Formalizing a set-theoretical model of CIC in Coq/IZF

Bruno Barras

INRIA Saclay

Feb. 14 & 15, 2011

Overview

Today:

- General setup: models and strong normalization
- Predicative universes

Tomorrow:

- Inductive types: nat and ordinals

  ```
  Inductive nat := O | S (_:nat).
  Inductive ord := O | LimS (_:nat->ord).
  ```

  (should generalize easily to `Inductive W A B := sup (x:A) (_:B x ->W A B).` and thus to any strictly positive inductive definition)

Motivations

Why a model of CIC ?

- Currently *no model of the full formalism of Coq*: features studied separately: Streicher, Coquand, Luo, Werner, H. Goguen
- No *strong intuition* of which axioms are consistent with CIC (Chicli-Pottier-Simpson paradox)

Why formally ?

- To be "sure"
- To make it *simpler* (for both the designer and the reader)

Which model do we want ?

- Smallest model vs
  Model with smallest number of assumptions
  (or: studying the proof theoretic strength of CIC vs supporting more axioms)

In particular, we do not limit ourselves to continuous or computable functions (countable model). We want to be able to support classical reals, powerful description axioms, extensionality and what not...

Set-theoretical model:

$$A \to B \text{ set of all set-theoretical function from } A \text{ to } B.$$

Set theory: IZF

Axiomatized Zermelo-Fraenkel without excluded-middle:

- a carrier type set with equality $=$ and membership $\in$,
- pair $\{a; b\}$,
- union $\bigcup a$,
- powerset $\mathcal{P}(a)$,
- separation $\{x \in A \mid P(x)\}$,
- replacement $\{y \mid \exists x \in A.R(x, y)\}$ (R functional relation),
- infinity
- unused: well-foundation (instead of regularity in ZF)

Library: couples, relations, functions, plump ordinals, fixpoint theorem, Grothendieck universes, ...

The Playground

$$\begin{tikzcd}
\text{MLTT} \arrow[r, "\text{(Imp)}"] & \text{CC} \arrow[rr, "\text{(Univ)}"] \arrow[dr, "\text{(Ind)}"] \arrow[ddr, "\text{(Ext)}"'] & & \text{ECC} \arrow[dr] & \\
& & \text{CC+NAT} \arrow[r] & \text{CC+W} \arrow[r] & \text{CIC} \arrow[d] \\
& & \text{CCUT} \arrow[rr] & & \text{ECIC}
\end{tikzcd}$$

Three independent features:

- Predicative universes
- Inductive types
- Extentional theory

Semantics first

Usual scheme:

- Introduce the syntax: terms and judgements
- Define the interpretation (recursion over the syntax)
- Prove soundness of the interpretation

Many systems: better avoid to start from the syntax!

- Shallow embedding
- Naturally extendable
- “Pick” the syntax (not shown in this talk)

Model construction scheme

Abstract model ($\mathcal{M}$):

- Describes the world of *ground expressions*
- Judgement: $[M : T]$ or $M \in \mathsf{El}(T)$

Dealing with free variables (de Bruijn):

- constr $\triangleq (\mathbb{N} \to \mathcal{M}) \to \mathcal{M}$
- valuations $\triangleq \mathbb{N} \to \mathcal{M}$

Judgements:

- $[\Gamma]$ = sets of valid valuations: ρ s.t. $(x : T) \in \Gamma \Rightarrow [x\rho : T\rho]$
- $[\Gamma \vdash M : T] \triangleq \forall \rho \in [\Gamma], [M\rho : T\rho]$
- Derive all necessary typing rules
  (so we have soundness: $\Gamma \vdash M : T \Rightarrow [\Gamma \vdash M : T]$)

Abstract model of Martin Löf's Type Theory

Structure:

- A setoid $(\mathcal{M}, =)$, membership $\_ \in \mathrm{El}(\_)$
- Operations: $\Lambda, @, \Pi$
  - $\Lambda(A, F)$: function F with domain A — $A : \mathcal{M} \quad F : \mathcal{M} \to \mathcal{M}$
  - $@(M, N)$: application — $M, N : \mathcal{M}$
  - $\Pi(A, B)$: set of dependent functions — $A : \mathcal{M} \quad B : \mathcal{M} \to \mathcal{M}$
- Properties:
  - Π-intro: $(\forall x \in \mathrm{El}(A).\ F(x) \in \mathrm{El}(B(x))) \ \Rightarrow\ \Lambda(A, F) \in \mathrm{El}(\Pi(A, B))$
  - Π-elim: $M \in \mathrm{El}(\Pi(A, B)) \ \wedge\ N \in \mathrm{El}(A) \ \Rightarrow\ @(M, N) \in \mathrm{El}(B(N))$
  - β-equality: $N \in \mathrm{El}(A) \ \Rightarrow\ @(\Lambda(A, F), N) = F(N)$

Straightforward implementation:

- $\mathcal{M}$ = set and El is the identity (alternative: HF)
- Π dependent product (usual encoding of functions)
- Note: Λ *uses* the domain argument

Abstract model of CC

Additional constants and properties:

- Prop: $*$
- Impredicativity: $(\forall x \in \mathrm{El}(A).\ B(x) \in \mathrm{El}(*)) \Rightarrow \Pi(A,B) \in \mathrm{El}(*)$

Note: topsort Kind is the proper class $\mathcal{M}$:

$$[M : T] \triangleq M \neq \mathrm{Kind} \wedge (T = \mathrm{Kind} \vee M \in \mathrm{El}(T))$$

Implementation:

- Aczel's encoding $$\begin{aligned} \Lambda(A,F) &\triangleq \{(x,y) \mid x \in A \wedge y \in F(x)\} \\ @(M,N) &\triangleq \{y \mid (N,y) \in M\} \end{aligned}$$
- But this is incompatible with Streicher's method because functions do not carry their domain.

Model in details

[Remember: constr $= (\mathbb{N} \to \mathcal{M}) \to \mathcal{M}$]

$$\begin{aligned} [n] &\triangleq \rho \mapsto \rho(n) \\ [\mathrm{Prop}] &\triangleq \_ \mapsto * \end{aligned} \qquad \begin{aligned} [\lambda x\colon T.\, M] &\triangleq \rho \mapsto \Lambda(T\rho,\, v \mapsto M(v :: \rho)) \\ [M\ N] &\triangleq \rho \mapsto @(M\rho,\, N\rho) \\ [\Pi x\colon A.\, B] &\triangleq \rho \mapsto \Pi(A\rho,\, v \mapsto B(v :: \rho)) \end{aligned}$$

$$[M\{0 \backslash N\}] \triangleq \rho \mapsto M(N\rho :: \rho)$$

Judgements:

- Valid valuations: $[\Gamma] \triangleq \{\rho \mid \forall (x : T) \in \Gamma.\, [\rho(x) : T\rho]\}$
- Typing: $[\Gamma \vdash M : T] \triangleq \forall \rho \in [\Gamma].\, [M\rho : T\rho]$
- Equality: $[\Gamma \vdash M = N] \triangleq \forall \rho \in [\Gamma].\, M\rho = N\rho$

Soundness

Deriving rules: (soundness: $\Gamma \vdash M : T \Rightarrow [\Gamma \vdash M : T]$)

$$\frac{}{[\Gamma \vdash n : \Gamma(n)]} \qquad \frac{}{[\Gamma \vdash \mathrm{Prop} : \mathrm{Kind}]}$$

$$\frac{[\Gamma \vdash M : \Pi x{:}A.\,B] \qquad [\Gamma \vdash N : A] \qquad A \neq \mathrm{Kind}}{[\Gamma \vdash M\ N\ :\ B\{x\backslash N\}]}$$

$$\frac{[\Gamma\,(x{:}T) \vdash M : U] \qquad T, U \neq \mathrm{Kind}}{[\Gamma \vdash \lambda x{:}T.\,M\ :\ \Pi x{:}T.\,U]}$$

$$\frac{[\Gamma \vdash M : T] \qquad [\Gamma \vdash T = T']}{[\Gamma \vdash M : T']} \qquad \frac{[\Gamma;(x{:}T) \vdash U : s_2] \quad T \neq \mathrm{Kind}}{[\Gamma \vdash \Pi x{:}T.\,U\ :\ s_2]}$$

Consistency

- Valid contexts: $[\Gamma] \neq \emptyset$
  - $\Gamma = [\,]$
  - extensionality
  - $\forall P.\ \neg\neg P \to P$ if we assume EM at the meta-level (IZF$\to$ZF)
- $[\Pi P\!:\!\mathrm{Prop}.\,P] = \emptyset$
  From soundness, absurd cannot be derived in a valid context:

$$\nexists M.\,\Gamma \vdash M : \Pi P\!:\!\mathrm{Prop}.\,P$$

Extendability

This model can be extended with any set of IZF

- Semantical judgement allows *Kind* variables
- Not allowed by derivations

Examples of valid contexts:

- (nat:Kind) (O:nat) (S:nat$\rightarrow$nat) ...
- Classical real numbers (in ZF)
- ...

Strong Normalization

Strong normalization models

SN is a syntactic result (proved semantically)

Strong normalization as a realizability model construction.

Terms and types use the same syntax.

Consistency model vs SN model

Recursive realizability:

- Attach a *satured set* to every type (Sat): set of realizers
- Attach a *pure λ-term* to every term (Tm): realizer
- Extra invariant: realizer belongs to the set of realizers of its type:

$$[M : T] \triangleq \mathrm{Val}(M) \in \mathrm{El}(T) \ \wedge \ \mathrm{Tm}(M) \in \mathrm{Sat}(T)$$

All types must be inhabited to ensure SN (reduction under binders).

Abstract SN model

Extra operations:

- $\mathsf{Sat} : \mathcal{M} \to \mathsf{SAT}$
- $\bullet : \mathcal{M}$

Extra properties:

- $\mathsf{Sat}(\Pi x{:}A.\,B) = \mathsf{Sat}(A) \to \bigcap_{x \in \mathsf{El}(A)} \mathsf{Sat}(B(x))$
- $\mathsf{Sat}(\mathsf{Prop}) = \mathcal{SN}$
- $\bullet \in \mathsf{El}(\Pi P{:}\mathsf{Prop}.\,P)$

Does not support strong elimination, but works for CC and ECC.

Supporting strong elimination

Previous definition does not support strong elimination:

- $\mathsf{Sat}(\Pi n\!:\!\mathbb{N}.\,P(n)) = \{f \mid \forall n\!\in\!\mathbb{N}.\,\forall u\!\in\!\mathsf{Sat}(\mathbb{N}).\,f\ \ u \in \mathsf{Sat}(P(n))\}$
- So, $f\ \ 0$ should be in $\mathsf{Sat}(P(1))$!
- Need coherence between n and u: Λ-sets

My take on Λ-sets

No need to define Λ-sets with specific properties:

- Replace Sat by a realizability relation $\Vdash_T$: $\mathrm{El}(T) \to \mathrm{SAT}$
  Notation: $t \Vdash_T v$ stands for $v \in \mathrm{El}(T) \ \wedge t \in \Vdash_T (v)$.
- $t \Vdash_{\Pi(A,B)} f \triangleq \forall u\, v.\, u \Vdash_A v \Rightarrow t\ u \Vdash_{B(v)} @(f, v)$ as usual.

Abstract SN model: new version

- Operations: El and $\Vdash$
- Impredicativity:
  $T \in \mathcal{SN} \;\wedge\; (\forall u\, v.\, u \Vdash_A v \Rightarrow U\; u \Vdash_* @(B, v)) \;\Rightarrow\; \mathsf{K}\; T\; U \Vdash_* \Pi(A, B)$

Implementation:

- $[*] \triangleq \langle \mathrm{El} = \{\langle \mathrm{El} = \{\emptyset\}; \mathrm{Sat} = \lambda\_.\, S\rangle \mid S \in \mathrm{SAT}\}; \mathrm{Sat} = \lambda\_.\, \mathcal{SN}\rangle$
- $[\Pi x : A.\, B] = \left\langle \mathrm{El} = \Pi(\mathrm{El}(A), v \mapsto \mathrm{El}(B(v))); \mathrm{Sat} = \Vdash_{\Pi(A,B)} \right\rangle$

SN Model in details

$$\begin{array}{rcl}
\mathsf{Tm}(n) & \triangleq & \sigma \mapsto \sigma(n) \\
\mathsf{Tm}(\mathsf{Prop}) & \triangleq & \_ \mapsto \mathsf{K} \\
\mathsf{Tm}(\lambda x\!:\!T.\,M) & \triangleq & \sigma \mapsto \mathsf{K}\ (\lambda x.\,M\sigma)\ T\sigma \\
\mathsf{Tm}(M\ N) & \triangleq & \sigma \mapsto M\sigma\ N\sigma \\
\mathsf{Tm}(\Pi x\!:\!A.\,B) & \triangleq & \sigma \mapsto \mathsf{K}\ (\lambda x.\,B\sigma)\ A\sigma
\end{array}$$

Formally: Tm and Val defined simultaneously
(constr $= (\mathbb{N} \to \mathcal{M}) \to \mathcal{M} \ \times\ (\mathbb{N} \to \Lambda) \to \Lambda$ with substitutivity
requirement: $M(\sigma' o \sigma) = M\sigma\{\sigma'\}$
(M does not introduce free variables)

Judgements

$$[\Gamma] \triangleq \{(\rho, \sigma) \mid \forall (x : T) \in \Gamma.\, \sigma(x) \Vdash_{T\rho} \rho(x)\}$$

$$[\Gamma \vdash M : T] \triangleq \forall (\rho, \sigma) \in [\Gamma].\, M\sigma \Vdash_{T\rho} M\rho$$
$$[\Gamma \vdash M = N] \triangleq \forall (\rho, \sigma) \in [\Gamma].\, M\rho = N\rho$$

Equality is based only on the denotation (not the realization). This makes extensionality principles admissible. Could we also require $M\sigma$ convertible to $N\sigma$?

Strong Normalization Theorem

Simulation of reductions:

- $((\lambda x : T.\, M)\ N)\sigma \rightarrow^{+} (M\{x\backslash N\})\sigma$

$[\Gamma \vdash M : T] \Rightarrow M(\_ \mapsto x) \in \mathcal{SN}$

Consistency out of the SN model

Assume $[\vdash M : \Pi P\!:\!\mathrm{Prop}.\, P]$.

- By soundness we have $M(\_ \mapsto \lambda x.x) \Vdash_{(\Pi P:\mathrm{Prop}.P)} M(\_ \mapsto \emptyset)$.
- So $M(\_ \mapsto \lambda x.x)$ is closed (by substitutivity),
- but $M(\_ \mapsto \lambda x.x) \in \bigcap_{S \in \mathrm{SAT}} S$ which contains no closed term.

Conclusion: there is no proof of absurdity.

Extendability

Which extensions are supported by this model ?

- Adding an IZF set as a type without reduction rules: new constants are interpreted (Tm) by neutral terms; new types assign $\mathcal{SN}$ to every value ($\Vdash$).
- Adding an IZF set as a type with reductions rules that can be simulated by β-reduction (sequential computations): interpret new constants and types accordingly. *Inductive types of CIC fall into this category.*
- If the reduction rules cannot be simulated by β, add new constants to the λ-calculus with appropriate reductions. The notion of saturated set has to be adapted.

Predicative Universes

Extended Calculus of Constructions

New rules:

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathsf{Type}_i : \mathsf{Type}_{i+1}} \qquad \frac{\Gamma \vdash T : \mathsf{Type}_i \quad \Gamma; (x : T) \vdash U : \mathsf{Type}_i}{\Gamma \vdash \Pi x : T.U : \mathsf{Type}_i}$$

$$\frac{\Gamma \vdash M : T \quad \Gamma \vdash T \leq T' \quad \Gamma \vdash T' : s}{\Gamma \vdash M : T'}$$

$$\frac{}{\Gamma \vdash \mathsf{Type}_i \leq \mathsf{Type}_{i+1}} \qquad \frac{\Gamma; (x : T) \vdash U \leq U'}{\Gamma \vdash \Pi x : T.U \leq \Pi x : T.U'}$$

Luo showed strong normalization of ECC in ZF but this proof does not extend to inductive types with strong elimination.

Abstract model of ECC

Consistency:

- A sequence of sets $(\Box_i)_{i\in\mathbb{N}}$
- Property: $\forall i.\, \Box_i \in \mathsf{El}(\Box_{i+1}) \;\wedge\; \mathsf{El}(\Box_i) \subseteq \mathsf{El}(\Box_{i+1})$
- Predicativity:
  $A \in \mathsf{El}(\Box_i)) \;\wedge\; (\forall x \in \mathsf{El}(A).\; B(x) \in \mathsf{El}(\Box_i)) \;\Rightarrow\; \Pi(A, B) \in \mathsf{El}(\Box_i)$

SN:

- $\forall i, \bullet_i \in \Pi(\Box_i, X \mapsto X)$

Beware: extensionality and strong normalization!
$h : \mathsf{nat} = (\mathsf{nat} \to \mathsf{nat}) \vdash (\lambda x.\, x\ x)\ (\lambda x.\, x\ x) : \mathsf{nat}$

Grothendieck Universes

Grothendieck universes: sets that are model of the theory

- Any set construction based on elements of the universe is still in the universe
- Equivalent to assuming the existence of an inaccessible cardinal:
  If μ inaccessible cardinal, then V_μ is a Grothendieck universe;
  conversely the ordinals of a Grothendieck universe form an inaccessible cardinal

Obviously models predicative products:

- $\vdash A : s \quad \wedge \quad x{:}A \vdash B : s \quad \Rightarrow \quad \vdash \Pi x{:}A.B \ : \ s$

No interference between universes and any set theoretical construction.

Abstract model of ECC implemented

Consistency:

- A sequence of Grothendieck universes $(\Box_i)_{i\in\mathbb{N}}$
- Property $\forall i.\, \Box_i \in \Box_{i+1}$ trivial.
- Cumulativity follows from transitivity of Grothendieck universes.

Conclusion (for today)

- Method: keep it extendable (modularity universes/construction within a universe)
- Tried to remain as extensional as possible

Tomorrow:

- inductive types (nat mostly),
- separation match/fix and termination by type-checking.

Inductive Types

Overview

Model construction

- Induction: constructors and pattern-matching
- Type with stages
- Recursive functions
- Convergence

Judgements

Strong normalization models

- Recursor

Recursive types

Given a *monotonic* type operator F, build $\mu X.\, F(X)$ the least fixpoint of F (when it exists).

Existence of a fixpoint:

- monotonicity not sufficient: $\mathcal{P}$ is monotone but has no fixpoint
- strict positivity condition sufficient

Typical examples:

- nat: $F_{\mathsf{nat}}(X) \triangleq 1 + X$ (isomorphic to $\{0\} \times \mathsf{unit} \cup \{1\} \times X$)
- ord: $F_{\mathsf{ord}}(X) \triangleq 1 + (\mathsf{nat} \to X)$
- W(A, B): $F_W(X) = \Sigma x : A.\, (B(x) \to W(A, B))$

Inductive principles: constructors

(For illustration purposes, we assume $F = F_{\mathrm{nat}}$)

Constructors ($X \rightsquigarrow F(X)$):

- $0 \triangleq (0, \mathrm{tt}) \in F(X)$
- $S \triangleq n \mapsto (1, n) \in X \to F(X)$
- Discrimination: $0 \neq S(n)$
- Injection: $S(m) = S(n) \Rightarrow m = n$
- Intuitionist's corner: stability $\bigcap_{i \in I} F(X_i) = F(\bigcap_{i \in I} X_i)$

Inductive principles: pattern-matching

Pattern-matching $(\forall P.\,(X \to P) \rightsquigarrow (F(X) \to P))$:

- $n \in F(X) \Rightarrow n = 0 \;\vee\; \exists a \in X.\, n = S(a)$
- $\mathrm{Natcase}(n, f, g) \triangleq$
  $\{y \in \{f\} \mid n = 0\} \cup \{y \in \{g(\mathrm{snd}(n))\} \mid \exists k.\, n = S(k)\}$
- Reduction: $\mathrm{Natcase}(0, f, g) = f \quad \mathrm{Natcase}(S(k), f, g) = g(k)$
- Typing: $\mathrm{Natcase}(n, f, g) \in P(n)$
  whenever $n \in F(X)$, $f \in P(0)$ and $g \in \Pi(X, v \mapsto P(S(v)))$

Type with stages

Let $I^\alpha \triangleq F^\alpha(\emptyset)$ (for any ordinal α).

Convergence of I^α to the least fixpoint of F investigated later on.

Properties of I^α:

- monotonicity: $\alpha \leq \beta \Rightarrow I^\alpha \subseteq I^\beta$ (in particular $I^\alpha \subseteq I^{\alpha^+}$)
- $I^{\alpha^+} = F(I^\alpha)$ (equi inductive type)
- I^α is stable: $\bigcap_{\alpha \in J} I^\alpha = I^{\bigcap_{\alpha\in J} \alpha_i}$

  So $\{\alpha \mid a \in I^\alpha\}$ has a least element (when not empty) for all a.

Constructors and patten-matching again

Typing:

- $\forall\alpha.\ 0 \in I^{\alpha^+}$
- $\forall\alpha.\ S \in I^{\alpha} \to I^{\alpha^+}$
- $\mathrm{Natcase}(n, f, g) \in P(n)$
  if $n \in I^{\alpha^+}$, $f \in P(0)$ and $g \in \Pi(I^{\alpha}, k \mapsto P(S(k)))$

Recursive functions: requirements

The goal is to build recursively a function of domain I^{α} (and codomain U_{α}), given a process G that transforms a function of domain I^{β} to a function of domain I^{β^+} (with $\beta \leq \alpha$).

Summary:

- Given $G_{\beta} \in (I^{\beta} \to U_{\beta}) \to (I^{\beta^+} \to U_{\beta^+})$
- We shall build $\mathrm{Fix}(G, \alpha) \in I^{\alpha} \to U_{\alpha}$
- Satisfying the fixpoint equation $\mathrm{Fix}(G, \alpha) = G_{\alpha}(\mathrm{Fix}(G, \alpha))$

Issues:

- G should not use its ordinal argument computationally (otherwise $G_{\beta}(f)$ and f might not agree on domain I^{β})
- The fixpoint equation is ill-typed!

Recursive functions: the construction

Given an ordinal α, $(U_\beta)_{\beta<\alpha}$ and $(G_\beta)_{\beta<\alpha}$ s.t.:

- G typing: $G_\beta \in (\Pi(I^\beta, U_\beta) \to (\Pi(I^{\beta^+}, U_{\beta^+}))$
- U monotone: $\forall \gamma \le \beta < \alpha . \forall x \in I^\gamma . \, U_\gamma(x) \subseteq U_\beta(x)$
- G monotone and "stage irrelevant": $(f \equiv_A g \triangleq \forall x \in A . f(x) = g(x))$
  $\forall \gamma \le \beta < \alpha . \; \forall f \in \Pi(I^\gamma, U_\gamma) . \; \forall g \in \Pi(I^\beta, U_\beta) .$
  $f \equiv_{I^\gamma} g \Rightarrow G_\gamma(f) \equiv_{I^{\gamma^+}} G_\beta(g)$

Then we define Fix by transfinite induction:

$$\mathrm{Fix}_\beta(G) \triangleq \bigcup_{\gamma<\beta} G_\gamma(\mathrm{Fix}_\gamma(G))$$

Properties (for $\beta < \alpha$):

- Typing: $\mathrm{Fix}_\beta(G) \in \Pi(I^\beta, U_\beta)$
- Fixpoint equation: $\mathrm{Fix}_\beta(G) = \Lambda(I^\beta, G_\beta(\mathrm{Fix}_\beta(G)))$

Recursive functions: comments

- Requirement on U is stronger than needed:
  Abel defined a better continuity criterion
- Set-theoretical artifacts:
  - η-expansion in the fixpoint equation
  - G needs the ordinal argument to know the exact domain of its argument (the recursive function at previous stage), but its result do not depend on it (“stage irrelevance”)

Convergence

Goal find an ordinal λ such that $I^\lambda = F(I^\lambda)$ (closure ordinal of I).

Case of first-order datatypes (e.g. nat):

- F is (ω-)continuous: $F(\bigcup_{i\in\omega} X_i) = \bigcup_{i\in\omega} F(X_i)$
- closure ordinal $\lambda = \omega$

Convergence: general case

General case (here: ordinals):

- To ensure convergence of I_{ord}^{α}, it is sufficient to find λ s.t. the supremum of any sequence in nat $\rightarrow \lambda$ is an ordinal $< \lambda$.
  E.g. Card(nat)+1 or Card(WF(nat)).
- Can we avoid using choice and exclude-middle ? (I bet yes)
  Construction of bigger cardinals using the Burali-Forti paradox.
- Generalizes to W-types because we did not rely on specific properties of nat (besides its cardinality $< \lambda$).

Judgements for stage irrelevance

Contexts: $\Gamma ::= [\,] \mid \Gamma; (x : T) \mid \Gamma; (\alpha < O) \mid \Gamma; (f : A \rightsquigarrow B)$

- $(x : T)$ regular variable
- $(\alpha < O)$ ordinal variable bounded by ordinal expression O
- $(f : A \rightsquigarrow B)$ recursive function variable (domain depends on ordinals)

Valuations: $\rho_1 \prec \rho_2 \in [\Gamma]$ iff

- $\forall (x : T) \in \Gamma.\, \rho_1(x) = \rho_2(x) \in T\rho_1$
- $\forall (\alpha < O) \in \Gamma.\, \rho_1(\alpha) \leq \rho_2(\alpha) < O\rho_2$
- $\forall (f : A \rightsquigarrow B) \in \Gamma.$
  $\rho_1(f) \in A\rho_1 \to B\rho_1 \wedge \rho_2(f) \in A\rho_2 \to B\rho_2 \wedge \rho_1(f) \equiv_{A\rho_1} \rho_2(f)$

Judgements:

- Monotonicity: $[\Gamma \vdash M \;\; \uparrow] \triangleq \forall \rho_1\, \rho_2 . \rho_1 \prec \rho_2 \in [\Gamma] \Rightarrow M\rho_1 \subseteq M\rho_2$
- Invariance: $[\Gamma \vdash M \;\; =] \triangleq \forall \rho_1\, \rho_2 . \rho_1 \prec \rho_2 \in [\Gamma] \Rightarrow M\rho_1 = M\rho_2$
- Domain: $[\Gamma \vdash M \;(\mathrm{dom}\; A)] \triangleq \forall \rho_1\, \rho_2 . \rho_1 \prec \rho_2 \in [\Gamma] \Rightarrow M\rho_1 \equiv_{A\rho_1} M\rho_2$

Rule samples

(Not: $\Gamma \vdash M : (x : T) \rightsquigarrow U \triangleq \Gamma \vdash M : \Pi x : T.\, U \ \wedge\ \Gamma \vdash M\ (\mathrm{dom}\ T)$)

$$\frac{(\beta < \alpha^+); (x : I^\beta) \vdash U \uparrow \qquad (\beta < \alpha^+); (f : (x : I^\beta) \rightsquigarrow U_{\beta,x}) \vdash M : (x : I^{\beta^+}) \rightsquigarrow U_{\beta^+,x}}{\vdash \mathrm{Fix}(\beta f.M,\ \alpha) : (x : I^\alpha) \rightsquigarrow U_{\alpha,x}}$$

$$\frac{\vdash T \uparrow \qquad (x : T) \vdash_= M : U}{\vdash \lambda x : T.\, M : (x : T) \rightsquigarrow U} \qquad \frac{\vdash M : (x : T) \rightsquigarrow U \qquad \vdash_= N : T}{\vdash_= M\ N : U\{x \backslash N\}}$$

$$\frac{(\beta < \alpha) \in \Gamma}{\Gamma \vdash \beta \uparrow} \qquad \frac{\vdash O \uparrow}{\vdash O^+ \uparrow} \qquad \frac{\vdash O \uparrow}{\vdash I^O \uparrow} \qquad \frac{\vdash T = \qquad (x : T) \vdash U \uparrow}{\vdash \Pi x : T.\, U \uparrow}$$

Expressivity: examples

- Recursor:
  $[\vdash \mathit{Nrec} : \Pi P : \mathsf{nat} \to \mathsf{Prop}.P(0) \to (\Pi k.P(k) \to P(S(k))) \to \Pi n.P(n)]$
- Annotated subtraction: $[\alpha < \infty \vdash \mathsf{minus}_\alpha : \mathsf{nat}^\alpha \to \mathsf{nat} \to \mathsf{nat}^\alpha]$
- Cannot deal with min : $\mathsf{nat}^\alpha \to \mathsf{nat}^\alpha \to \mathsf{nat}^\alpha$

Strong Normalization

Strong Normalization of CC+NAT (recursor)

$$
\begin{aligned}
\mathsf{Fam} &\triangleq \mathsf{Nat} \to \mathsf{SAT} \quad \text{(family of saturated sets = realizability relation)}\\
\mathcal{F}_{\mathsf{nat}}(A) &\triangleq k \mapsto \bigcap_{P \in \mathsf{Fam}} P(0) \to \left(\bigcap_{n} A(n) \to P(n) \to P(S(n))\right) \to P(k)\\
J &\triangleq \{P \in \mathsf{Fam} \mid \forall k, \mathcal{F}_{\mathsf{nat}}(P,k) \subseteq P(k)\}\\
\Vdash_{\mathsf{Nat}} &\triangleq n \mapsto \bigcap_{P \in J} P(n)
\end{aligned}
$$

$$
\begin{array}{lll}
\mathsf{nat}(O) & \triangleq (\rho \mapsto \left\langle \mathsf{El} = \mathsf{Nat}^{O\rho};\ \mathsf{Sat} = \Vdash_{\mathsf{nat}} \right\rangle, & \sigma \mapsto K)\\
\mathsf{Ze} & \triangleq (\rho \mapsto 0, & \sigma \mapsto \lambda x f.\ x)\\
\mathsf{Su}(n) & \triangleq (\rho \mapsto S(n\rho), & \sigma \mapsto \lambda x f.\ f\, n\sigma\, (n\sigma\, x\, f))\\
\mathsf{Nrec}(n,f,g) & \triangleq (\rho \mapsto \mathsf{Natrec}(n\rho, f\rho, g\rho), & \sigma \mapsto n\sigma\ f\sigma\ g\sigma)
\end{array}
$$

Future Work

Strong normalization of the fixpoint operator

Issues:

- Simulate Fix's weird reduction strategy in the pure λ-calculus (or extend the pure λ-calculus with G s.t. $G\ (\lambda x.t) \rightarrow \lambda x.x$ and $G\ t' \nrightarrow$ if t' not an abstraction)
- Reductibility issue: realizers of non-constructor guarded inductive expressions (e.g. I^{α} with α ordinal variable) have to be neutral: `fix F n := match n with S (S k) =>F (S k) | ... end` should be rejected (because `F (S (S 0))` is not SN). This will be the case if `S k` is not a realizer of Nat^{α} for any ordinal variable α.

Dealing with empty inductive definitions

Strong normalization requires all types are inhabited

- Distinguish *total values* and *partial values*
  Qu: does extensionality forces us to have strict evaluation?
- Either *one value* belongs to every type
- Or each type carries its default value

Solution 2: match depends on the return type (B):

$$(\text{match } \bullet(\text{NAT}) \text{ with } \dots \text{ end} : B) = \bullet(B)$$

Solution 1: $\emptyset$ in all types

- already in all function types
- inductive types: add an extra constructor $\emptyset$ (new values: $\mathsf{S}(\emptyset)$, $\mathsf{S}(\mathsf{S}(\emptyset))$, ...)

$$\text{match } \emptyset \text{ with } \dots \text{ end} = \emptyset$$

Coinductive types

Various approaches. Among them:

- Process with hidden state $\nu X.F(X) \triangleq \Sigma Y.Y \times (Y \to F(Y))$
  - Impredicative definition
  - Cannot be extensional
- Set of compatible well-founded approximations with a closure property:
  - Add an extra constructor $\perp$
  - Extentional

  Stream: set of finite prefixes (lists)

But syntax is the difficult part...