A Shiny Hammer and Many Things to Hit

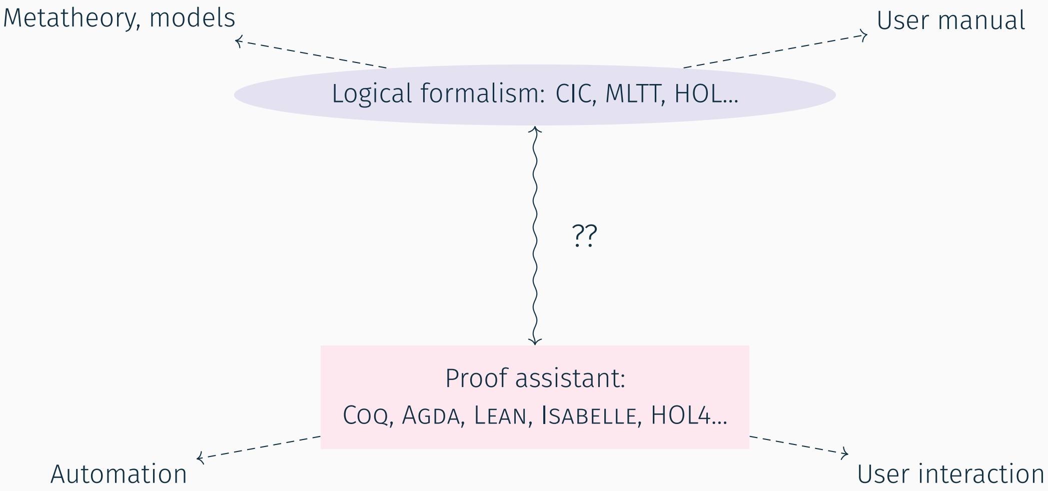
Bidirectional typing is not only an implementation technique

Meven Lennon-Bertrand

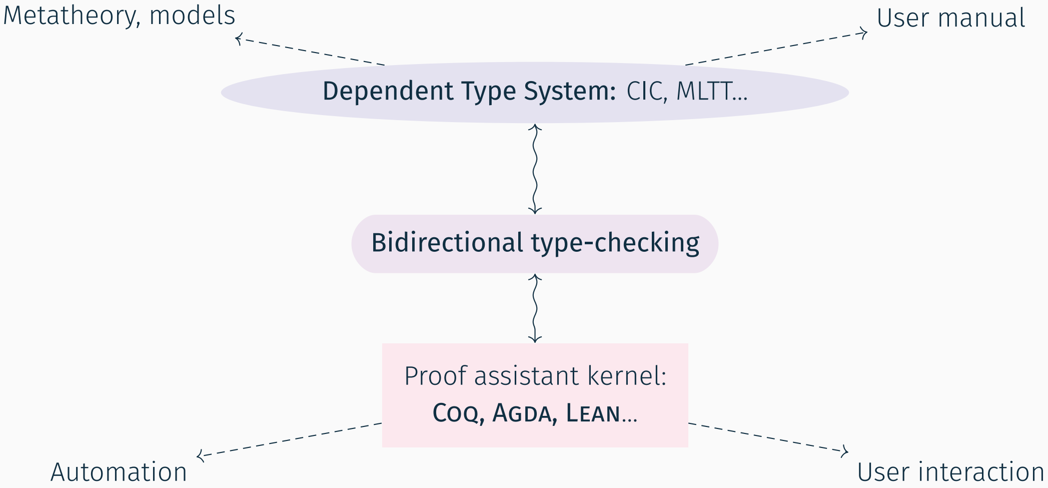
University of Strathclyde – June 27th 2023

University of Cambridge
Department of Computer Science and Technology

Specifying Proof Assistants



Specifying Proof Assistants



Metatheory, models

User manual

Dependent Type System: CIC, MLTT...

Bidirectional type-checking

Proof assistant kernel:

Coq, Agda, Lean...

Automation

User interaction

The hammer: bidirectional typing

$$\frac{(x{:}\,T \in \Gamma)}{\Gamma \vdash x : T} \qquad \frac{}{\Gamma \vdash \star : 1} \qquad \frac{\Gamma, x{:}\,A \vdash t : B}{\Gamma \vdash \lambda\, x.t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output.

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x : T} \qquad \frac{}{\Gamma \vdash \star : 1} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output.

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \star : 1} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda\, x.t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x : T}$$

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output.

$$\frac{(x{:}\,T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \star \triangleright 1} \qquad \boxed{\frac{\Gamma, x{:}\,A \vdash t \triangleleft B}{\Gamma \vdash \lambda\, x.t \triangleleft A \to B}} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

$$\frac{\Gamma, x{:}\,A \vdash t : B}{\Gamma \vdash \lambda\, x.t : A \to B}$$

Starting simple: simply-typed λ-calculus

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output.

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \star \triangleright 1} \qquad \frac{\Gamma, x : A \vdash t \triangleleft B}{\Gamma \vdash \lambda x.t \triangleleft A \to B} \qquad \boxed{\frac{\Gamma \vdash t \triangleright A \to B \qquad \Gamma \vdash u \triangleleft A}{\Gamma \vdash t\, u \triangleright B}}$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output.

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \star \triangleright 1} \qquad \frac{\Gamma, x : A \vdash t \triangleleft B}{\Gamma \vdash \lambda\, x.t \triangleleft A \to B} \qquad \frac{\Gamma \vdash t \triangleright A \to B \qquad \Gamma \vdash u \triangleleft A}{\Gamma \vdash t\ u \triangleright B}$$

What about checking a variable? And $(\lambda\, x.\star)\ \star$?

Starting simple: simply-typed λ-calculus

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output.

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \star \triangleright 1} \qquad \frac{\Gamma, x : A \vdash t \triangleleft B}{\Gamma \vdash \lambda\, x.t \triangleleft A \to B} \qquad \frac{\Gamma \vdash t \triangleright A \to B \qquad \Gamma \vdash u \triangleleft A}{\Gamma \vdash t\ u \triangleright B}$$

$$\frac{\Gamma \vdash t \triangleright T \qquad T = T'}{\Gamma \vdash t \triangleleft T'} \qquad \frac{\Gamma \vdash t \triangleleft T}{\Gamma \vdash t :: T \triangleright T}$$

$$((\lambda\, x.\star) :: 1 \to 1)\ \star$$

Boundaries and Invariants

A typing judgment $\Gamma \vdash t : T$ has *boundaries*. What about their well-formation?

A typing judgment $\Gamma \vdash t : T$ has *boundaries*. What about their well-formation?

Cautiousness: globally enforce well-formation

$$\frac{\vdash \Gamma \qquad (x{:}\, A) \in \Gamma}{\Gamma \vdash x : A} \qquad\qquad \frac{\Gamma, x{:}\, A \vdash t : B}{\Gamma \vdash \lambda\, x{:}\, A.t : \prod x{:}\, A.B}$$

A typing judgment $\Gamma \vdash t : T$ has *boundaries*. What about their well-formation?

Cautiousness: globally enforce well-formation

$$\frac{\vdash \Gamma \qquad (x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda\, x : A.t : \prod x : A.B}$$

Uncautiousness? Well-formation as an invariant

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma \vdash A : \square \qquad \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda\, x : A.t : \prod x : A.B}$$

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output/subject.

Well-Formation Must Flow

Inference and checking

$\Gamma \vdash t : T$ separates into

inference: $\Gamma \vdash t \triangleright T$ checking: $\Gamma \vdash t \triangleleft T$

Similar meaning, different modes: input/output/subject.

The TypOS discipline

- *A rule is a server for its conclusion and a client for its premises.*
- Modes guide invariant preservation
- In a conclusion, you *assume* inputs are well-formed, and *ensure* outputs are
- In a premise, you *ensure* inputs are well-formed, and *assume* outputs are

$$\frac{\vdash \Gamma \qquad (x : T \in \Gamma)}{\Gamma \vdash x : T} \qquad \frac{\vdash \Gamma}{\Gamma \vdash \Box_i : \Box_{i+1}} \qquad \frac{\Gamma \vdash A : \Box_i \qquad \Gamma, x : A \vdash B : \Box_j}{\Gamma \vdash \prod x : A.B : \Box_{i \vee j}}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda\, x : A.t : \prod x : A.B} \qquad \frac{\Gamma \vdash t : \prod x : A.B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[u]}$$

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t : T'}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \square_i : \square_{i+1}}$$

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T}$$

$$\frac{}{\Gamma \vdash \square_i \triangleright \square_{i+1}}$$

$$\frac{\Gamma \vdash A : \square_i \qquad \Gamma, x : A \vdash B : \square_j}{\Gamma \vdash \prod x : A.B : \square_{i \vee j}}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda\, x : A.t : \prod x : A.B}$$

$$\frac{\Gamma \vdash t : \prod x : A.B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[u]}$$

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t : T'}$$

$$\frac{\Gamma \vdash A : \square_i \qquad \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda\, x : A . t : \prod x : A . B}$$

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \square_i \triangleright \square_{i+1}} \qquad \frac{\Gamma \vdash A \triangleright_{\square} \square_i \qquad \Gamma, x : A \vdash B \triangleright_{\square} \square_j}{\Gamma \vdash \prod x : A . B \triangleright \square_{i \vee j}}$$

$$\boxed{\frac{\Gamma \vdash A \triangleright_{\square} \square_i \qquad \Gamma, x : A \vdash t \triangleright B}{\Gamma \vdash \lambda\, x : A . t \triangleright \prod x : A . B}} \qquad \frac{\Gamma \vdash t : \prod x : A . B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B[u]}$$

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t : T'}$$

$$\frac{\Gamma \vdash t : \Pi\, x\colon A.B \qquad \Gamma \vdash u\colon A}{\Gamma \vdash t\ u : B[u]}$$

$$\frac{(x\colon T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \square_i \triangleright \square_{i+1}} \qquad \frac{\Gamma \vdash A \triangleright_{\square} \square_i \qquad \Gamma, x\colon A \vdash B \triangleright_{\square} \square_j}{\Gamma \vdash \Pi\, x\colon A.B \triangleright \square_{i \vee j}}$$

$$\frac{\Gamma \vdash A \triangleright_{\square} \square_i \qquad \Gamma, x\colon A \vdash t \triangleright B}{\Gamma \vdash \lambda\, x\colon A.t \triangleright \Pi\, x\colon A.B} \qquad \boxed{\frac{\Gamma \vdash t \triangleright_{\Pi} \Pi\, x\colon A.B \qquad \Gamma \vdash u \triangleleft A}{\Gamma \vdash t\ u \triangleright B[u]}}$$

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t : T'}$$

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t : T'}$$

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x \triangleright T} \qquad \frac{}{\Gamma \vdash \square_i \triangleright \square_{i+1}} \qquad \frac{\Gamma \vdash A \triangleright_{\square} \square_i \qquad \Gamma, x : A \vdash B \triangleright_{\square} \square_j}{\Gamma \vdash \Pi\, x : A.B \triangleright \square_{i \vee j}}$$

$$\frac{\Gamma \vdash A \triangleright_{\square} \square_i \qquad \Gamma, x : A \vdash t \triangleright B}{\Gamma \vdash \lambda\, x : A.t \triangleright \Pi\, x : A.B} \qquad \frac{\Gamma \vdash t \triangleright_{\Pi} \Pi\, x : A.B \qquad \Gamma \vdash u \triangleleft A}{\Gamma \vdash t\ u \triangleright B[u]}$$

$$\frac{\Gamma \vdash t \triangleright T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t \triangleleft T'} \qquad \frac{\Gamma \vdash t \triangleright T \qquad T \to^{\star} \square_i}{\Gamma \vdash t \triangleright_{\square} \square_i} \qquad \frac{\Gamma \vdash t \triangleright T \qquad T \to^{\star} \Pi\, x : A.B}{\Gamma \vdash t \triangleright_{\Pi} \Pi\, x : A.B}$$

- Different modes command **different computation judgments** ($\to^{\star}$ vs $\cong$)
- **No free conversion** thanks to the judgments' structure

Theorems!

Theorems!

Nothing's changed...

- Soundness: if $\vdash \Gamma$ and $\Gamma \vdash t \triangleright T$ then $\Gamma \vdash t : T$

Theorems!

Nothing's changed...

- Soundness: if $\vdash \Gamma$ and $\Gamma \vdash t \triangleright T$ then $\Gamma \vdash t : T$
- Completeness: if $\Gamma \vdash t : T$, there exists T' such that $\Gamma \vdash t \triangleright T'$ and $\Gamma \vdash T' \cong T$

Nothing's changed...

- Soundness: if $\vdash \Gamma$ and $\Gamma \vdash t \triangleright T$ then $\Gamma \vdash t : T$
- Completeness: if $\Gamma \vdash t : T$, there exists T' such that $\Gamma \vdash t \triangleright T'$ and $\Gamma \vdash T' \cong T$

Key: some sort of confluence.

Nothing's changed...

- Soundness: if $\vdash \Gamma$ and $\Gamma \vdash t \triangleright T$ then $\Gamma \vdash t : T$
- Completeness: if $\Gamma \vdash t : T$, there exists T' such that $\Gamma \vdash t \triangleright T'$ and $\Gamma \vdash T' \cong T$

... unless it has!

Easy proofs of

- uniqueness of types/principality
- strengthening

Nail I: Certifying Coq's kernel

Jww. the MetaCoq team

Real life

Real life

The Predicative Calculus of Universe-Polymorphic Inductive Constructions

CCω +

- Complex universes
- Very general (co-)inductive types
- Cumulativity/subtyping

Real life

The Predicative Calculus of Universe-Polymorphic Inductive Constructions

CCω +

- Complex universes
- Very general (co-)inductive types
- Cumulativity/subtyping

MetaCoq: Coq, in Coq

Real life

The Predicative Calculus of Universe-Polymorphic Inductive Constructions

CCω +

- Complex universes
- Very general (co-)inductive types
- Cumulativity/subtyping

MetaCoq: Coq, in Coq

- Formalized meta-theory of PCUIC

The Predicative Calculus of Universe-Polymorphic Inductive Constructions

CCω +

- Complex universes
- Very general (co-)inductive types
- Cumulativity/subtyping

MetaCoq: Coq, in Coq

- Formalized meta-theory of PCUIC
- Normalization axiom to implement a certified type-checker

Real life

The Predicative Calculus of Universe-Polymorphic Inductive Constructions

CCω +

- Complex universes
- Very general (co-)inductive types
- Cumulativity/subtyping

MetaCoq: Coq, in Coq

- Formalized meta-theory of PCUIC
- Normalization axiom to implement a certified type-checker
- Extraction, meta-programming…

Real life

The Predicative Calculus of Universe-Polymorphic Inductive Constructions

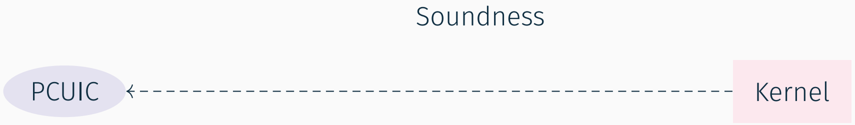
CCω +

- Complex universes
- Very general (co-)inductive types
- Cumulativity/subtyping

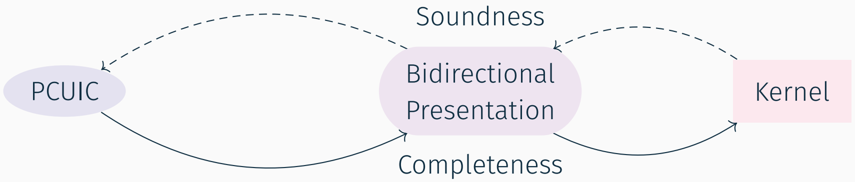
MetaCoq: Coq, in Coq

- Formalized meta-theory of PCUIC
- Normalization axiom to implement a certified type-checker
- Extraction, meta-programming…

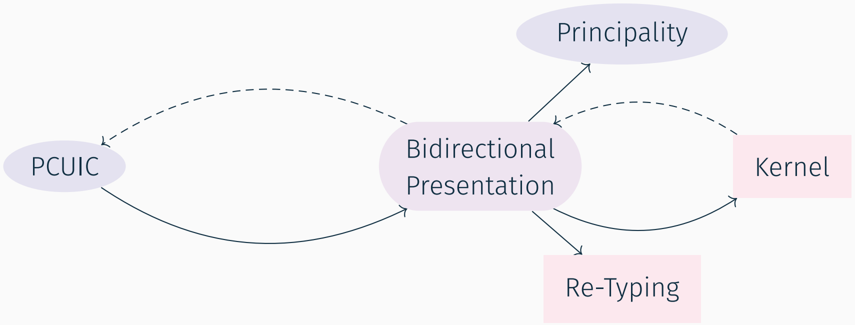
A Correct and Complete Kernel



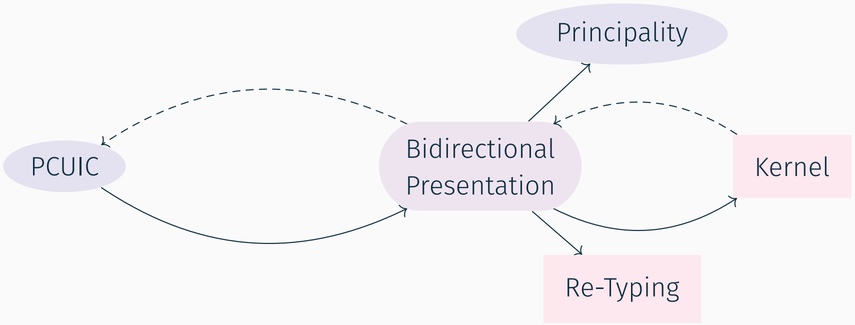
A Correct and Complete Kernel



A Correct and Complete Kernel

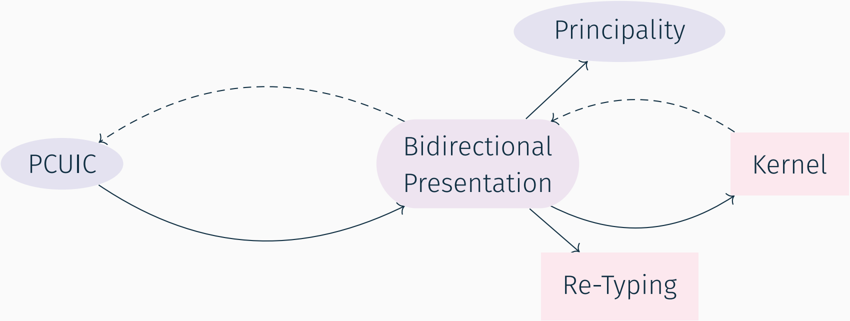


A Correct and Complete Kernel



When starting the proof, we realized… it was false!

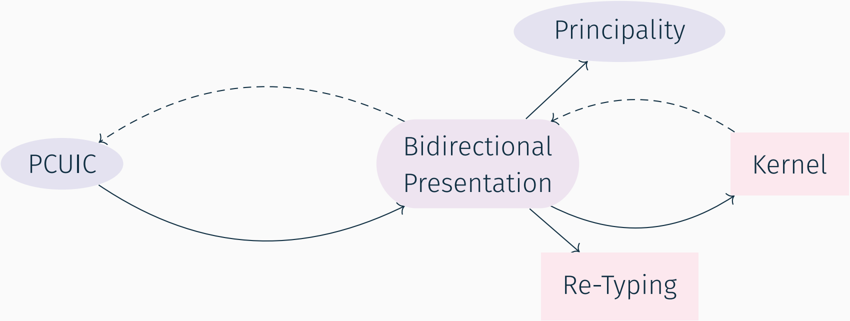
A Correct and Complete Kernel



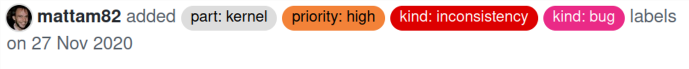
When starting the proof, we realized… it was false!

mattam82 added part: kernel priority: high kind: inconsistency kind: bug labels on 27 Nov 2020

A Correct and Complete Kernel



When starting the proof, we realized… it was false!



Ended up with a complete re-design of pattern-matching…

Nail II: Gradual Dependent Types

Jww. K. Maillard, N. Tabareau, and É. Tanter

Gradual Typing in One Slide

Mixing static and dynamic typing

- Static type system with a dynamic type $?$
- Optimistic (static) typing & (dynamic) runtime checks

Mixing static and dynamic typing

- Static type system with a dynamic type $?$
- Optimistic (static) typing & (dynamic) runtime checks

Subject reduction is broken?

$$\vdash (\lambda\, x{:}\, ?.\, x + 1)\ \mathrm{true} : \mathbb{N}$$

$$(\lambda\, x{:}\, ?.\, x + 1)\ \mathrm{true} \to^{\star}\ \mathrm{true} + 1$$

$$\nvdash \mathrm{true} + 1$$

Gradual Typing in One Slide

Mixing static and dynamic typing

- Static type system with a dynamic type ?
- Optimistic (static) typing & (dynamic) runtime checks

Not in the cast calculus!

$$\vdash (\lambda x\!: ?.(\langle \mathbf{N} \Leftarrow ?\rangle\ x) + 1)\ (\langle ? \Leftarrow \mathbf{B}\rangle\ \text{true}) : \mathbf{N}$$

Gradual Typing in One Slide

Mixing static and dynamic typing

- Static type system with a dynamic type ?
- Optimistic (static) typing & (dynamic) runtime checks

Not in the cast calculus!

$$\vdash (\lambda x\colon ?.(\langle \mathrm{N} \Leftarrow ?\rangle\ x) + 1)\ (\langle ? \Leftarrow \mathrm{B}\rangle\ \mathrm{true}) : \mathrm{N}$$

$$(\lambda x\colon ?.(\langle \mathrm{N} \Leftarrow ?\rangle\, x) + 1)\ (\langle ? \Leftarrow \mathrm{B}\rangle\, \mathrm{true}) \to^{\star} (\langle \mathrm{N} \Leftarrow ?\rangle \langle ? \Leftarrow \mathrm{B}\rangle\, \mathrm{true}) + 1$$

$$\to^{\star} (\langle \mathrm{N} \Leftarrow \mathrm{B}\rangle\ \mathrm{true}) + 1 \to^{\star} \mathrm{err}$$

Gradual Typing in One Slide

Mixing static and dynamic typing

- Static type system with a dynamic type ?
- Optimistic (static) typing & (dynamic) runtime checks

Not in the cast calculus!

$$\vdash (\lambda x\colon ?.(\langle \mathbf{N} \Leftarrow ?\rangle\ x) + 1)\ (\langle ? \Leftarrow \mathbf{B}\rangle\ \mathbf{true}) : \mathbf{N}$$

$$(\lambda x\colon ?.(\langle \mathbf{N} \Leftarrow ?\rangle\, x) + 1)\, (\langle ? \Leftarrow \mathbf{B}\rangle\, \mathbf{true}) \to^{\star} (\langle \mathbf{N} \Leftarrow ?\rangle\, \langle ? \Leftarrow \mathbf{B}\rangle\, \mathbf{true}) + 1$$

$$\to^{\star} (\langle \mathbf{N} \Leftarrow \mathbf{B}\rangle\ \mathbf{true}) + 1 \to^{\star} \mathbf{err}$$

But we still want a cast-free source language…

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T}$$

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T}$$

Issues

- Non-transitivity: $S \sim ? \sim T$

Bidirectionalism to the rescue

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T}$$

Issues

- Non-transitivity: $S \sim ? \sim T$

$$\frac{\dfrac{\Gamma \vdash t : S \qquad S \sim ?}{\Gamma \vdash t : ?} \qquad ? \sim T}{\Gamma \vdash t : T}$$

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T}$$

Issues

- Non-transitivity: $S \sim ? \sim T$

Solutions

- Bidirectional typing

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T} \qquad\qquad \frac{\Gamma \vdash t \triangleright S \qquad S \sim T}{\Gamma \vdash t \triangleleft T}$$

Issues

- Non-transitivity: $S \sim ? \sim T$

Solutions

- Bidirectional typing

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T} \qquad \frac{\Gamma \vdash t \triangleright S \qquad S \sim T}{\Gamma \vdash t \triangleleft T}$$

Issues

- Non-transitivity: $S \sim ? \sim T$
- Computation needs checks

Solutions

- Bidirectional typing

Bidirectionalism to the rescue

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T} \qquad\qquad \frac{\Gamma \vdash t \triangleright S \qquad S \sim T}{\Gamma \vdash t \triangleleft T}$$

Issues

- Non-transitivity: $S \sim \, ? \sim T$
- Computation needs checks

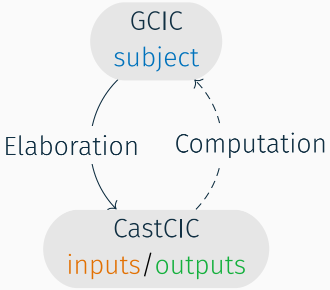
Solutions

- Bidirectional typing
- Type-directed elaboration

Bidirectionalism to the rescue

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T}$$

$$\frac{\Gamma \vdash t \triangleright S \qquad S \sim T}{\Gamma \vdash t \triangleleft T}$$



Issues

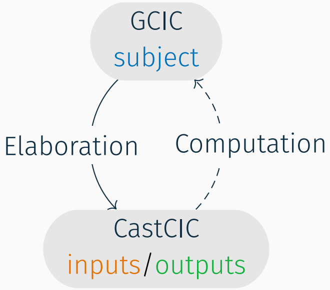
- Non-transitivity: $S \sim ? \sim T$
- Computation needs checks

Solutions

- Bidirectional typing
- Type-directed elaboration

$$\frac{\Gamma \vdash t : S \qquad S \sim T}{\Gamma \vdash t : T} \qquad \frac{\Gamma \vdash t \triangleright S \qquad S \sim T}{\Gamma \vdash t \triangleleft T}$$

$$\frac{\Gamma \vdash t \rightsquigarrow t' \triangleright S \qquad S \sim T}{\Gamma \vdash t \rightsquigarrow \langle T \Leftarrow S \rangle\, t' \triangleleft T}$$



Issues

- Non-transitivity: $S \sim ? \sim T$
- Computation needs checks

Solutions

- Bidirectional typing
- Type-directed elaboration

Nail III: Logical relations

Jww. K. Maillard and L. Pujet

What About Conversion?

It's bidirectional too!

Conversion $\cong$ checks, neutral comparison $\approx$ infers

$$\frac{t \to^\star t' \qquad u \to^\star u' \qquad A \to^\star A' \qquad \Gamma \vdash t' \cong_{\mathrm{h}} u' \triangleleft A'}{\Gamma \vdash t \cong u \triangleleft A}$$

$$\frac{\Gamma, x\colon A \vdash f\ x \cong g\ x \triangleleft B}{\Gamma \vdash f \cong_{\mathrm{h}} g \triangleleft \Pi\, x\colon A.\ B} \qquad \frac{\Gamma \vdash m \approx n \triangleright_{\Pi} \Pi\, x\colon A.\ B \qquad \Gamma \vdash t \cong u \triangleleft A}{\Gamma \vdash m\ t \approx n\ u \triangleright B[t]}$$

What About Conversion?

Conversion ≅ checks, neutral comparison ≈ infers

$$\frac{t \to^\star t' \qquad u \to^\star u' \qquad A \to^\star A' \qquad \Gamma \vdash t' \cong_{\mathrm{h}} u' \triangleleft A'}{\Gamma \vdash t \cong u \triangleleft A}$$

$$\frac{\Gamma, x : A \vdash f\,x \cong g\,x \triangleleft B}{\Gamma \vdash f \cong_{\mathrm{h}} g \triangleleft \Pi\, x : A.\, B} \qquad \frac{\Gamma \vdash m \approx n \triangleright_{\Pi} \Pi\, x : A.\, B \qquad \Gamma \vdash t \cong u \triangleleft A}{\Gamma \vdash m\,t \approx n\,u \triangleright B[t]}$$

logrel-coq: logical relations for dependent type theory, in Coq.

How to concretely translate the TypOS discipline?

How to concretely translate the TypOS discipline?

It's a custom induction principle!

Bundled induction

How to concretely translate the TYPOS discipline?

It's a custom induction principle!

To show $\forall\ \Gamma\ t\ T,\ [\ \vdash\ \Gamma] \Rightarrow [\Gamma\ \vdash\ T] \Rightarrow [\Gamma\ \vdash\ t \triangleleft T] \Rightarrow P\ \Gamma\ t\ T$
by induction on the last premise, you get extra help in induction steps.

Bundled induction

How to concretely translate the TYPOS discipline?

It's a custom induction principle!

To show $\forall\ \Gamma\ t\ T,\ [\ \vdash \Gamma] \Rightarrow [\Gamma \vdash T] \Rightarrow [\Gamma \vdash t \triangleleft T] \Rightarrow P\ \Gamma\ t\ T$
by induction on the last premise, you get extra help in induction steps.

Shown once and for all, used virtually everywhere.

Nail IV: Type systems equivalence

Jww. K. Maillard, T. Laurent

Subsumptive and coercive subtyping

$$\textsc{Sub}\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \preceq T'}{\Gamma \vdash t : T'} \qquad \text{vs} \qquad \textsc{Coe}\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \preceq T'}{\Gamma \vdash \mathrm{coe}_{T,T'}\ t : T'}$$

$$\textsc{Sub}\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \preceq T'}{\Gamma \vdash t : T'}$$

Good for users

vs

$$\textsc{Coe}\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \preceq T'}{\Gamma \vdash \mathrm{coe}_{T,T'}\ t : T'}$$

Good for meta-theory

Subsumptive and coercive subtyping

$$\textsc{Sub}\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \preceq T'}{\Gamma \vdash t : T'}$$

vs

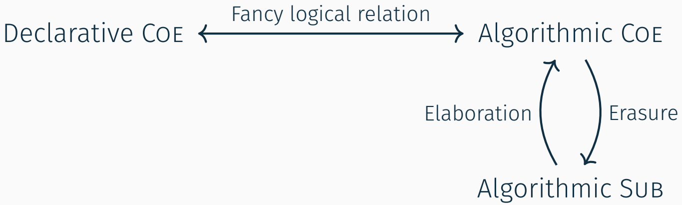
$$\textsc{Coe}\ \frac{\Gamma \vdash t : T \qquad \Gamma \vdash T \preceq T'}{\Gamma \vdash \mathrm{coe}_{T,T'}\ t : T'}$$

Good for users

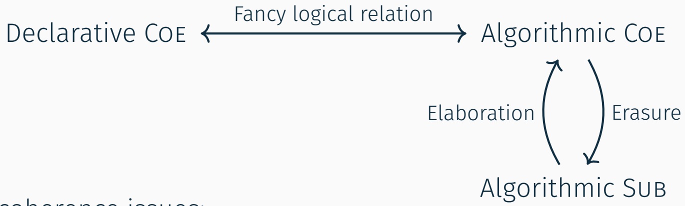
Good for meta-theory

Subtle coherence issues…

Squashing coherences

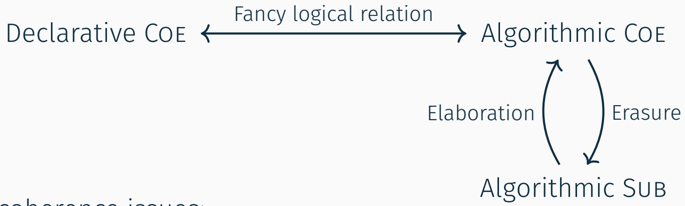


Squashing coherences



Key to sidestep coherence issues:

- New equations for coe
- uniqueness of (conversion) derivations



Key to sidestep coherence issues:

- **New equations** for coe
- **uniqueness** of (conversion) derivations

Not just for subtyping: typed vs untyped conversion is similar...

Building a good hammer: On annotations

Jww. N. Krishnaswami

Different kind of annotations

How to design a complete bidirectional type system?

Different kind of annotations

How to design a complete bidirectional type system?

Solution 1:
Annotations

$\lambda\, x \colon A.\, t$

Coq, Lean…

All terms infer

Different kind of annotations

How to design a complete bidirectional type system?

Solution 1: Annotations	Solution 2: Restricted terms
$\lambda\, x\colon A.\, t$	$\lambda\, x.\, t$
Coq, Lean...	Agda...
All terms infer	Neutrals infer Normal forms check

How to design a complete bidirectional type system?

Solution 1: Annotations	Solution 2: Restricted terms	Solution 3: Free-standing annotations
$\lambda\, x : A.\, t$	$\lambda\, x.\, t$	$\lambda\, x.\, t$ and $t :: A$
Coq, Lean…	Agda…	Conor, RED* family…
All terms infer	Neutrals infer Normal forms check	Inferring terms Checking terms

Different kind of annotations

How to design a complete bidirectional type system?

Solution 1: Annotations	Solution 2: Restricted terms	Solution 3: Free-standing annotations
$\lambda\, x : A.\, t$	$\lambda\, x.\, t$	$\lambda\, x.\, t$ and $t :: A$
Coq, Lean...	Agda...	Conor, red* family...
All terms infer	Neutrals infer Normal forms check	Inferring terms Checking terms

Can we design a single system, with a single completeness proof?

$$c, A, B ::= \underline{i} \mid \square_k \mid \Pi x{:}\, A.B \mid \lambda\, x.\, c$$

$$i ::= c :: A \mid x \mid i\ c \mid \lambda\, x{:}\, A.\, i$$

A proposition

$$c, A, B ::= \underline{i} \mid \Box_k \mid \Pi x \colon A.B \mid \lambda\, x.\, c$$

$$i ::= c :: A \mid x \mid i\ c \mid \lambda\, x \colon A.\, i$$

Solution 1

Solution 2

$$c, A, B ::= \underline{i} \mid \square_k \mid \Pi x{:}\, A.B \mid \lambda\, x.\, c$$

$$i ::= c :: A \mid x \mid i\, c \mid \lambda\, x{:}\, A.\, i$$

A proposition

Solution 3

$$c, A, B ::= \underline{i} \mid \square_k \mid \Pi x{:}\, A.B \mid \lambda\, x.\, c$$

$$i ::= c :: A \mid x \mid i\; c \mid \lambda\, x{:}\, A.\, i$$

$$c, A, B ::= \underline{i} \mid \square_k \mid \Pi x\colon A.B \mid \lambda\, x.\, c$$

$$i ::= c :: A \mid x \mid i\ c \mid \lambda\, x\colon A.\, i$$

Complete, by construction.

A proposition

$$c, A, B ::= \underline{i} \mid \square_k \mid \Pi x\colon A.B \mid \lambda\, x.\, c \qquad \mid \Sigma\, x\colon A.B \mid \langle c, c \rangle \mid \mathbf{W}\, x\colon A.B \mid \mathrm{sup}(c, c)$$

$$i ::= c :: A \mid x \mid i\; c \mid \lambda\, x\colon A.\, i \qquad \mid i.{}_1 \mid i.{}_2 \mid \langle i, c \rangle_{x.B} \mid \mathrm{ind}_{\mathbf{W}}(i; x.A; c) \mid \mathrm{sup}_{x.B}(i, c)$$

Complete, by construction... and extends nicely.

Type annotations reduce (see observational equality, cast calculus, coercions...):

$$((\lambda\, x.\, t) :: \Pi\, x : A.B)\; u \to (\lambda\, x : A.\, (t :: B))\; u \to (t[u :: A]) :: B[u :: A]$$

Type annotations reduce (see observational equality, cast calculus, coercions…):

$$((\lambda\, x.\, t) :: \Pi\, x : A.B)\; u \to (\lambda\, x : A.\, (t :: B))\; u \to (t[u :: A]) :: B[u :: A]$$

Plays natively well with bidirectional conversion:

$$\Gamma \vdash A \cong A' \qquad \text{and} \qquad \Gamma \vdash c \cong c' \triangleleft A \qquad \text{but} \qquad \Gamma \vdash n \approx n' \triangleright A$$

Type annotations reduce (see observational equality, cast calculus, coercions...):

$$((\lambda\, x.\, t) :: \Pi\, x{:}\, A.B)\; u \to (\lambda\, x{:}\, A.\, (t :: B))\; u \to (t[u :: A]) :: B[u :: A]$$

Plays natively well with bidirectional conversion:

$$\Gamma \vdash A \cong A' \qquad \text{and} \qquad \Gamma \vdash c \cong c' \lhd A \qquad \text{but} \qquad \Gamma \vdash n \approx n' \rhd A$$

(Stuck) annotations can/should be ignored ($\mathrm{TT}^{\mathrm{obs}}$ again):

$$\frac{\Gamma \vdash n \approx n' \rhd A}{\Gamma \vdash \underline{n} :: A' \approx n' \rhd A'}$$

Wrapping up

Wrapping up

Bidirectional typing is good for meta-theory

- Control over conversion
- Unique, well-behaved derivations
- Always ready for implementation

Wrapping up

Bidirectional typing is good for meta-theory

- Control over conversion
- Unique, well-behaved derivations
- Always ready for implementation

What now?

- What kind of annotations do we want?
- Algorithmic or semi-algorithmic conversion?
- A bidirectional logical framework?

$$\frac{\Gamma \vdash t \triangleright T \qquad \Gamma \vdash T \cong T'}{\Gamma \vdash t \triangleleft T'}$$

$$\frac{\Gamma \vdash t \triangleright T \qquad T \to^{\star} \square_i}{\Gamma \vdash t \triangleright_{\square} \square_i}$$

$$\frac{\Gamma \vdash t \triangleright T \qquad T \to^{\star} \prod x : A . B}{\Gamma \vdash t \triangleright_{\Pi} \prod x : A . B}$$

THANK YOU!

(AND LET'S TALK!)

$$\frac{\Gamma, x : A \vdash f\ x \cong g\ x \triangleleft B}{\Gamma \vdash f \cong_{\mathrm{h}} g \triangleleft \prod x : A .\ B}$$

$$\frac{\Gamma \vdash m \approx n \triangleright_{\Pi} \prod x : A .\ B \qquad \Gamma \vdash t \cong u \triangleleft A}{\Gamma \vdash m\,t \approx n\,u \triangleright B[t]}$$

Bibliography

[AÖV18] Andreas Abel, Joakim Öhman, and Andrea Vezzosi. "Decidability of Conversion for Type Theory in Type Theory". In: *Proc. ACM Program. Lang.* (Jan. 2018). DOI: 10.1145/3158111.

[SH12] Vincent Siles and Hugo Herbelin. "Pure Type System conversion is always typable". In: *J. Funct. Program.* 22.2 (2012), pp. 153–180. DOI: 10.1017/S0956796812000044.

[AC07] Andreas Abel and Thierry Coquand. "Untyped Algorithmic Equality for Martin-Löf's Logical Framework with Surjective Pairs". In: *Fundamenta Informaticae* 77.4 (2007). TLCA'05 special issue., pp. 345–395. URL: http://fi.mimuw.edu.pl/abs77.html#15.

[PT22] Loïc Pujet and Nicolas Tabareau. "Observational Equality: Now for Good". In: *Proc. ACM Program. Lang.* 6.POPL (Jan. 2022). DOI: 10.1145/3498693.

[Tak95] M. Takahashi. "Parallel Reductions in λ-Calculus". In: *Information and Computation* 118.1 (1995), pp. 120–127. ISSN: 0890-5401. DOI: 10.1006/inco.1995.1057. URL: https://www.sciencedirect.com/science/article/pii/S0890540185710577.