Continuity in dependent type theory

joint work with Martin Baillon, Pierre-Marie Pédrot

Assia Mahboubi

HoTTest seminar – March 31th 2022

Inria, LS2N, Université de Nantes, Vrije Universiteit Amsterdam

Continuity of definable functions in a constructive setting

Theorem (folklore?)

Every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ *definable in Gödel's system* T *is continuous.*

Continuity of definable functions in a constructive setting

Theorem (folklore?)

Every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ *definable in Gödel's system* T *is continuous.*

with Gödel's system T = simply typed λ-calculus + $\mathbb{N}$ + recursor.

Continuity of definable functions in a constructive setting

Theorem (folklore?)

Every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ *definable in Gödel's system* T *is continuous.*

with Gödel's system T = simply typed λ-calculus + $\mathbb{N}$ + recursor.

[Foundations of Constructive Mathematics. M.J. Beeson. Springer, 1985]

Continuity of definable functions in a constructive setting

Theorem

Every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ *definable in* $\mathcal{S}$ *is continuous.*

Continuity of definable functions in a constructive setting

Theorem

Every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ definable in $\mathcal{S}$ is continuous.

with $\mathcal{S}$ a dependent type theory.

Continuity of definable functions in a constructive setting

Theorem

Every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ *definable in* $\mathcal{S}$ *is continuous.*

with $\mathcal{S}$ a dependent type theory.

Generalizing [Continuity of Gödel's system T functionals via effectful forcing. M. Escardó. MFPS'2013.]

Local continuity

A function $f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is continuous at $\alpha : \mathbb{N} \to \mathbb{N}$ if:

$$\forall W \in \mathcal{V}_{\mathbb{N}}(f\ \alpha), \exists V \in \mathcal{V}_{\mathbb{N}\to\mathbb{N}}(f), \forall \alpha' : \mathbb{N} \to \mathbb{N}, \quad \alpha' \in V \Rightarrow (f\ \alpha') \in W.$$

Local continuity

A function $f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is continuous at $\alpha : \mathbb{N} \to \mathbb{N}$ if:

$$\forall W \in \mathcal{V}_{\mathbb{N}}(f\ \alpha), \exists V \in \mathcal{V}_{\mathbb{N}\to\mathbb{N}}(f), \forall \alpha' : \mathbb{N} \to \mathbb{N}, \quad \alpha' \in V \Rightarrow (f\ \alpha') \in W.$$

With:

- Discrete topology on $\mathbb{N}$
- Product topology on $\mathbb{N} \to \mathbb{N}$

This becomes:

$$\exists w : \text{list } \mathbb{N}, \forall \alpha' : \mathbb{N} \to \mathbb{N}, \quad (\alpha'[w] = \alpha[w]) \Rightarrow (f\ \alpha') = (f\ \alpha).$$

Dependent type theories

- CC_ω : a predicative variant of CIC, with dependent pairs
- Identity types, à la MLTT
- Inductive types with parameters and indices

CC_ω with dependent pairs: typing rules

$$A, B, M, N ::= \square_i \mid x \mid M\ N \mid \lambda x : A.\ M \mid \Pi x : A.\ M \mid \Sigma x : A.\ B \mid M.\pi_1 \mid M.\pi_2 \mid (M, N)$$

$$\Gamma, \Delta ::= \cdot \mid \Gamma, x : A$$

CC_ω with dependent pairs: typing rules

$$A, B, M, N ::= \square_i \mid x \mid M\ N \mid \lambda x : A.\ M \mid \Pi x : A.\ M \mid \Sigma x : A.\ B \mid M.\pi_1 \mid M.\pi_2 \mid (M, N)$$

$$\Gamma, \Delta ::= \cdot \mid \Gamma, x : A$$

$$\frac{}{\vdash \cdot} \qquad \frac{\Gamma \vdash A : \square_i}{\vdash \Gamma, x : A} \qquad \frac{\vdash \Gamma \qquad (x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\vdash \Gamma \qquad i < j}{\Gamma \vdash \square_i : \square_j}$$

CC_ω with dependent pairs: typing rules

$$A, B, M, N ::= \square_i \mid x \mid M\ N \mid \lambda x : A.\, M \mid \Pi x : A.\, M \mid \Sigma x : A.\, B \mid M.\pi_1 \mid M.\pi_2 \mid (M, N)$$

$$\Gamma, \Delta ::= \cdot \mid \Gamma, x : A$$

$$\frac{}{\vdash \cdot} \qquad \frac{\Gamma \vdash A : \square_i}{\vdash \Gamma, x : A} \qquad \frac{\vdash \Gamma \qquad (x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\vdash \Gamma \qquad i < j}{\Gamma \vdash \square_i : \square_j}$$

$$\frac{\Gamma \vdash A : \square_i \qquad \Gamma \vdash M : B}{\Gamma, x : A \vdash M : B} \qquad \frac{\Gamma \vdash A : \square_i \qquad \Gamma, x : A \vdash B : \square_j}{\Gamma \vdash \Pi x : A.\, B : \square_{\max(i,j)}}$$

$$\frac{\Gamma \vdash M : \Pi x : A.\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B\{x := N\}} \qquad \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\, B : \square_i}{\Gamma \vdash \lambda x : A.\, M : \Pi x : A.\, B}$$

CC_ω with dependent pairs: typing rules

$$A, B, M, N ::= \Box_i \mid x \mid M\ N \mid \lambda x : A.\ M \mid \Pi x : A.\ M \mid \Sigma x : A.\ B \mid M.\pi_1 \mid M.\pi_2 \mid (M, N)$$

$$\Gamma, \Delta ::= \cdot \mid \Gamma, x : A$$

$$\frac{}{\vdash \cdot} \qquad \frac{\Gamma \vdash A : \Box_i}{\vdash \Gamma, x : A} \qquad \frac{\vdash \Gamma \qquad (x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\vdash \Gamma \qquad i < j}{\Gamma \vdash \Box_i : \Box_j}$$

$$\frac{\Gamma \vdash A : \Box_i \qquad \Gamma \vdash M : B}{\Gamma, x : A \vdash M : B} \qquad \frac{\Gamma \vdash A : \Box_i \qquad \Gamma, x : A \vdash B : \Box_j}{\Gamma \vdash \Pi x : A.\ B : \Box_{\max(i,j)}}$$

$$\frac{\Gamma \vdash M : \Pi x : A.\ B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B\{x := N\}} \qquad \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\ B : \Box_i}{\Gamma \vdash \lambda x : A.\ M : \Pi x : A.\ B}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \Box_i \qquad \Gamma \vdash A \equiv B}{\Gamma \vdash M : B}$$

CC_ω with dependent pairs: typing rules

$$A, B, M, N ::= \square_i \mid x \mid M\ N \mid \lambda x : A.\ M \mid \Pi x : A.\ M \mid \Sigma x : A.\ B \mid M.\pi_1 \mid M.\pi_2 \mid (M, N)$$

$$\Gamma, \Delta ::= \cdot \mid \Gamma, x : A$$

$$\frac{}{\vdash \cdot} \qquad \frac{\Gamma \vdash A : \square_i}{\vdash \Gamma, x : A} \qquad \frac{\vdash \Gamma \qquad (x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\vdash \Gamma \qquad i < j}{\Gamma \vdash \square_i : \square_j}$$

$$\frac{\Gamma \vdash A : \square_i \qquad \Gamma \vdash M : B}{\Gamma, x : A \vdash M : B} \qquad \frac{\Gamma \vdash A : \square_i \qquad \Gamma, x : A \vdash B : \square_j}{\Gamma \vdash \Pi x : A.\ B : \square_{\max(i,j)}}$$

$$\frac{\Gamma \vdash M : \Pi x : A.\ B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B\{x := N\}} \qquad \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A.\ B : \square_i}{\Gamma \vdash \lambda x : A.\ M : \Pi x : A.\ B}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \square_i \qquad \Gamma \vdash A \equiv B}{\Gamma \vdash M : B}$$

$$\frac{\Gamma \vdash A : \square_i \qquad \Gamma, x : A \vdash B : \square_j}{\Gamma \vdash \Sigma x : A.\ B : \square_{\max(i,j)}} \qquad \frac{\Gamma \vdash M : \Sigma x : A.\ B}{\Gamma \vdash M.\pi_1 : A}$$

$$\frac{\Gamma \vdash M : \Sigma x : A.\ B}{\Gamma \vdash M.\pi_2 : B\{x := M.\pi_1\}}$$

Inductive types in CIC

$$\texttt{Inductive}\ \mathbb{N} := \mathsf{O} : \mathbb{N} \mid \mathsf{S} : \mathbb{N} \to \mathbb{N}$$

Inductive types in CIC

$$\texttt{Inductive}\ \mathbb{N} := \mathsf{O} : \mathbb{N} \mid \mathsf{S} : \mathbb{N} \to \mathbb{N}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathbb{N} : \Box_i} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{O} : \mathbb{N}} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{S} : \mathbb{N} \to \mathbb{N}}$$

Inductive types in CIC

$$\texttt{Inductive}\ \mathbb{N} := \mathsf{O} : \mathbb{N} \mid \mathsf{S} : \mathbb{N} \to \mathbb{N}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathbb{N} : \square_i} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{O} : \mathbb{N}} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{S} : \mathbb{N} \to \mathbb{N}}$$

$$\frac{\Gamma \vdash P : \mathbb{N} \to \square_i \qquad \Gamma \vdash t_{\mathsf{O}} : P\ \mathsf{O} \qquad \Gamma \vdash t_{\mathsf{S}} : \Pi n : \mathbb{N}.\ P\ n \to P\ (\mathsf{S}\ n)}{\Gamma \vdash \mathbb{N}_{\mathsf{ind}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}} : \Pi n : \mathbb{N}.\ P\ n}$$

Inductive types in CIC

$$\texttt{Inductive}\ \mathbb{N} := \mathsf{O} : \mathbb{N} \mid \mathsf{S} : \mathbb{N} \to \mathbb{N}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathbb{N} : \Box_i} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{O} : \mathbb{N}} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{S} : \mathbb{N} \to \mathbb{N}}$$

$$\frac{\Gamma \vdash P : \mathbb{N} \to \Box_i \qquad \Gamma \vdash t_{\mathsf{O}} : P\ \mathsf{O} \qquad \Gamma \vdash t_{\mathsf{S}} : \Pi n : \mathbb{N}.\, P\ n \to P\ (\mathsf{S}\ n)}{\Gamma \vdash \mathbb{N}_{\mathsf{ind}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}} : \Pi n : \mathbb{N}.\, P\ n}$$

$$\mathbb{N}_{\mathsf{ind}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}}\ \mathsf{O} \equiv t_{\mathsf{O}} \qquad \mathbb{N}_{\mathsf{ind}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}}\ (\mathsf{S}\ n) \equiv t_{\mathsf{S}}\ n\ (\mathbb{N}_{\mathsf{ind}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}}\ n)$$

Local continuity

$$\begin{array}{lcl} \mathscr{C} & : & ((\mathbb{N} \to \mathbb{N}) \to \mathbb{N}) \to \Box \\ \mathscr{C}\ f & := & \Pi(\alpha : \mathbb{N} \to \mathbb{N}).\, \Sigma(\ell : \mathsf{list}\ \mathbb{N}).\, \Pi(\beta : \mathbb{N} \to \mathbb{N}).\, \alpha \approx_\ell \beta \to f\ \alpha = f\ \beta. \end{array}$$

with

$$\alpha \approx_\ell \beta := \mathsf{map}\ \ell\ \alpha = \mathsf{map}\ \ell\ \beta$$

Wished theorem

Theorem

For any $\vdash_{\mathcal{S}} f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$*, there exists a proof* $\vdash_{\mathcal{T}} p : \mathscr{C}\ f$*.*

with $\mathcal{T}$ and $\mathcal{S}$ two "appropriate" dependent type theories.

Effectful computations

See:

$$\vdash f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$$

as a natural number computed using calls to a fixed oracle α:

$$\alpha : \mathbb{N} \to \mathbb{N} \vdash n : \mathbb{N}$$

Effectful computations

See:

$$\vdash f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$$

as a natural number computed using calls to a fixed oracle α:

$$\alpha : \mathbb{N} \to \mathbb{N} \vdash n : \mathbb{N}$$

More generally, we will study:

$$\alpha : \Pi i : \mathbb{I}, O\ i \vdash a : A$$

for fixed arbitrary types:

- of questions to the oracle: $\vdash \mathbb{I} : \square_0$
- of answers from the oracle: $\vdash O : \mathbb{I} \to \square_0$

Effectful computations

See:

$$\vdash f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$$

as a natural number computed using calls to a fixed oracle α:

$$\alpha : \mathbb{N} \to \mathbb{N} \vdash n : \mathbb{N}$$

More generally, we will study:

$$\alpha : \Pi i : \mathbb{I}, O\ i \vdash a : A$$

for fixed arbitrary types:

- of questions to the oracle: $\vdash \mathbb{I} : \square_0$
- of answers from the oracle: $\vdash O : \mathbb{I} \to \square_0$

Denote $\mathbb{Q} := \Pi i : \mathbb{I}, O\ i$ the type of oracles.

Dialogue trees

$$\texttt{Inductive}\ \mathfrak{D}\ (A : \square) : \square\ :=\ \eta : A \to \mathfrak{D}\ A\ \mid\ \beta : \Pi(i : \mathbb{I}).\, (\mathbb{O}\ i \to \mathfrak{D}\ A) \to \mathfrak{D}\ A.$$

Dialogue trees

$$\texttt{Inductive}\ \mathfrak{D}\ (A : \square) : \square\ :=\ \eta : A \to \mathfrak{D}\ A\ \mid\ \beta : \Pi(i : \mathbb{I}).\ (\mathbb{O}\ i \to \mathfrak{D}\ A) \to \mathfrak{D}\ A.$$

A term $\vdash d : \mathfrak{D}\ A$ represents a dialogue tree with:

- labels on inner nodes in $\mathbb{I}$;
- labels on leaves in A;
- arcs from inner node i indexed by $(\mathbb{O}\ i)$.

From dialogue trees to functions

A dialogue tree $\vdash d : \mathfrak{D}\ A$ shall compute a term in A using an oracle $\alpha : \mathbb{Q}$:

$$\begin{array}{lcl} \partial & : & \Pi\{A : \square\}\,(\alpha : \mathbb{Q})\,(d : \mathfrak{D}\ A).\,A \\ \partial\ \alpha\ (\eta\ x) & := & x \\ \partial\ \alpha\ (\beta\ i\ k) & := & \partial\ \alpha\ (k\ (\alpha\ i)). \end{array}$$

From dialogue trees to functions

A dialogue tree $\vdash d : \mathfrak{D}\ A$ shall compute a term in A using an oracle $\alpha : \mathbb{Q}$:

$$\begin{array}{lcl}\partial & : & \Pi\{A : \Box\}\,(\alpha : \mathbb{Q})\,(d : \mathfrak{D}\ A).\,A \\ \partial\ \alpha\ (\eta\ x) & := & x \\ \partial\ \alpha\ (\beta\ i\ k) & := & \partial\ \alpha\ (k\ (\alpha\ i)).\end{array}$$

Definition

A function $f : \mathbb{Q} \to A$ is said to be *eloquent* if there is a dialogue tree $d : \mathfrak{D}\ A$ and a proof that $\Pi\alpha : \mathbb{Q}.\ f\ \alpha = \partial\ \alpha\ d$.

From dialogue trees to functions

A dialogue tree $\vdash d : \mathfrak{D}\ A$ shall compute a term in A using an oracle $\alpha : \mathbb{Q}$:

$$\begin{array}{lcl} \partial & : & \Pi\{A : \square\}\,(\alpha : \mathbb{Q})\,(d : \mathfrak{D}\ A).\,A \\ \partial\ \alpha\ (\eta\ x) & := & x \\ \partial\ \alpha\ (\beta\ i\ k) & := & \partial\ \alpha\ (k\ (\alpha\ i)). \end{array}$$

Definition

A function $f : \mathbb{Q} \to A$ is said to be *eloquent* if there is a dialogue tree $d : \mathfrak{D}\ A$ and a proof that $\Pi\alpha : \mathbb{Q}.\ f\ \alpha = \partial\ \alpha\ d$.

Remark: Every eloquent function f is continuous.

Proof. Let d_f be a dialogue tree associated with f, and $\alpha : \mathbb{Q}$. Now α selects a path in d_f from the root to a leaf. Consider ℓ_α : list $\mathbb{I}$ the corresponding list of labels.

From functions to dialogue trees

Theorem

For any $\vdash_{\mathcal{S}} f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$, *there exist a proof* $\vdash_{\mathcal{T}} p : \mathscr{C}\ f$.

Proof. Construct a model of $\mathcal{S}$ for which which every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is eloquent.

From functions to dialogue trees

Theorem

For any $\vdash_{\mathcal{S}} f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$, *there exist a proof* $\vdash_{\mathcal{T}} p : \mathscr{C}\ f$.

Proof. Construct a model of $\mathcal{S}$ in $\mathcal{T}$, for which which every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is eloquent.

Syntactic models

For $\mathcal{S}$ and $\mathcal{T}$ two dependent type theory, a syntactic model of $\mathcal{S}$ in $\mathcal{T}$ is:

- a translation $[\_]$ of terms of $\mathcal{S}$ into terms of $\mathcal{T}$;
- a translation $[\![\_]\!]$ of types of $\mathcal{S}$ into types of $\mathcal{T}$;
- a translation $[\![\_]\!]$ of contexts of $\mathcal{S}$ into contexts of $\mathcal{T}$;

Syntactic models

For $\mathcal{S}$ and $\mathcal{T}$ two dependent type theory, a syntactic model of $\mathcal{S}$ in $\mathcal{T}$ is:

- a translation $[\_]$ of terms of $\mathcal{S}$ into terms of $\mathcal{T}$;
- a translation $[\![\_]\!]$ of types of $\mathcal{S}$ into types of $\mathcal{T}$;
- a translation $[\![\_]\!]$ of contexts of $\mathcal{S}$ into contexts of $\mathcal{T}$;

Translations are typically defined by induction on the syntax of their argument.

Syntactic models

Expected properties:

- Computational soundness: $M \equiv N$ implies $[M] \equiv [N]$
- Typing soundness: $\Gamma \vdash_{\mathcal{S}} M : A$ implies $[\![\Gamma]\!] \vdash_{\mathcal{T}} [M] : [\![A]\!]$
- Consistency preservation: $[\Pi A : \Box_i . A]$ is not inhabited.

Remark: Consistency of the source $\mathcal{S}$ follows from consistency of the target $\mathcal{T}$.

Example: independence of funext from CC_ω

Take $\mathcal{S}$ as CC_ω and $\mathcal{T}$ as $\mathsf{CC}_\omega + \mathbb{B}$.

$$
\begin{array}{lcl}
[\square_i]_f & := & \square_i \\
[x]_f & := & x \\
[\lambda x : A.\, M]_f & := & (\lambda x : [\![A]\!]_f.\, [M]_f, \mathtt{true}) \\
[M\ N]_f & := & \pi_1\ [M]_f\ [N]_f \\
[\Pi x : A.\, B]_f & := & (\Pi x : [\![A]\!]_f.\, [\![B]\!]_f) \times \mathbb{B} \\
[\![A]\!]_f & := & [A]_f
\end{array}
$$

[The next 700 syntactical models of type theory. S. Boulier, P.-M. Pédrot,N. Tabareau. Procs. of CPP'17]

Example: independence of funext from CC_ω

Now define:

$$\texttt{funext} := \Pi(A : \square_i)(B : \square_i)(f\ g : A \to B).(\Pi x : A.(f\ x =_B g\ x) \to f =_{A \to B} g$$

Theorem

There exists a closed proof $\vdash_{\mathcal{T}} [\![\texttt{funext} \to \bot]\!]_f$

Example: independence of funext from CC_ω

Now define:

$$\texttt{funext} := \Pi(A : \square_i)(B : \square_i)(f\ g : A \to B).(\Pi x : A.(f\ x =_B g\ x) \to f =_{A \to B} g$$

Theorem

There exists a closed proof $\vdash_{\mathcal{T}} [\![\texttt{funext} \to \bot]\!]_f$

Proof. Define $f := (\lambda x : \mathbb{B}.x, \mathit{true})$ and $g := (\lambda x : \mathbb{B}.x, \mathit{false})$.

We have $f, g : [\![\mathbb{B} \to \mathbb{B}]\!]_f$ and $(f =_{\mathbb{B} \to \mathbb{B}} g) \to \bot$.

But for any $x : \mathbb{B}$, $[f\ x =_{\mathbb{B}} g\ x]_f$ is $x =_{\mathbb{B}} x$.

From functions to dialogue trees

Theorem

For any $\vdash_{\mathcal{S}} f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$, *there exist a proof* $\vdash_{CIC} p : \mathscr{C}\ f$.

Proof. Construct a model of $\mathcal{S}$ in CIC, for which which every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is eloquent.

From functions to dialogue trees

Theorem

For any $\vdash_{\mathcal{S}} f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$*, there exist a proof* $\vdash_{CIC} p : \mathscr{C}\ f$.

Proof. Construct a model of $\mathcal{S}$ in CIC, for which which every function $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is eloquent.

The construction comes in three stages.

First model: branching translation

Remember that we fixed two parameters $\mathbb{I} : \square_0$ and $O : \mathbb{I} \to \square_0$.

Definition

For any type $\vdash A : \square$, a *pythia* is term:

$$\beta_A : \Pi i : \mathbb{I}.(O\ i \to A) \to A$$

Remember that we fixed two parameters $\mathbb{I} : \square_0$ and $O : \mathbb{I} \to \square_0$.

Definition

For any type $\vdash A : \square$, a *pythia* is term:

$$\beta_A : \Pi i : \mathbb{I}.(O\ i \to A) \to A$$

This first model equips every type $\vdash_{\mathcal{S}} A : \square$ in the source $\mathcal{S}$ with a pythia.

First model: branching translation

For every type $\vdash_{\mathcal{S}} A : \square$, define:

$$[A]_b := (\llbracket A \rrbracket_b : \square, \beta_A) \quad \text{with } \beta_A \text{ a pythia}$$

First model: branching translation, for $\mathbb{N}$

Define $[\![\mathbb{N}]\!]_b$ as $\mathbb{N}_b$ with:

$\texttt{Inductive}\ \mathbb{N}_b : \square :=$

$\mathsf{O}_b : \mathbb{N}_b \mid \mathsf{S}_b : \mathbb{N}_b \to \mathbb{N}_b \mid \beta_{\mathbb{N}} : \Pi(i : \mathbb{I}).\,(\mathbb{O}\ i \to \mathbb{N}_b) \to \mathbb{N}_b.$

First model: branching translation, for $\mathbb{N}$

Define $\llbracket \mathbb{N} \rrbracket_b$ as $\mathbb{N}_b$ with:

$\texttt{Inductive}\ \mathbb{N}_b : \square :=$

$\mathsf{O}_b : \mathbb{N}_b \ \mid\ \mathsf{S}_b : \mathbb{N}_b \to \mathbb{N}_b \ \mid\ \beta_{\mathbb{N}} : \Pi(i : \mathbb{I}).\,(\mathbb{O}\ i \to \mathbb{N}_b) \to \mathbb{N}_b.$

The non-dependent eliminator is:

$$
\begin{array}{lcl}
[\mathbb{N}_{\mathsf{cse}}]_b & : & \Pi P : \llbracket \square \rrbracket_b .\ \llbracket P \rrbracket_b \to (\mathbb{N}_b \to \llbracket P \rrbracket_b \to \llbracket P \rrbracket_b) \to \mathbb{N}_b \to \llbracket P \\
[\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ \mathsf{O}_b & := & p_{\mathsf{O}} \\
[\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ (\mathsf{S}_b\ n) & := & p_{\mathsf{S}}\ n\ ([\mathbb{N}_{\mathsf{cse}}]_b\ \ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ n)
\end{array}
$$

First model: branching translation, for $\mathbb{N}$

Define $[\![\mathbb{N}]\!]_b$ as $\mathbb{N}_b$ with:

$\texttt{Inductive}\ \mathbb{N}_b : \square :=$

$\mathsf{O}_b : \mathbb{N}_b \mid \mathsf{S}_b : \mathbb{N}_b \to \mathbb{N}_b \mid \beta_{\mathbb{N}} : \Pi(i : \mathbb{I}).\,(\mathbb{O}\ i \to \mathbb{N}_b) \to \mathbb{N}_b.$

The non-dependent eliminator is:

$$
\begin{array}{lcl}
[\mathbb{N}_{\mathsf{cse}}]_b & : & \Pi P : [\![\square]\!]_b.\ [\![P]\!]_b \to (\mathbb{N}_b \to [\![P]\!]_b \to [\![P]\!]_b) \to \mathbb{N}_b \to [\![P
\end{array}
$$

$$
\begin{array}{lcl}
[\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ \mathsf{O}_b & := & p_{\mathsf{O}} \\
[\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ (\mathsf{S}_b\ n) & := & p_{\mathsf{S}}\ n\ ([\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ n) \\
[\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ (\beta_{\mathbb{N}}\ i\ k) & := & \beta_P\ i\ (\lambda(o : \mathbb{O}\ i).\,[\mathbb{N}_{\mathsf{cse}}]_b\ P\ p_{\mathsf{O}}\ p_{\mathsf{S}}\ (k\ o))
\end{array}
$$

First model: branching translation from BTT to CIC

Agenda of the dependent eliminator:

- For $P : \mathbb{N}_b \to [\![\square]\!]_b$;
- From p_O and p_S;
- Produce a term of type $(P\ (\beta_{\mathbb{N}}\ i\ k)).\pi_1$

First model: branching translation from BTT to CIC

Agenda of the dependent eliminator:

- For $P : \mathbb{N}_b \to [\![\square]\!]_b$;
- From p_O and p_S;
- Produce a term of type $(P\ (\beta_{\mathbb{N}}\ i\ k)).\pi_1$

But this is impossible.

[An Effectful Way to Eliminate Addiction to Dependence, P.-M. Pédrot, N, Tabareau. Procs. of LICS 2017.]

First model: branching translation from BTT to CIC

Agenda of the dependent eliminator:

- For $P : \mathbb{N}_b \rightarrow [\![\square]\!]_b$;
- From p_O and p_S;
- Produce a term of type $(P\ (\beta_{\mathbb{N}}\ i\ k)).\pi_1$

But this is impossible.

Way out: restrict elimination in the source, and take $\mathcal{S} := \mathrm{BTT}$.

[An Effectful Way to Eliminate Addiction to Dependence, P.-M. Pédrot, N, Tabareau. Procs. of LICS 2017.]

Restricted elimination in BTT

$$\frac{\Gamma \vdash P : \square \qquad \Gamma \vdash t_{\mathsf{O}} : P \qquad \Gamma \vdash t_{\mathsf{S}} : \mathbb{N} \to P \to P}{\Gamma \vdash \mathbb{N}_{\mathsf{cse}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}} : \mathbb{N} \to P}$$

$$\frac{\Gamma \vdash P : \mathbb{N} \to \square \qquad \Gamma \vdash t_{\mathsf{O}} : \mathbb{N}_{\mathsf{str}}\ \mathsf{O}\ P \qquad \Gamma \vdash t_{\mathsf{S}} : \Pi(n : \mathbb{N}).\, \mathbb{N}_{\mathsf{str}}\ n\ P \to \mathbb{N}_{\mathsf{str}}\ (\mathsf{S}\ n)\ P}{\Gamma \vdash \mathbb{N}_{\mathsf{rec}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}} : \Pi(n : \mathbb{N}).\, \mathbb{N}_{\mathsf{str}}\ n\ P}$$

Restricted elimination in BTT

$$\frac{\Gamma \vdash P : \square \qquad \Gamma \vdash t_{\mathsf{O}} : P \qquad \Gamma \vdash t_{\mathsf{S}} : \mathbb{N} \to P \to P}{\Gamma \vdash \mathbb{N}_{\mathsf{cse}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}} : \mathbb{N} \to P}$$

$$\frac{\Gamma \vdash P : \mathbb{N} \to \square \qquad \Gamma \vdash t_{\mathsf{O}} : \mathbb{N}_{\mathsf{str}}\ \mathsf{O}\ P \qquad \Gamma \vdash t_{\mathsf{S}} : \Pi(n : \mathbb{N}).\, \mathbb{N}_{\mathsf{str}}\ n\ P \to \mathbb{N}_{\mathsf{str}}\ (\mathsf{S}\ n)\ P}{\Gamma \vdash \mathbb{N}_{\mathsf{rec}}\ P\ t_{\mathsf{O}}\ t_{\mathsf{S}} : \Pi(n : \mathbb{N}).\, \mathbb{N}_{\mathsf{str}}\ n\ P}$$

where

$$\begin{aligned}
&\mathbb{N}_{\mathsf{str}}\ (n : \mathbb{N})\ (P : \mathbb{N} \to \square) : \square := \\
&\quad \mathbb{N}_{\mathsf{cse}}\ ((\mathbb{N} \to \square) \to \square)\ (\lambda(Q : \mathbb{N} \to \square).\, Q\ \mathsf{O}) \\
&\qquad (\lambda(m : \mathbb{N})\,(_ : (\mathbb{N} \to \square) \to \square)\,(Q : \mathbb{N} \to \square).\, Q\ (\mathsf{S}\ m))\ n\ P.
\end{aligned}$$

First model: branching translation from BTT to CIC

Theorem

The branching translation $[\_]_b$ defines a syntactic model from BTT to CIC.

Dialogue in the branching model

Dialogue in the branching model

$$\texttt{Inductive}\ \mathbb{N}_b : \square :=$$

$$\mathsf{O}_b : \mathbb{N}_b \ \mid\ \mathsf{S}_b : \mathbb{N}_b \to \mathbb{N}_b \ \mid\ \beta_{\mathbb{N}} : \Pi(i : \mathbb{I}).\,(\mathbb{O}\ i \to \mathbb{N}_b) \to \mathbb{N}_b.$$

$$\begin{array}{lcl} \partial^{\mathbb{N}} & : & \mathbb{Q} \to \mathbb{N}_b \to \mathbb{N} \\ \partial^{\mathbb{N}}\ \alpha\ \mathsf{O}_b & := & \mathsf{O} \\ \partial^{\mathbb{N}}\ \alpha\ (\mathsf{S}_b\ n_b) & := & \mathsf{S}\ (\partial^{\mathbb{N}}\ \alpha\ n_b) \\ \partial^{\mathbb{N}}\ \alpha\ (\beta_{\mathbb{N}}\ i\ k) & := & \partial^{\mathbb{N}}\ \alpha\ (k\ (\alpha\ i)). \end{array}$$

Next step

Relate:

$$\alpha : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \vdash \boldsymbol{n} : \mathbb{N}$$

with:

$$\vdash \boldsymbol{n}_b : \mathbb{N}_b$$

Second model: axiom translation from BTT to CIC

This second model forces the presence a reserved variable $\alpha : \mathbb{Q}$ in the context.

$$
\begin{array}{lcl}
[\square_i]_a & := & \square_i \\
[\![A]\!]_a & := & [A]_a \\
[\![\cdot]\!]_a & := & \alpha : \mathbb{Q} \\
[\![\Gamma,\ x : A]\!]_a & := & [\![\Gamma]\!]_a, x_a : [\![A]\!]_a \\
\ldots & := & \ldots
\end{array}
$$

Second model: axiom translation from BTT to CIC

This second model forces the presence a reserved variable $\alpha : \mathbb{Q}$ in the context.

$$
\begin{array}{lcl}
[\square_i]_a & := & \square_i \\
[\![A]\!]_a & := & [A]_a \\
[\![\cdot]\!]_a & := & \alpha : \mathbb{Q} \\
[\![\Gamma,\ x : A]\!]_a & := & [\![\Gamma]\!]_a, x_a : [\![A]\!]_a \\
\ldots & := & \ldots
\end{array}
$$

Theorem

The branching translation $[\_]_a$ defines a syntactic model from BTT to CIC.

Note: $[\_]_a$ **should really be denoted** $[\_]_a^\alpha$, **for a given name** α.

Third model: algebraic parametricity translation

Given $\alpha : \mathbb{N} \to \mathbb{N}, \Gamma \vdash t : A$, relate:

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

with:

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

using an internal logical relation:

$$[\![\Gamma]\!]_\varepsilon \vdash [t]_\varepsilon : [\![A]\!]_\varepsilon \ [t]_a \ [t]_b$$

Third model: algebraic parametricity translation

Given $\alpha : \mathbb{N} \to \mathbb{N}, \Gamma \vdash t : A$, relate:

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

with:

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

using an internal logical relation:

$$[\![\Gamma]\!]_\varepsilon \vdash [t]_\varepsilon : [\![A]\!]_\varepsilon\ [t]_a\ [t]_b$$

In fact, the parametricity predicate $[\![A]\!]_\varepsilon$ has to be algebraic as well.

Third model: algebraic parametricity translation

For every type $\vdash_{\mathcal{S}} A : \Box$:

$$[A]_\varepsilon := (\llbracket A \rrbracket_\varepsilon, \beta^\varepsilon_A)$$

with:

- $\llbracket A \rrbracket_\varepsilon : \llbracket A \rrbracket_a \to \llbracket A \rrbracket_b \to \Box$
- $\beta^\varepsilon_A : \Pi(x_a : \llbracket A \rrbracket_a)\,(i : \mathbb{I})\,(k : \mathbb{O}\ i \to \llbracket A \rrbracket_b).\ \llbracket A \rrbracket_\varepsilon\ x_a\ (k\ (\alpha\ i)) \to \llbracket A \rrbracket_\varepsilon\ x_a\ (\beta_A\ i\ k)$

Third model: algebraic parametricity translation

$\texttt{Inductive}\ \mathbb{N}_\varepsilon\ (\alpha : \mathbb{Q}) : \mathbb{N} \to \mathbb{N}_b \to \square :=$

$\mid \mathsf{O}_\varepsilon : \mathbb{N}_\varepsilon\ \alpha\ \mathsf{O}\ \mathsf{O}_b$

$\mid \mathsf{S}_\varepsilon : \Pi(n_a : \mathbb{N})\,(n_b : \mathbb{N}_b)\,(n_\varepsilon : \mathbb{N}_\varepsilon\ \alpha\ n_a\ n_b).\, \mathbb{N}_\varepsilon\ \alpha\ (\mathsf{S}\ n_a)\ (\mathsf{S}_b\ n_b)$

Third model: algebraic parametricity translation

$\texttt{Inductive}\ \mathbb{N}_\varepsilon\ (\alpha : \mathbb{Q}) : \mathbb{N} \to \mathbb{N}_b \to \square :=$

$\mid \mathsf{O}_\varepsilon : \mathbb{N}_\varepsilon\ \alpha\ \mathsf{O}\ \mathsf{O}_b$

$\mid \mathsf{S}_\varepsilon : \Pi (n_a : \mathbb{N})\,(n_b : \mathbb{N}_b)\,(n_\varepsilon : \mathbb{N}_\varepsilon\ \alpha\ n_a\ n_b).\, \mathbb{N}_\varepsilon\ \alpha\ (\mathsf{S}\ n_a)\ (\mathsf{S}_b\ n_b)$

$\mid \beta^\varepsilon_\mathbb{N} : \Pi (n_a : \mathbb{N})\,(i : \mathbb{I})\,(k : \mathbb{O}\ i \to \mathbb{N}_b)\,(n_\varepsilon : \mathbb{N}_\varepsilon\ \alpha\ n_a\ (k\ (\alpha\ i))).\, \mathbb{N}_\varepsilon\ \alpha\ n_a\ (\beta_\mathbb{N}\ i\ k)$

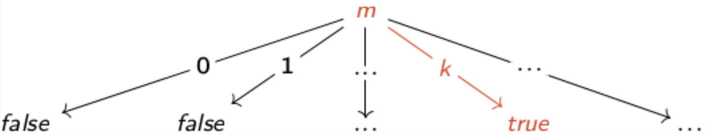
Third model: algebraic parametricity translation

$$\beta_A^\varepsilon : \Pi(x_a : [\![A]\!]_a)\,(i : \mathbb{I})\,(k : \mathbb{O}\ i \to [\![A]\!]_b).\ [\![A]\!]_\varepsilon\ x_a\ (k\ (\alpha\ i)) \to [\![A]\!]_\varepsilon\ x_a\ (\beta_A\ i\ k)$$

Third model: algebraic parametricity translation

$\beta_A^\varepsilon : \Pi(x_a : [\![A]\!]_a)\,(i : \mathbb{I})\,(k : \mathbb{O}\ i \to [\![A]\!]_b).\ [\![A]\!]_\varepsilon\ x_a\ (k\ (\alpha\ i)) \to [\![A]\!]_\varepsilon\ x_a\ (\beta_A\ i\ k)$

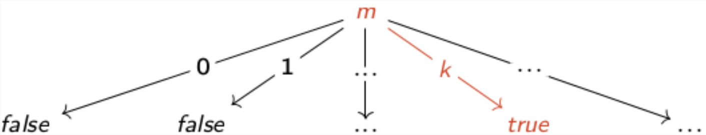
For example, consider the following branching boolean b, for $\mathbb{O} := \_ : \mathbb{N} \mapsto \mathbb{N}$:



Third model: algebraic parametricity translation

$$\beta^{\varepsilon}_{A} : \Pi(x_a : [\![A]\!]_a)\,(i : \mathbb{I})\,(k : \mathbb{O}\ i \to [\![A]\!]_b).\ [\![A]\!]_{\varepsilon}\ x_a\ (k\ (\alpha\ i)) \to [\![A]\!]_{\varepsilon}\ x_a\ (\beta_A\ i\ k)$$

For example, consider the following branching boolean b, for $\mathbb{O} := \_ : \mathbb{N} \mapsto \mathbb{N}$:



As soon as:

$$\alpha\ m = k$$

We can prove that:

$$[\![\mathbb{B}]\!]_{\varepsilon}\ \text{true}\ b$$

Third model: algebraic parametricity translation

Theorem

The branching translation $[\_]_\varepsilon$ *defines a syntactic model from BTT to CIC.*

Note: $[\_]_\varepsilon$ **is parameterized by the name** α **and by** $\mathbb{I}$ **and** $\mathbb{O}$.

Continuity

Theorem

If $\vdash_{BTT} f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ *then* $\vdash_{CIC} \mathscr{C}\ \lambda\alpha.([f]_a^\alpha\ \alpha)$

Key properties

- Unicity:

$$\vdash_{CIC} \_ : \Pi(\alpha : \mathbb{Q})\,\langle n : \mathbb{N}\rangle.\, n_a = \partial^{\mathbb{N}}\ \alpha\ n_b$$

- Dialogue relation:

$$\vdash_{CIC} \_ : \Pi(\alpha : \mathbb{Q})\,(n_b : \mathbb{N}_b).\, \mathbb{N}_\varepsilon\ \alpha\ (\partial^{\mathbb{N}}\ \alpha\ n_b)\ n_b$$

- Generic element γ_b:

$$\vdash_{CIC} \_ : \Pi(\alpha : \mathbb{N} \to \mathbb{N})\,(n_b : \mathbb{N}_b).\, \partial^{\mathbb{N}}\ \alpha\ (\gamma_b\ n_b) = \alpha\ (\partial^{\mathbb{N}}\ \alpha\ n_b)$$

Continuity

Theorem

$$\vdash_{CIC} \_ : \Pi(\alpha : \mathbb{N} \to \mathbb{N}).\, [f]_a^\alpha\ \alpha = \partial^{\mathbb{N}}\ \alpha\ ([f]_b\ \gamma_b)$$

Proof: First construct:

$$\alpha : \mathbb{N} \to \mathbb{N} \vdash_{CIC} \gamma_\varepsilon : [\![\mathbb{N} \to \mathbb{N}]\!]_\varepsilon\ \alpha\ \gamma_b$$

Continuity

Theorem

$$\vdash_{CIC} \_ : \Pi(\alpha : \mathbb{N} \to \mathbb{N}).\, [f]_a^{\alpha}\ \alpha = \partial^{\mathbb{N}}\ \alpha\ ([f]_b\ \gamma_b)$$

Proof: First construct:

$$\alpha : \mathbb{N} \to \mathbb{N} \vdash_{CIC} \gamma_\varepsilon : [\![\mathbb{N} \to \mathbb{N}]\!]_\varepsilon\ \alpha\ \gamma_b$$

Then by soundness:

$$\begin{array}{lll} \alpha : \mathbb{N} \to \mathbb{N} & \vdash_{CIC} & [f]_a\ \alpha : \mathbb{N} \\ & \vdash_{CIC} & [f]_b\ \gamma_b : \mathbb{N}_b \\ \alpha : \mathbb{N} \to \mathbb{N} & \vdash_{CIC} & [f]_\varepsilon\ \alpha\ \gamma_b\ \gamma_\varepsilon : \mathbb{N}_\varepsilon\ \alpha\ ([f]_a\ \alpha)\ ([f]_b\ \gamma_b) \end{array}$$

And conclude using the previous properties.

Conclusion

- Transpose M. Escardó's proof to a dependently typed setting
- Formalized in Coq
- Internalization?
- Scope of the methodology?

[Gardening with the Pythia. Procs of CSL 2022]