.1}$$

$$\frac{\Gamma \vdash_w pu : u_1 \ {}_{\Pi(x:A).B_1'\ x}{=}_{\Pi(x:A).B_2'\ x}\ u_2 \qquad \Gamma \vdash_w pv : v_1\ {}_{A}{=}_{A}\ v_2}{\Gamma \vdash_w ?e : u_1\ v_1\ {}_{B_1'\ v_1}{=}_{B_2'\ v_2}\ u_2\ v_2}$$

$$\mathsf{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathsf{transp}(e,a) = b$$

Congruence and binders

$$\frac{\Gamma \vdash_{w} pu : u_1 \; {}_{\Pi(x:A).B}{=}_{\Pi(x:A).B} \; u_2 \qquad\qquad \Gamma \vdash_{w} pv : v_1 \; {}_{A}{=}_{A} \; v_2}{\Gamma \vdash_{w} ?e : u_1 \; v_1 \; {}_{B \; v_1}{=}_{B \; v_2} \; u_2 \; v_2}$$

$$\mathsf{Pack}\; A_1 \; A_2 := \Sigma \; (x_1 : A_1) \; (x_2 : A_2). \; x_1 \; {}_{A_1}{=}_{A_2} \; x_2$$

$$a \; {}_{A}{=}_{B} \; b := \Sigma \; (e : A = B). \; \mathsf{transp}(e,a) = b$$

Congruence and binders

these we just deal with UIP

$$\frac{\Gamma \vdash_w \mathtt{pu} : u_1 \ {}_{\Pi(x:A).B}{=}_{\Pi(x:A).B} \ u_2 \qquad \Gamma \vdash_w \mathtt{pv} : v_1 \ {}_{A}{=}_{A} \ v_2}{\Gamma \vdash_w \mathtt{?e} : u_1 \ v_1 \ {}_{B\ v_1}{=}_{B\ v_2} \ u_2 \ v_2}$$

$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$

$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$

Congruence and binders

$$\frac{\Gamma \vdash_w pu : u_1 = u_2 \qquad\qquad \Gamma \vdash_w pv : v_1 = v_2}{\Gamma \vdash_w ?e : u_1\ v_1\ {}_{B\ v_1}{=}_{B\ v_2}\ u_2\ v_2}$$

$$\mathsf{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathsf{transp}(e,a) = b$$

Congruence and binders

$$\frac{\Gamma \vdash_{w} \mathtt{pv} : v_1 = v_2}{\Gamma \vdash_{w} \mathtt{?e} : u\ v_1 \ {}_{B\ v_1}{=}_{B\ v_2}\ u\ v_2}$$

$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$

$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$

Congruence and binders

$$\Gamma \;\vdash_{w}\; ?e \;:\; u \;\; v \;\; {}_{B\ v}{=}_{B\ v} \;\; u \;\; v$$

$$\text{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \text{transp}(e,a) = b$$

Congruence and binders

$$\Gamma \vdash_{w} \mathtt{refl} : B\ v = B\ v$$

$$\Gamma \vdash_{w} ?p : \mathtt{transp}(\mathtt{refl},\ u\ v) = u\ v$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1 = A_2}\ x_2$$

$$a\ {}_{A = B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e, a) = b$$

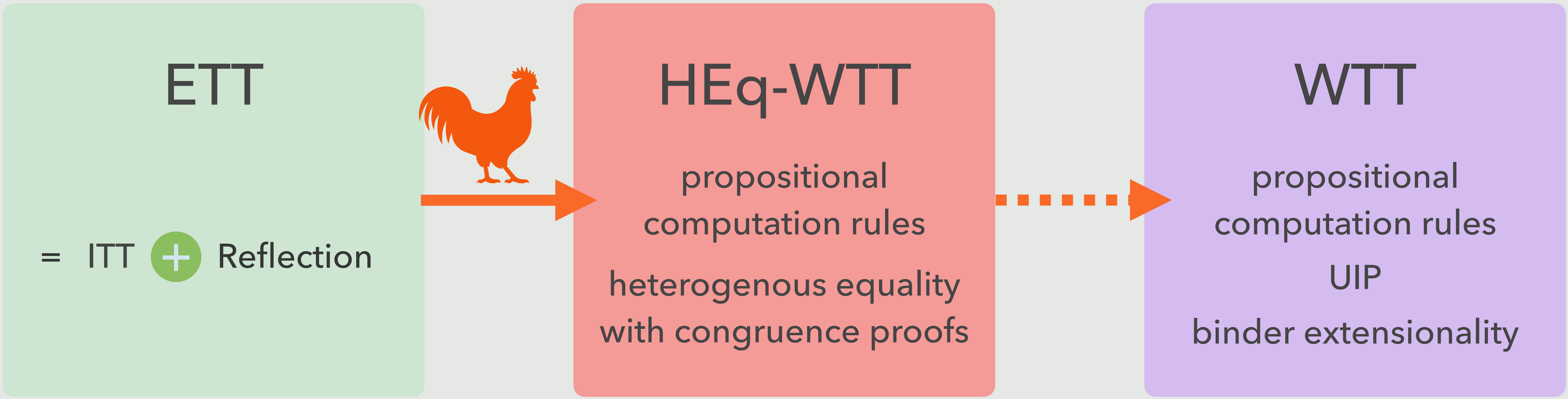
Congruence and binders

$$\Gamma \vdash_{w} \mathtt{refl} : \mathtt{B\ v} = \mathtt{B\ v}$$

$$\Gamma \vdash_{w} \mathtt{transp\_comp} : \mathtt{transp}(\mathtt{refl}, \mathtt{u\ v}) = \mathtt{u\ v}$$

$$\mathtt{Pack}\ A_1\ A_2 := \Sigma\ (x_1 : A_1)\ (x_2 : A_2).\ x_1\ {}_{A_1}{=}_{A_2}\ x_2$$

$$a\ {}_{A}{=}_{B}\ b := \Sigma\ (e : A = B).\ \mathtt{transp}(e,a) = b$$



ETT

= ITT + Reflection

HEq-WTT

propositional computation rules

heterogenous equality with congruence proofs

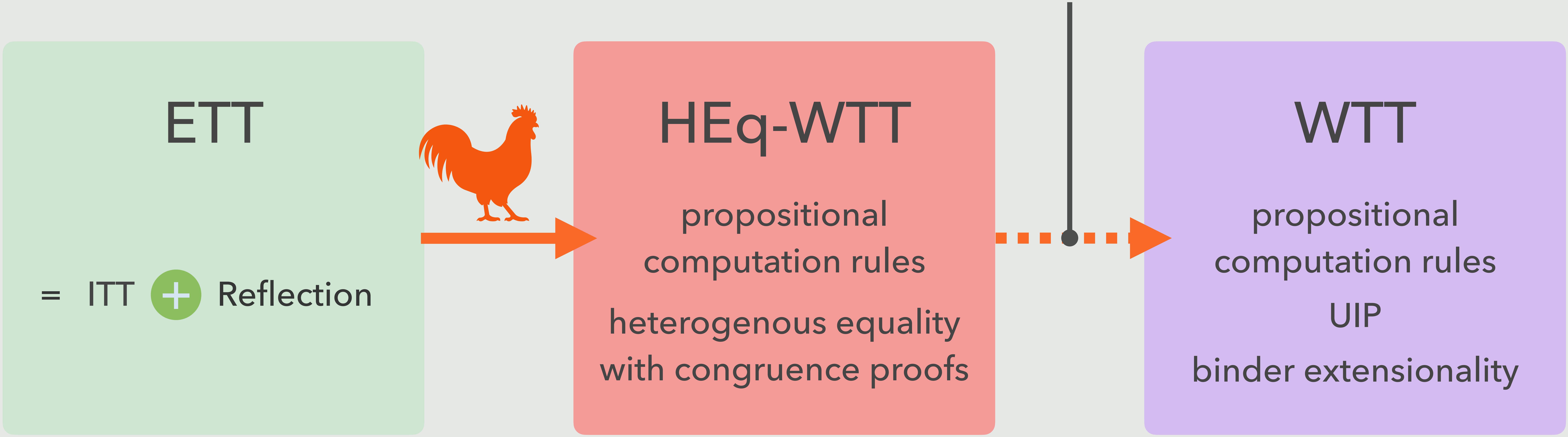
WTT

propositional computation rules

UIP

binder extensionality

Probably have all the tools (sound and complete checker) but still very tedious to formalise!



ETT

= ITT + Reflection

HEq-WTT

propositional computation rules

heterogenous equality with congruence proofs

WTT

propositional computation rules

UIP

binder extensionality

Perspectives

Proof certificates

de Bruijn: A Plea for **Weaker** Frameworks

Problem: proof terms are too big
which brings us back to WTT design

Local computation

Example: parallel plus

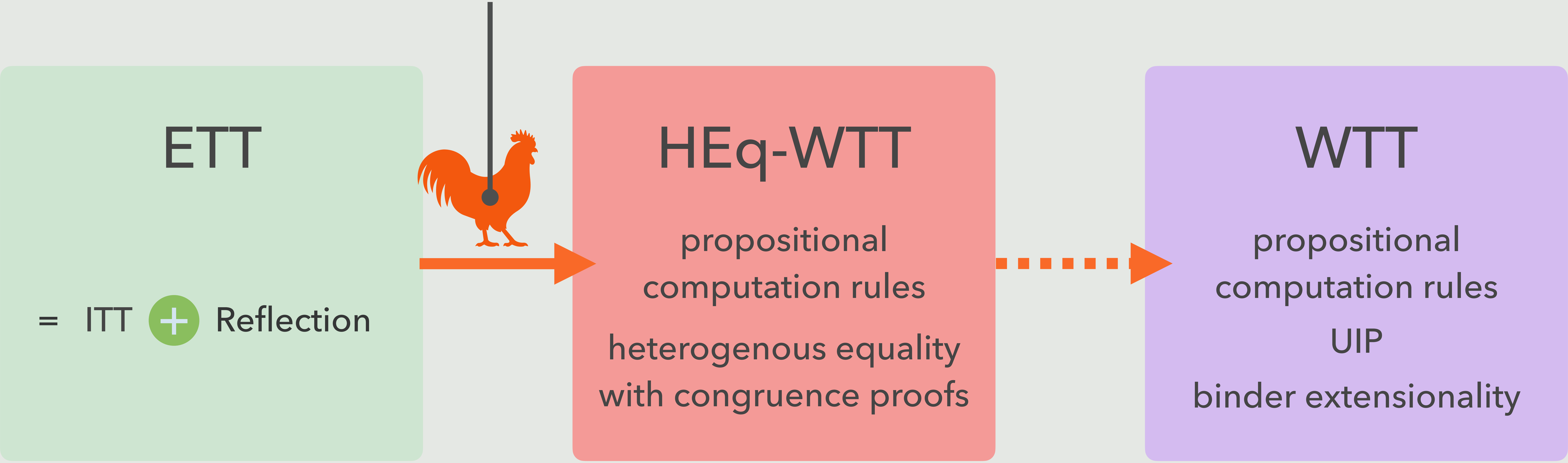
```
λx. x + 0 ≡ λx. x
```

gets translated to

```
λx. x + 0 = λx. x
```

using funext

Also extended to 2 level type theories!



ETT

= ITT + Reflection

HEq-WTT

propositional computation rules

heterogenous equality with congruence proofs

WTT

propositional computation rules

UIP

binder extensionality

/TheoWinterhalter/ett-to-wtt