A Reasonably Gradual Dependent Type Theory

Kenji Maillard

Inria Nantes, team Gallinette

j.w.w.


Meven Lennon-Bertrand


Nicolas Tabareau


Éric Tanter

Types'22, Nantes
Thursday, 23rd of June, 2022

Prototyping with dependent types


```
Inductive vect (A : Type) : ℕ → Type :=
| nil : vect A 0
| _::_ {n : ℕ} (head : A) (tail : vect A n) : vect A (n+1)
```

Prototyping with dependent types


```
Inductive vect (A : Type) : ℕ → Type :=
| nil : vect A 0
| _::_ {n : ℕ} (head : A) (tail : vect A n) : vect A (n+1)
```

```
Equations filter {A : Type} (P : A → 𝔹) {len : ℕ} (l : vect A len) : vect A ? :=
```

Prototyping with dependent types


```
Inductive vect (A : Type) : ℕ → Type :=
| nil : vect A 0
| _ :: _ {n : ℕ} (head : A) (tail : vect A n) : vect A (n+1)
```

```
Equations filter {A : Type} (P : A → 𝔹) {len : ℕ} (l : vect A len) : vect A ? :=
filter P nil := nil
filter P (head :: tail) when not (P head) := filter tail
filter P (head :: tail) := head :: (filter tail) : vect A ?
```

Prototyping with dependent types

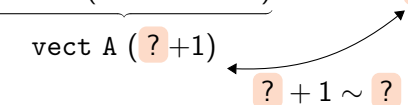

```
Inductive vect (A : Type) : ℕ → Type :=
| nil : vect A 0
| _ :: _ {n : ℕ} (head : A) (tail : vect A n) : vect A (n+1)
```

```
Equations filter {A : Type} (P : A → 𝔹) {len : ℕ} (l : vect A len) : vect A ? :=
filter P nil := nil
filter P (head :: tail) when not (P head) := filter tail
filter P (head :: tail) := head :: (filter tail) : vect A ?
                                                    vect A (? +1)
```

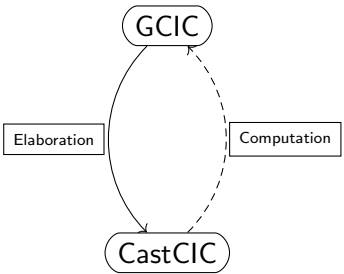
Prototyping with dependent types


```
Inductive vect (A : Type) : ℕ → Type :=
| nil : vect A 0
| _ :: _ {n : ℕ} (head : A) (tail : vect A n) : vect A (n+1)
```

```
Equations filter {A : Type} (P : A → 𝔹) {len : ℕ} (l : vect A len) : vect A ? :=
filter P nil := nil
filter P (head :: tail) when not (P head) := filter tail
filter P (head :: tail) := head :: (filter tail) : vect A ?
```

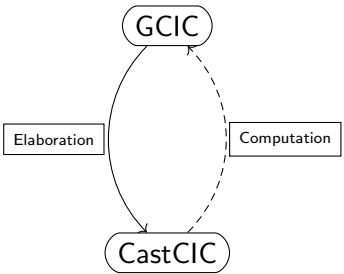


GCIC & CastCIC: a marriage of Gradual & Dependent Types



- Unknown terms and types $?_A : A$
- Casts $t : A \Longrightarrow \langle B \Leftarrow A \rangle\, t : B$
- Errors $\mathtt{err}_A : A$

GCIC & CastCIC: a marriage of Gradual & Dependent Types

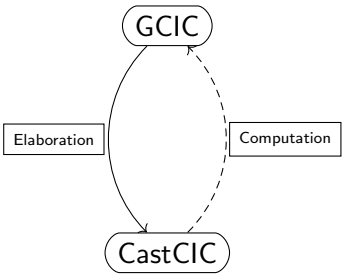


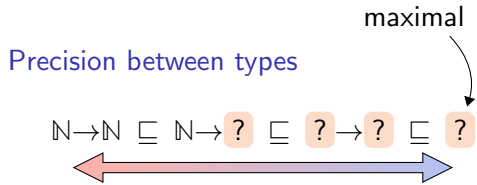
Precision between types

$\mathbb{N} \to \mathbb{N} \sqsubseteq \mathbb{N} \to ? \sqsubseteq ? \to ? \sqsubseteq ?$

- Unknown terms and types $?_A : A$
- Casts $t : A \Longrightarrow \langle B \Leftarrow A \rangle\, t : B$
- Errors $\mathtt{err}_A : A$

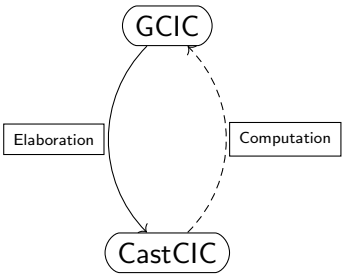
GCIC & CastCIC: a marriage of Gradual & Dependent Types





- Unknown terms and types $?_A : A$
- Casts $t : A \Longrightarrow \langle B \Leftarrow A \rangle\, t : B$
- Errors $\mathtt{err}_A : A$

GCIC & CastCIC: a marriage of Gradual & Dependent Types



- Unknown terms and types $?_A : A$
- Casts $t : A \Longrightarrow \langle B \Leftarrow A \rangle\, t : B$
- Errors $\mathtt{err}_A : A$

maximal

Precision between types

$\mathbb{N} \to \mathbb{N} \sqsubseteq \mathbb{N} \to ? \sqsubseteq ? \to ? \sqsubseteq ?$

Precision enforces cast validity:

$$A \sqsubseteq B \implies A \underset{\langle A \Leftarrow B \rangle}{\overset{\langle B \Leftarrow A \rangle}{\rightleftarrows}} B$$

Graduality [New & Ahmed 2018]

The Fire Triangle of Graduality

Conservativity over CIC +
Gradual properties

$?_{\square} : \square$

The Fire Triangle of Graduality

Conservativity over CIC +
Gradual properties

$$?_{\square} : \square$$

$$?_{\square} \to ?_{\square} : \square$$

The Fire Triangle of Graduality

Conservativity over CIC +
Gradual properties

$$?_{\square} : \square$$

$$?_{\square} \to ?_{\square} : \square$$

$$?_{\square} \to ?_{\square} \quad \sqsubseteq \quad ?_{\square}$$

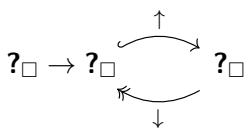
The Fire Triangle of Graduality

Conservativity over CIC +
Gradual properties

$$?_\square : \square$$

$$?_\square \to ?_\square : \square$$

$$?_\square \to ?_\square \quad \sqsubseteq \quad ?_\square$$



The Fire Triangle of Graduality

Conservativity over CIC +
Gradual properties

$$?_\square : \square$$

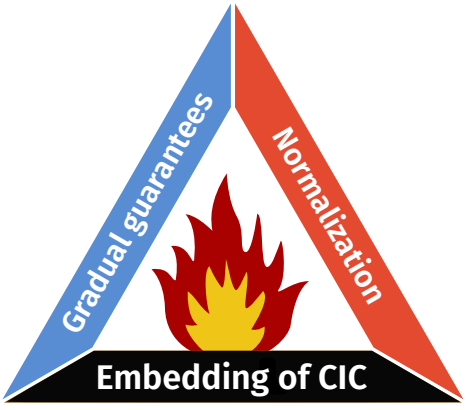
$$?_\square \to ?_\square : \square$$

$$?_\square \to ?_\square \quad \sqsubseteq \quad ?_\square$$

$$?_\square \to ?_\square \quad \uparrow \downarrow \quad ?_\square$$

$$\delta : ?_\square \to ?_\square := \lambda x : ?_\square. \downarrow x\, x$$

$$\Omega : ?_\square := \delta \uparrow \delta$$



The Fire Triangle of Graduality

Conservativity over CIC + Gradual properties

$$?_\square : \square$$

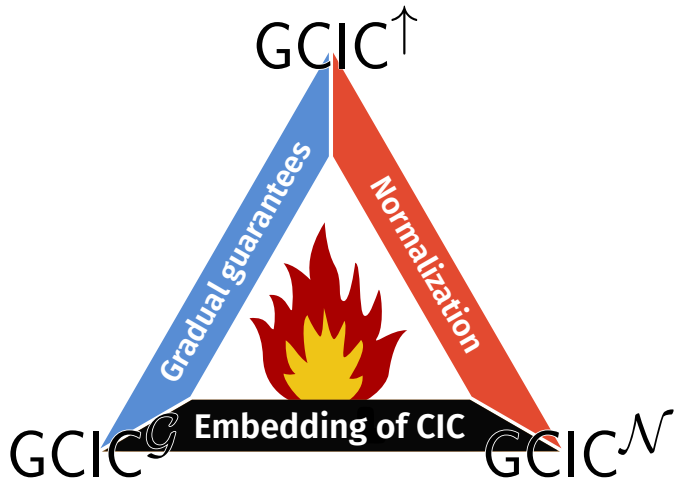
$$?_\square \to ?_\square : \square$$

$$?_\square \to ?_\square \quad \sqsubseteq \quad ?_\square$$

$$?_\square \to ?_\square \quad \uparrow \downarrow \quad ?_\square$$

$$\delta : ?_\square \to ?_\square := \lambda x : ?_\square . \downarrow x\, x$$

$$\Omega : ?_\square := \delta \uparrow \delta$$



Internal precision in GRIP

$\mathsf{CastCIC}^{\mathcal{N}}$: Normalizing, conservative over CIC

How far is $\mathsf{CastCIC}^{\mathcal{N}}$ from being gradual ?

Internal precision in GRIP

$\mathsf{CastCIC}^{\mathcal{N}}$: Normalizing, conservative over CIC

$$\mathsf{GRIP} = \mathsf{CastCIC}^{\mathcal{N}} + \sqsubseteq$$

$$\frac{\Gamma \vdash A, B : \square_i}{\Gamma \vdash A \sqsubseteq_i B : \mathbb{P}} \qquad \frac{\Gamma \vdash A, B : \square_i \qquad \Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash t \ {}_A\sqsubseteq_B \ u : \mathbb{P}}$$

$\mathbb{P}$: Pure universe of propositions

Internal precision in GRIP

$\mathsf{CastCIC}^{\mathcal{N}}$: Normalizing, conservative over CIC

$$\mathsf{GRIP} = \mathsf{CastCIC}^{\mathcal{N}} + \sqsubseteq$$

$$\frac{\Gamma \vdash A, B : \square_i}{\Gamma \vdash A \sqsubseteq_i B : \mathbb{P}} \qquad \frac{\Gamma \vdash A, B : \square_i \qquad \Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash t \ {}_A\sqsubseteq_B \ u : \mathbb{P}}$$

$\mathbb{P}$: Pure universe of propositions

no error, no casts, no ?

definitionally proof-irrelevant

Proof of precision – An example

$$\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad \mathbb{N}\to\mathbb{N} \sqsubseteq_{\texttt{?}_\square \to \mathbb{N}} \qquad \texttt{add1?} := \lambda x : \texttt{?}_\square.(\langle \mathbb{N} \Leftarrow \texttt{?}_\square \rangle\, x) + 1$$

Proof of precision – An example

$$\frac{\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad {}_{\mathbb{N}\to\mathbb{N}}\sqsubseteq_{?_\square\to\mathbb{N}} \qquad \texttt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle\, z) + 1}{\forall x\, y,\ x\ {}_{\mathbb{N}}\sqsubseteq_{?_\square}\ z \quad \Longrightarrow \quad x + 1\ {}_{\mathbb{N}}\sqsubseteq_{\mathbb{N}}\ (\langle \mathbb{N} \Leftarrow ?_\square \rangle\, z) + 1}$$

Proof of precision – An example

$$\frac{A \sqsubseteq B}{a \; {}_A\!\sqsubseteq_B \; b \iff a \; {}_A\!\sqsubseteq_A \langle A \Leftarrow B \rangle \; b}$$

$$\frac{\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad {}_{\mathbb{N}\to\mathbb{N}}\!\sqsubseteq_{?_\square \to \mathbb{N}} \qquad \texttt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle \; z) + 1}{\forall x \, y, \; x \; {}_{\mathbb{N}}\!\sqsubseteq_{?_\square} \; z \quad \Longrightarrow \quad x + 1 \; {}_{\mathbb{N}}\!\sqsubseteq_{\mathbb{N}} \; (\langle \mathbb{N} \Leftarrow ?_\square \rangle \; z) + 1}$$

Proof of precision – An example

$$\frac{A \sqsubseteq B}{a \ {}_A\!\sqsubseteq_B \ b \iff a \ {}_A\!\sqsubseteq_A \langle A \Leftarrow B \rangle \ b}$$

$$\frac{\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad {}_{\mathbb{N}\to\mathbb{N}}\!\sqsubseteq_{?_\square\to\mathbb{N}} \qquad \texttt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}{\forall x\, y,\ x \ {}_{\mathbb{N}}\!\sqsubseteq_{?_\square} \ z \quad \Longrightarrow \quad x + 1 \ {}_{\mathbb{N}}\!\sqsubseteq_{\mathbb{N}} \ (\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}$$

Reasoning principles for $\sqsubseteq$

- (heterogeneous) transitivity,
- quasi-reflexivity,
- adjunction properties,
- decomposition through upper bounds.

Proof of precision – An example

$$\frac{A \sqsubseteq B}{a \ {}_A\!\sqsubseteq_B \ b \iff a \ {}_A\!\sqsubseteq_A \langle A \Leftarrow B \rangle \ b}$$

$$\frac{\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad \mathbb{N} \to \mathbb{N} \sqsubseteq_{?_\square \to \mathbb{N}} \qquad \texttt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}{\forall x \, y, \ x \ {}_{\mathbb{N}}\!\sqsubseteq_{?_\square} \ z \quad \Longrightarrow \quad x + 1 \ {}_{\mathbb{N}}\!\sqsubseteq_{\mathbb{N}} \ (\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}$$

Reasoning principles for $\sqsubseteq$

- ▷ (heterogeneous) transitivity,
- ▷ quasi-reflexivity,
- ▷ adjunction properties,
- ▷ decomposition through upper bounds.

However

- ▷ $?_{\square_i} \to ?_{\square_i} \not\sqsubseteq_i ?_{\square_i}$

Proof of precision – An example

$$\frac{A \sqsubseteq B}{a \ {}_A\sqsubseteq_B \ b \iff a \ {}_A\sqsubseteq_A \langle A \Leftarrow B \rangle \ b}$$

$$\frac{\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad {}_{\mathbb{N}\to\mathbb{N}}\sqsubseteq_{?_\square\to\mathbb{N}} \qquad \texttt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}{\forall x\, y,\ x \ {}_{\mathbb{N}}\sqsubseteq_{?_\square} \ z \quad \Longrightarrow \quad x + 1 \ {}_{\mathbb{N}}\sqsubseteq_{\mathbb{N}} \ (\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}$$

Reasoning principles for $\sqsubseteq$

- ▷ (heterogeneous) transitivity,
- ▷ quasi-reflexivity,
- ▷ adjunction properties,
- ▷ decomposition through upper bounds.

However

- ▷ $?_{\square_i} \to ?_{\square_i} \not\sqsubseteq_i ?_{\square_i}$
- ▷ $?_{\square_i}$ maximal for ${}_{\square_i}\sqsubseteq_{\square_i}$, not for $\sqsubseteq_i$

Proof of precision – An example

$$\frac{A \sqsubseteq B}{a \ {}_A\!\sqsubseteq_B \ b \iff a \ {}_A\!\sqsubseteq_A \langle A \Leftarrow B \rangle \ b}$$

$$\frac{\mathtt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad {}_{\mathbb{N}\to\mathbb{N}}\!\sqsubseteq_{?_\square \to \mathbb{N}} \qquad \mathtt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}{\forall x \, y, \ x \ {}_{\mathbb{N}}\!\sqsubseteq_{?_\square} \ z \quad \Longrightarrow \quad x + 1 \ {}_{\mathbb{N}}\!\sqsubseteq_{\mathbb{N}} \ (\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}$$

Reasoning principles for $\sqsubseteq$

- ▷ (heterogeneous) transitivity,
- ▷ quasi-reflexivity,
- ▷ adjunction properties,
- ▷ decomposition through upper bounds.

However

- ▷ $?_{\square_i} \to ?_{\square_i} \not\sqsubseteq_i ?_{\square_i}$
- ▷ $?_{\square_i}$ maximal for ${}_{\square_i}\!\sqsubseteq_{\square_i}$, not for $\sqsubseteq_i$
- ▷ $?_{\square_i} \to ?_{\square_i}$ not reflexive for ${}_{\square_i}\!\sqsubseteq_{\square_i}$

Proof of precision – An example

$$\frac{A \sqsubseteq B}{a \ {}_A\sqsubseteq_B \ b \iff a \ {}_A\sqsubseteq_A \langle A \Leftarrow B \rangle \ b}$$

$$\frac{\texttt{add1} := \lambda x : \mathbb{N}.x + 1 \qquad {}_{\mathbb{N}\to\mathbb{N}}\sqsubseteq_{?_\square \to \mathbb{N}} \qquad \texttt{add1?} := \lambda z : ?_\square.(\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}{\forall x \, y, \ x \ {}_{\mathbb{N}}\sqsubseteq_{?_\square} \ z \quad \Longrightarrow \quad x + 1 \ {}_{\mathbb{N}}\sqsubseteq_{\mathbb{N}} \ (\langle \mathbb{N} \Leftarrow ?_\square \rangle \ z) + 1}$$

Reasoning principles for $\sqsubseteq$

- ▷ (heterogeneous) transitivity,
- ▷ quasi-reflexivity,
- ▷ adjunction properties,
- ▷ decomposition through upper bounds.

However

- ▷ $?_{\square_i} \to ?_{\square_i} \not\sqsubseteq_i ?_{\square_i}$
- ▷ $?_{\square_i}$ maximal for ${}_{\square_i}\sqsubseteq_{\square_i}$, not for $\sqsubseteq_i$
- ▷ $?_{\square_i} \to ?_{\square_i}$ not reflexive for ${}_{\square_i}\sqsubseteq_{\square_i}$
- ▷ $\iota(?_{\square_i} \to ?_{\square_i}) \sqsubseteq_{i+1} ?_{\square_{i+1}}$

How good is GRIP?

Consistency $\nvdash_{\mathsf{GRIP}} t : \bot$ (where $\bot : \mathbb{P}$)

Idea: Build a model of GRIP in MLTT+SProp using partial pre-orders.
$[\![\bot]\!]$ is empty in the model.

How good is GRIP?

Consistency $\not\vdash_{\mathsf{GRIP}} t : \bot$ (where $\bot : \mathbb{P}$)

Idea: Build a model of GRIP in MLTT+SProp using partial pre-orders.
$[\![\bot]\!]$ is empty in the model.

Normalization

Idea: The translation $[\![-]\!]$ preserves reduction sequences into a normalizing target (MLTT + SProp).

How good is GRIP?

Consistency $\not\vdash_{\mathsf{GRIP}} t : \bot$ (where $\bot : \mathbb{P}$)

Idea: Build a model of GRIP in MLTT+SProp using partial pre-orders.
$[\![\bot]\!]$ is empty in the model.

Normalization

Idea: The translation $[\![-]\!]$ preserves reduction sequences into a normalizing target (MLTT + SProp).

Gradual fragment $\mathsf{GRIP}^{\uparrow}$

Idea: Restrict catch to match and raise universe levels on Π

$$\frac{\Gamma \vdash_{\mathsf{GRIP}^{\uparrow}} A : \square_i \qquad \Gamma, x : A \vdash_{\mathsf{GRIP}^{\uparrow}} B : \square_i}{\Gamma \vdash_{\mathsf{GRIP}^{\uparrow}} \Pi\, x : A.B : \square_{i+1}}$$

Summary

GRIP = partially gradual type theory + internal precision

- ▷ More details in the paper `https://hal.inria.fr/hal-03596652`
- ▷ Proof of concept for GRIP in Agda
- ▷ Partial pre-order based model in Coq

Summary

GRIP = partially gradual type theory + internal precision

- More details in the paper `https://hal.inria.fr/hal-03596652`
- Proof of concept for GRIP in Agda
- Partial pre-order based model in Coq

GRIP is a type theory, not yet a proof assistant:

- Elaboration from a gradual source
- General inductive types
- An actual implementation ?

```
kyod@takasomen ~/W/G/G/GCIC (load-in-subdir)> esy repl
> def delta(x : ?T0) : ?T0 = (x : ?T0 -> ?T0) x;;
delta : Π(x : ?□0). ?□0 defined.
> def omega : ?T0 = delta (delta : ?T0);;
omega : ?□0 defined.
> eval omega;;
err_?_□0
> set variant G;;
OK
> eval omega;;
[reduction_error] not enough fuel
```

WIP GCIC prototype
(T. Díaz, S. Malewski, E. Tanter)