Higher Inductive Types in Coinductive Definitions via Guarded Recursion.

Magnus Baunsgaard Kristensen **Rasmus Ejlers Møgelberg**
Andrea Vezzosi

IT University of Copenhagen

April 25, 2023

Coinduction

- Streams

$$\mathsf{Str}(A) \simeq A \times \mathsf{Str}(A)$$

- Productive stream definitions

$$\text{zeros} := 0 :: \text{zeros}$$
$$\text{map}\, f\, (x :: xs) := f(x) :: (\text{map}\, f\, xs)$$

- Non-productive stream definitions

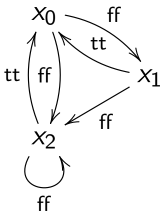
$$\text{undef} := 0 :: \text{tl}(\text{undef})$$

- Encoding productivity
  - Syntactic checks
  - Sized types
  - Guarded recursion
- Working directly with corecursion (Paco)

Coinduction

- Much previous work on M-types
- But how about non-deterministic processes?

$$\mathrm{LTS} \simeq \mathrm{P_f}(A \times \mathrm{LTS})$$

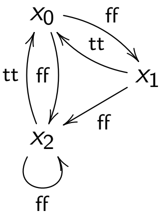


- Or probabilistic processes?

Coinduction in type theory

- How to represent functors like $\mathsf{P_f}$?
- How to recursively define *productive* processes?

$$
\begin{aligned}
\mathsf{LTS} &\simeq \mathsf{P_f}(A \times \mathsf{LTS}) \\
x_0 &= \mathsf{fold}(\{(\mathsf{ff}, x_1), (\mathsf{ff}, x_2)\}) \\
x_1 &= \mathsf{fold}(\{(\mathsf{tt}, x_0), (\mathsf{ff}, x_2)\}) \\
x_2 &= \mathsf{fold}(\{(\mathsf{tt}, x_0), (\mathsf{ff}, x_2)\})
\end{aligned}
$$



- How to ensure

$$(x = y) \simeq \mathsf{Bisim}(x, y)$$

Clocked Cubical Type Theory (CCTT)

- Represent $\mathsf{P_f}$ as higher-inductive type (HIT)
- Encode coinductive types using multiclocked guarded recursion

$$\mathsf{LTS}^{\kappa} \simeq \mathsf{P_f}(A \times \triangleright^{\kappa}\mathsf{LTS}^{\kappa}) \qquad \mathsf{LTS} \stackrel{\text{def}}{=} \forall\kappa.\mathsf{LTS}^{\kappa}$$

- Program and reason about coinductive data using guarded recursion
- Results in paper (Kristensen et al. [2022])
  - Definition of CCTT
  - Principle of induction under clocks
  - Computational contents to clock irrelevance
  - Denotational semantics
- Partially implemented in extension of Cubical Agda

Clocked Cubical Type Theory

Extend Cubical Type theory ...

- Path types

$$\frac{\Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \lambda i.\, t : \mathrm{Path}_A(t[0/i], t[1/i])}$$

$$\frac{\Gamma \vdash p : \mathrm{Path}_A(a_0, a_1) \qquad \Gamma \vdash r : \mathbb{I}}{\Gamma \vdash p\, r : A}$$

- Composition

$$\frac{\begin{array}{cc} \Gamma \vdash \varphi : \mathbb{F} & \Gamma, i : \mathbb{I} \vdash A \text{ type} \\ \Gamma, \varphi, i : \mathbb{I} \vdash u : A & \Gamma \vdash u_0 : A[0/i][\varphi \mapsto u[0/i]] \end{array}}{\Gamma \vdash \mathrm{comp}^i_A\, [\varphi \mapsto u]\, u_0 : A[1/i][\varphi \mapsto u[1/i]]}$$

- Glueing, HITs, ...

. . . with multiclocked guarded recursion

- Clocks

$$\frac{\Gamma, \kappa : \mathsf{clock} \vdash t : A}{\Gamma \vdash \lambda\kappa.t : \forall\kappa.A} \qquad \frac{\Gamma \vdash t : \forall\kappa.A \qquad \Gamma \vdash \kappa' : \mathsf{clock}}{\Gamma \vdash t[\kappa'] : A[\kappa'/\kappa]}$$

- Later type

$$\frac{\Gamma \vdash \kappa : \mathsf{clock} \qquad \alpha \notin \Gamma}{\Gamma, \alpha : \kappa \vdash} \qquad \frac{\Gamma, \alpha : \kappa \vdash t : A}{\Gamma \vdash \triangleright(\alpha : \kappa).A \ \mathsf{type}}$$

- Introduction and (simplified) elimination rules

$$\frac{\Gamma, \alpha : \kappa \vdash t : A}{\Gamma \vdash \lambda(\alpha : \kappa).t : \triangleright(\alpha : \kappa).A} \qquad \frac{\Gamma \vdash t : \triangleright(\alpha : \kappa).A}{\Gamma, \beta : \kappa, \Gamma' \vdash t\,[\beta] : A[\beta/\alpha]}$$

- Force

$$\forall\kappa. \triangleright^{\kappa} A \simeq \forall\kappa.A$$

Fixed points

$$\frac{\Gamma \vdash t : \triangleright^{\kappa} A \to A}{\Gamma \vdash \mathsf{dfix}^{\kappa}\, t : \triangleright^{\kappa} A}$$

$$\frac{\Gamma \vdash t : \triangleright^{\kappa} A \to A}{\Gamma \vdash \mathsf{pfix}^{\kappa}\, t : \triangleright(\alpha : \kappa).\mathsf{Path}_A((\mathsf{dfix}^{\kappa}\, t)\,[\alpha], t(\mathsf{dfix}^{\kappa}\, t))}$$

- **Lemma:** The type $\Sigma(x : A).\mathsf{Path}_A(x, f\,(\lambda(\alpha : \kappa).x))$ is contractible for every $f : \triangleright^{\kappa} A \to A$.
- Nakano fixed point operator

$$\mathsf{fix}^{\kappa} \stackrel{\text{def}}{=} \lambda f. f(\mathsf{dfix}^{\kappa} f) : (\triangleright^{\kappa} A \to A) \to A$$

Example: Streams

- Guarded stream type

$$\overline{\mathsf{Str}^{\kappa}}(\mathbb{N}) \stackrel{\text{def}}{=} \mathsf{fix}^{\kappa}(\lambda X.\overline{\mathbb{N}}\,\overline{\times}\,\overline{\triangleright}\,(\alpha:\kappa).X\,[\alpha]) \qquad : \mathsf{U}$$

$$\mathsf{Str}^{\kappa}(\mathbb{N}) \stackrel{\text{def}}{=} \mathsf{El}\,(\overline{\mathsf{Str}^{\kappa}}(\mathbb{N}))$$

- Using

$$\frac{\Gamma, \alpha : \kappa \vdash A : \mathsf{U}}{\Gamma \vdash \overline{\triangleright}\,(\alpha:\kappa).A : \mathsf{U}}$$

- Then

$$\mathsf{Str}^{\kappa}(\mathbb{N}) \simeq \mathbb{N} \times \triangleright^{\kappa}\mathsf{Str}^{\kappa}(\mathbb{N})$$

- Recursive programs

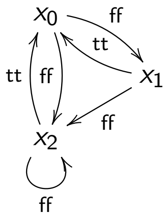
$$\mathsf{zeros} \stackrel{\text{def}}{=} \mathsf{fix}^{\kappa}(\lambda xs.0 :: xs)$$

Example: Guarded LTS

- Guarded LTS type

$$\mathsf{LTS}^\kappa \simeq \mathsf{P_f}(A \times \triangleright^\kappa \mathsf{LTS}^\kappa)$$

- Guarded recursive definitions



$$\begin{aligned}\mathsf{fix}^\kappa(\lambda x : \triangleright^\kappa(\mathsf{LTS}^\kappa)^3.(&\mathsf{fold}(\{(\mathsf{ff}, \triangleright^\kappa(\pi_1)(x)), (\mathsf{ff}, \triangleright^\kappa(\pi_2)(x))\}),\\ &\mathsf{fold}(\{(\mathsf{tt}, \triangleright^\kappa(\pi_0)(x)), (\mathsf{ff}, \triangleright^\kappa(\pi_2)(x))\}),\\ &\mathsf{fold}(\{(\mathsf{tt}, \triangleright^\kappa(\pi_0)(x), (\mathsf{ff}, \triangleright^\kappa(\pi_2)(x)))\})))\end{aligned}$$

Encoding coinductive types

Encoding coinductive types

- **Definition.** A functor[1] $F : (I \to U) \to (I \to U)$ *commutes with clock quantification* if

$$F(\forall \kappa. X) \simeq \forall \kappa. F(X)$$

- **Definition.** $g : Y \to F(Y)$ is a *final coalgebra* if the following type is contractible for all $f : X \to F(X)$

$$\Sigma(h : X \to Y).\, g \circ h = F(h) \circ f$$

- **Theorem.** If F commutes with clock quantification, then $\nu(F)$ is final coalgebra

$$\nu^\kappa(F) \simeq F(\triangleright^\kappa(\nu^\kappa(F))$$

$$\nu(F) \stackrel{\text{def}}{=} \forall \kappa. \nu^\kappa(F)$$

[1] in the naive sense

Encoding coinductive types

- **Theorem.** If F commutes with clock quantification, then $\nu(F)$ is final coalgebra

$$\nu^\kappa(F) \simeq F(\triangleright^\kappa(\nu^\kappa(F))$$

$$\nu(F) \stackrel{\text{def}}{=} \forall\kappa.\nu^\kappa(F)$$

- Note

$$\begin{aligned}\nu(F) &\simeq \forall\kappa.F(\triangleright^\kappa \nu^\kappa(F)) \\ &\simeq F(\forall\kappa.\triangleright^\kappa \nu^\kappa(F)) \\ &\simeq F(\forall\kappa.\nu^\kappa(F)) \\ &= F(\nu(F))\end{aligned}$$

Example: Streams

- Functor

$$F(X) = \mathbb{N} \times X$$

- Type equivalences

$$\begin{aligned} \forall\kappa.F(X) &= \forall\kappa.(\mathbb{N} \times X) \\ &\simeq (\forall\kappa.\mathbb{N}) \times (\forall\kappa.X) \end{aligned}$$

- Need $\mathbb{N}$ clock irrelevant

$$\mathbb{N} \simeq \forall\kappa.\mathbb{N}$$

Examples

- The delay monad $LA \simeq A + LA$
- Functor

$$F(X) = A + X$$

- Equivalence

$$\begin{aligned}\forall\kappa.F(X) &= \forall\kappa.(A + X)\\ &\simeq (\forall\kappa.A) + (\forall\kappa.X)\\ &\simeq A + \forall\kappa.X\\ &= F(\forall\kappa.X)\end{aligned}$$

- So need A clock irrelevant and

$$\forall\kappa.(X + Y) \simeq (\forall\kappa.X) + (\forall\kappa.Y)$$

Example: Non-deterministic processes

- Encoding non-deterministic processes

$$\mathsf{LTS}^\kappa \simeq \mathsf{P_f}(A \times \triangleright^\kappa \mathsf{LTS}^\kappa) \qquad \mathsf{LTS} \overset{\mathrm{def}}{=} \forall\kappa.\mathsf{LTS}^\kappa$$

- Requires

$$\begin{aligned}\forall\kappa.\mathsf{P_f}(A \times X) &\simeq \mathsf{P_f}(\forall\kappa.(A \times X)) \\ &\simeq \mathsf{P_f}((\forall\kappa.A) \times (\forall\kappa.X)) \\ &\simeq \mathsf{P_f}(A \times (\forall\kappa.X))\end{aligned}$$

Induction under clocks

Induction under clocks

- New principle for HITs
- Case of Bool:

$$\frac{\Gamma, x : \forall \vec{\kappa}.\mathsf{Bool} \vdash C(x)\ \mathsf{type} \qquad \Gamma \vdash u_{\mathsf{tt}} : C(\lambda \vec{\kappa}.\mathsf{tt}) \qquad \Gamma \vdash u_{\mathsf{ff}} : C(\lambda \vec{\kappa}.\mathsf{ff}) \qquad \Gamma \vdash t : \forall \vec{\kappa}.\mathsf{Bool}}{\Gamma \vdash \mathsf{elim}_C(u_{\mathsf{tt}}, u_{\mathsf{ff}}, t) : C(t)}$$

- Plus definitional equalities

$$\mathsf{elim}_C(u_{\mathsf{tt}}, u_{\mathsf{ff}}, \lambda \vec{\kappa}.\mathsf{tt}) \equiv u_{\mathsf{tt}}$$
$$\mathsf{elim}_C(u_{\mathsf{tt}}, u_{\mathsf{ff}}, \lambda \vec{\kappa}.\mathsf{ff}) \equiv u_{\mathsf{ff}}$$

Proving Bool clock irrelevant

$$\frac{\Gamma, x : \forall\vec{\kappa}.\mathrm{Bool} \vdash C(x)\ \mathrm{type} \qquad \Gamma \vdash u_{\mathrm{tt}} : C(\lambda\vec{\kappa}.\mathrm{tt}) \qquad \Gamma \vdash u_{\mathrm{ff}} : C(\lambda\vec{\kappa}.\mathrm{ff}) \qquad \Gamma \vdash t : \forall\vec{\kappa}.\mathrm{Bool}}{\Gamma \vdash \mathrm{elim}_C(u_{\mathrm{tt}}, u_{\mathrm{ff}}, t) : C(t)}$$

- Used to construct terms

$$\lambda x.\lambda\kappa.x : \mathrm{Bool} \to \forall\kappa.\mathrm{Bool}$$
$$\lambda x.\mathrm{elim}_{\mathrm{Bool}}(\mathrm{tt}, \mathrm{ff}, x) : (\forall\kappa.\mathrm{Bool}) \to \mathrm{Bool}$$

- and prove Bool clock irrelevant

$$\lambda x.\mathrm{elim}(\mathrm{refl}, \mathrm{refl}, x) : \Pi(x : \forall\kappa.\mathrm{Bool}).x = \lambda\kappa.\mathrm{elim}_{\mathrm{Bool}}(\mathrm{tt}, \mathrm{ff}, x)$$

Induction under clocks for $\mathbb{N}$

$$\frac{\Gamma, x : \forall\kappa.\mathbb{N} \vdash C(x) \text{ type} \qquad \Gamma \vdash u_0 : C(\lambda\kappa.0) \\ \Gamma, x : \forall\kappa.\mathbb{N}, y : C(x) \vdash u_s : C(\lambda\kappa.s(x\,[\kappa])) \qquad \Gamma \vdash t : \forall\kappa.\mathbb{N}}{\Gamma \vdash \mathsf{elim}_C(u_0, u_s, t) : C(t)}$$

- Plus definitional equalities

$$\begin{aligned} \mathsf{elim}_C(u_0, u_s, \lambda\kappa.0) &\equiv u_0 \\ \mathsf{elim}_C(u_0, u_s, \lambda\kappa.s(n)) &\equiv u_s(\lambda\kappa.n, \mathsf{elim}_C(u_0, u_s, \lambda\kappa.n)) \end{aligned}$$

- Used to prove $\mathbb{N}$ clock-irrelevant

Induction under clocks for spheres

$$\mathsf{base} : \mathbb{S}^n$$

$$\mathsf{surface} : (\bar{i} : \mathbb{I}^n) \to \mathbb{S}^n \left[\bigvee_{0 \le k < n} (i_k = 0 \vee i_k = 1) \mapsto \mathsf{base} \right]$$

- Induction under clocks

$$\frac{\Gamma, x : \forall\kappa.\mathbb{S}^n \vdash D \text{ type} \quad \Gamma \vdash u_b : D[\lambda\kappa.\mathsf{base}] \quad \Gamma \vdash t : \forall\kappa.\mathbb{S}^n \qquad \Gamma, \bar{i} : \mathbb{I}^n \vdash u_s : D[\lambda\kappa.\mathsf{surface}(\bar{i})] \left[\bigvee_{0 \le k < n} (i_k = 0 \vee i_k = 1) \mapsto u_b \right]}{\Gamma \vdash \mathsf{elim}_D(u_b, u_s, t) : D[t]}$$

- Definitional equalities

$$\mathsf{elim}_D(u_b, u_s, \lambda\kappa.\mathsf{base}) = u_b$$
$$\mathsf{elim}_D(u_b, u_s, \lambda\kappa.\mathsf{surface}(\bar{i})) = u_s$$

Spheres clock irrelevant

- Induction under clocks

$$\frac{\Gamma, x : \forall\kappa.\mathbb{S}^n \vdash D \text{ type} \quad \Gamma \vdash u_b : D[\lambda\kappa.\text{base}] \quad \Gamma \vdash t : \forall\kappa.\mathbb{S}^n \qquad \Gamma, \bar{i} : \mathbb{I}^n \vdash u_s : D[\lambda\kappa.\text{surface}(\bar{i})] \left[\bigvee_{0 \le k < n} (i_k = 0 \vee i_k = 1) \mapsto u_b \right]}{\Gamma \vdash \text{elim}_D(u_b, u_s, t) : D[t]}$$

- Equivalence

$$\lambda x.\lambda\kappa.x : \mathbb{S}^n \to \forall\kappa.\mathbb{S}^n$$

$$\lambda x.\text{elim}_D(\text{base}, \text{surface}(\bar{i}), x) : \forall\kappa.\mathbb{S}^n \to \mathbb{S}^n$$

Propositional truncation

$$\mathsf{in} : A \to \|A\|_{-1}$$

$$\mathsf{squash} : (a_0, a_1 : \|A\|_{-1}) \to (i : \mathbb{I}) \to \|A\|_{-1} \left[\begin{array}{l} i = 0 \mapsto a_0 \\ i = 1 \mapsto a_1 \end{array} \right]$$

- Induction under clocks

$$\frac{\begin{array}{c} \Gamma \vdash A : \forall\kappa.\mathsf{U} \quad \Gamma, x : \forall\kappa.\|A\,[\kappa]\|_{-1} \vdash D \text{ type} \\ \Gamma, x : \forall\kappa.A \vdash u_{\mathsf{in}} : D[\lambda\kappa.\mathsf{in}(x\,[\kappa])] \qquad \Gamma \vdash t : \forall\kappa.\|A\,[\kappa]\|_{-1} \\ \Gamma, a_0, a_1 : \forall\kappa.\|A\,[\kappa]\|_{-1}, y_0 : D[a_0], y_1 : D[a_1], i : \mathbb{I} \\ \vdash u_{\mathsf{sq}} : D[\lambda\kappa.\mathsf{squash}(a_0\,[\kappa], a_1\,[\kappa], i)] \left[\begin{array}{l} i = 0 \mapsto y_0 \\ i = 1 \mapsto y_1 \end{array} \right] \end{array}}{\Gamma \vdash \mathsf{elim}_D(u_{\mathsf{in}}, u_{\mathsf{squash}}, t) : D[t]}$$

Truncation commutes with clock abstraction

$$\frac{\begin{array}{c}\Gamma \vdash A : \forall\kappa.\mathsf{U} \quad \Gamma, x : \forall\kappa.\|A\,[\kappa]\|_{-1} \vdash \|\forall\kappa.A\,[\kappa]\|_{-1} \text{ type} \\ \Gamma, x : \forall\kappa.A \vdash \mathsf{in}(x) : \|\forall\kappa.A\,[\kappa]\|_{-1} \qquad \Gamma \vdash t : \forall\kappa.\|A\,[\kappa]\|_{-1} \\ \Gamma, a_0, a_1 : \forall\kappa.\|A\,[\kappa]\|_{-1}, y_0 : \|\forall\kappa.A\,[\kappa]\|_{-1}, y_1 : \|\forall\kappa.A\,[\kappa]\|_{-1}, i : \mathbb{I} \\ \vdash \mathsf{squash}(y_0, y_1, i) : \|\forall\kappa.A\,[\kappa]\|_{-1} \begin{bmatrix} i = 0 \mapsto y_0 \\ i = 1 \mapsto y_1 \end{bmatrix}\end{array}}{\Gamma \vdash \mathsf{elim}_D(\mathsf{in}, \mathsf{squash}, t) : \|\forall\kappa.A\,[\kappa]\|_{-1}}$$

- Equivalence

$$\lambda x.\lambda\kappa.\|(-)\,[\kappa]\|_{-1}(x) : \|\forall\kappa.A\,[\kappa]\|_{-1} \to \forall\kappa.\|A\,[\kappa]\|_{-1}$$
$$\mathsf{elim}_D(\mathsf{in}, \mathsf{squash}, -) : \forall\kappa.\|A\,[\kappa]\|_{-1} \to \|\forall\kappa.A\,[\kappa]\|_{-1}$$

- Generalises to higher truncations as well

Finite powersets as a HIT

$$
\begin{aligned}
\emptyset &: \mathsf{P_f}(A) \\
\{-\} &: A \to \mathsf{P_f}(A) \\
\cup &: \mathsf{P_f}(A) \to \mathsf{P_f}(A) \to \mathsf{P_f}(A) \\
\mathsf{nl} &: \Pi(X : \mathsf{P_f}(A)).\, X \cup \emptyset = X \\
\mathsf{assoc} &: \Pi(X\ Y\ Z : \mathsf{P_f}(A)).\, X \cup (Y \cup Z) = (X \cup Y) \cup Z \\
\mathsf{comm} &: \Pi(X\ Y : \mathsf{P_f}(A)).\, X \cup Y = Y \cup X \\
\mathsf{idem} &: \Pi(X : \mathsf{P_f}(A)).\, X \cup X = X \\
\mathsf{trunc} &: \mathsf{isSet}(\mathsf{P_f}(A))
\end{aligned}
$$

Induction under clocks for $\mathsf{P_f}$

- Assuming $\Gamma \vdash X : \forall\kappa.\mathsf{U}$

$$\frac{\begin{array}{c}\Gamma, x : \forall\kappa.\mathsf{P_f}(X\,[\kappa]) \vdash C(x)\ \mathsf{type} \qquad \Gamma \vdash u_\emptyset : C(\lambda\kappa.\emptyset) \\ \Gamma, x : \forall\kappa.(X\,[\kappa]) \vdash u_{\{-\}} : C(\lambda\kappa.\{x\,[\kappa]\}) \\ \Gamma, x, x' : \forall\kappa.\mathsf{P_f}(X\,[\kappa]), y : C(x), y' : C(x') \vdash u_\cup : C(\lambda\kappa.x\,[\kappa] \cup x'\,[\kappa]) \\ \dots \\ \Gamma, x : \forall\kappa.\mathsf{P_f}(A\,[\kappa]), y : C(x), i : \mathbb{I} \vdash u_{\mathsf{idem}} : C(\lambda\kappa.\mathsf{idem}(x\,[\kappa], i)) \\ u_{\mathsf{idem}}(x, y, 0) \equiv u_\cup(x, x, y, y) \qquad u_{\mathsf{idem}}(x, y, 1) \equiv y \\ \dots \qquad \Gamma \vdash t : \forall\kappa.\mathsf{P_f}(X\,[\kappa])\end{array}}{\Gamma \vdash \mathsf{elim}_C(u_\emptyset, u_{\{-\}}, \dots, t) : C(t)}$$

- Can be used to prove

$$\forall\kappa.\mathsf{P_f}(X\,[\kappa]) \simeq \mathsf{P_f}(\forall\kappa.X\,[\kappa])$$

Bisimilarity as guarded recursive type

- Let $x, y : \mathsf{LTS}$

$$\mathsf{Bisim}^{\kappa}(x, y) \stackrel{\text{def}}{=} \mathsf{Sim}^{\kappa}(x, y) \times \mathsf{Sim}^{\kappa}(y, x)$$

- Where

$$\begin{aligned}\mathsf{Sim}^{\kappa}(x, y) \simeq \Pi(x' : \mathsf{LTS}, a : A).(a, x') \in \mathsf{ufld}(x) \to \\ \exists y' : \mathsf{LTS}.(a, y') \in \mathsf{ufld}(y) \times \triangleright(\alpha : \kappa).\mathsf{Bisim}^{\kappa}(x', y')\end{aligned}$$

- Then coinductive bisimilarity for $x, y : \mathsf{LTS}$ is

$$\mathsf{Bisim}(x, y) \stackrel{\text{def}}{=} \forall \kappa.\mathsf{Bisim}^{\kappa}(x, y)$$

- **Theorem.** $(x = y) \simeq \mathsf{Bisim}(x, y)$

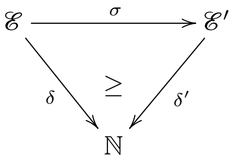
Denotational semantics

Denotational semantics

- Cubical Type Theory modelled in $\mathrm{PSh}(\mathscr{C})$
- Clocked Type Theory modelled in $\mathrm{PSh}(\mathscr{T})$
- Category $\mathscr{T}$ of time objects
  - Objects:

$$(\mathscr{E}, \delta : \mathscr{E} \to \mathbb{N})$$

  - $\mathscr{E}$ finite set (of clocks)
  - Morphisms $\sigma : (\mathscr{E}, \delta) \to (\mathscr{E}', \delta')$:



- Clocked Cubical Type Theory modelled in $\mathrm{PSh}(\mathscr{C} \times \mathscr{T})$

Model construction

- Model constructed in internal language following Licata et al. [2018]
- Interval type and cofibrations imported from $\mathrm{PSh}(\mathscr{C})$
- $\triangleright$ and ticks modelled using dependent right adjoint
- Composition structure on $\triangleright$ from general theorem proved in paper

Modelling guarded recursion

- Object of clocks

$$\mathrm{Clk}(I, \mathscr{E}, \delta) \stackrel{\mathsf{def}}{=} \mathscr{E}$$

- $\triangleright$ is a modal operator on slice

$$\mathsf{PSh}(\mathscr{C} \times \mathscr{T})/\mathrm{Clk}$$

- Defined on closed types as

$$\triangleright A(I, \mathscr{E}, \delta, \lambda) = \begin{cases} A(I, \mathscr{E}, \delta[\lambda \mapsto n], \lambda) & \text{if } \delta(\lambda) = n+1 \\ 1 & \text{if } \delta(\lambda) = 0 \end{cases}$$

- Fixed points defined by natural number induction

Quantification over clocks

- Object of clocks
$$\mathrm{Clk}(I, \mathscr{E}, \delta) \stackrel{\mathsf{def}}{=} \mathscr{E}$$
- $(\Pi(\kappa : \mathrm{Clk}).A)(I, \mathscr{E}, \delta)$ isomorphic to limit of sequence
$$A(I, (\mathscr{E}, \lambda), \delta[\lambda \mapsto 0]) \leftarrow A(I, (\mathscr{E}, \lambda), \delta[\lambda \mapsto 1]) \leftarrow \dots$$
- So
$$\Pi(\kappa : \mathrm{Clk}).(A + B) \simeq (\Pi(\kappa : \mathrm{Clk}).A) + (\Pi(\kappa : \mathrm{Clk}).B)$$

Modelling induction under clocks

- Model HITs following Coquand et al. [2018]
- This means each element in $H(I, \mathscr{E}, \delta)$ is either
  - A constructor for H (including those for paths), or
  - An hcomp
- Each map in diagram

$$H(I, (\mathscr{E}, \lambda), \delta[\lambda \mapsto 0]) \leftarrow H(I, (\mathscr{E}, \lambda), \delta[\lambda \mapsto 1]) \leftarrow \dots$$

- preserves this structure strictly
- This allows us to prove soundness of induction under clocks principle

Coinductive types in model

- To commute with $\forall\kappa$ means to commute with ω-limits
- Coinductive types interpreted as limits

$$F(1)(I,\mathscr{E},\delta) \leftarrow F^2(1)(I,\mathscr{E},\delta) \leftarrow F^3(1)(I,\mathscr{E},\delta) \leftarrow \dots$$

- Also case of $F = \mathsf{P_f}$ is ω-limit
- In set theory final coalgebra for $\mathsf{P_f}$ is constructed in $\omega + \omega$ steps (Worrell [2005])
- Veltri [2021] internalises Worrell's construction in Cubical Agda

Coinductive types in model

- To commute with $\forall\kappa$ means to commute with ω-limits
- Coinductive types interpreted as limits

$$F(1)(I, \mathscr{E}, \delta) \leftarrow F^2(1)(I, \mathscr{E}, \delta) \leftarrow F^3(1)(I, \mathscr{E}, \delta) \leftarrow \dots$$

- Also case of $F = \mathrm{P_f}$ is ω-limit
- In set theory final coalgebra for $\mathrm{P_f}$ is constructed in $\omega + \omega$ steps (Worrell [2005])
- Veltri [2021] internalises Worrell's construction in Cubical Agda
- $\mathrm{PSh}(\mathscr{C} \times \mathscr{T})$ is a category of **strict** cubical presheaves over $\mathscr{T}$
- Cobar construction would not model equivalence

$$\mathrm{P_f}(\forall\kappa.-) \simeq \forall\kappa.\mathrm{P_f}(-)$$

Beyond coinduction

Models of untyped lambda calculus with non-determinism

- Two guarded powerdomains

$$\mathsf{P}^{\kappa}_{\Diamond}(A) \simeq \mathsf{P}_{\mathsf{f}}(A + \triangleright^{\kappa}\mathsf{P}^{\kappa}_{\Diamond}(A))$$
$$\mathsf{P}^{\kappa}_{\Box}(A) \stackrel{\text{def}}{=} \mathsf{L}^{\kappa}\mathsf{P}_{\mathsf{f}}(A)$$

- Denotational model ($T = \mathsf{P}^{\kappa}_{\Diamond}$ or $\mathsf{P}^{\kappa}_{\Box}$)

$$\mathsf{SVal}^{\kappa} \stackrel{\text{def}}{=} (\triangleright^{\kappa}(\mathsf{SVal}^{\kappa} \to T(\mathsf{SVal}^{\kappa})))$$

- Used to prove applicative bisimilarity a congruence
- Details in (Møgelberg and Vezzosi [2021])
- Other applications: Models of higher-order store (Sterling et al. [2022])

Conclusion

Results

- General theorem for encoding of coinductive types
- Induction under clocks formulated for general schema (Cavallo and Harper [2019]) for HITs
- Denotational semantics in $\mathsf{PSh}(\mathscr{C} \times \mathscr{T})$
- Give computational content to *tick irrelevance axiom*

$$\mathsf{tirr}^{\kappa} : \Pi(x : \triangleright^{\kappa} A).\triangleright(\alpha : \kappa).\triangleright(\beta : \kappa).\mathsf{Path}_A(x\,[\alpha], x\,[\beta])$$

- **Conjecture (Canonicity).** Any term $\vec{i} : \mathbb{I}, \vec{\kappa} : \mathsf{clock} \vdash t : \mathbb{N}$ is equal to either 0 or a successor.

Future work

- Clock-irrelevant universes of clock-irrelevant types
- Proving canonicity
- Implementation of induction under clocks in Agda
- Start using Guarded Cubical Agda for coinduction and program verification!

References I

M. B. Kristensen, R. E. Møgelberg, and A. Vezzosi. Greatest hits: Higher inductive types in coinductive definitions via induction under clocks. In *LICS*, 2022.

R. D. Licata, I. Orton, A.M. Pitts, and B. Spitters. Internal universes in models of homotopy type theory. In *FSCD*, 2018.

T. Coquand, S. Huber, and A. Mörtberg. On higher inductive types in cubical type theory. In *LICS*, 2018.

James Worrell. On the final sequence of a finitary set functor. *Theor. Comput. Sci.*, 338(1-3):184–199, 2005.

Niccolò Veltri. Type-theoretic constructions of the final coalgebra of the finite powerset functor. In *FSCD*, 2021.

Rasmus Ejlers Møgelberg and Andrea Vezzosi. Two guarded recursive powerdomains for applicative simulation. In *MFPS*, volume 351 of *EPTCS*, pages 200–217, 2021.

References II

Jonathan Sterling, Daniel Gratzer, and Lars Birkedal. Denotational semantics of general store and polymorphism. Unpublished manuscript, July 2022.

Evan Cavallo and Robert Harper. Higher inductive types in cubical computational type theory. *Proceedings of the ACM on Programming Languages*, 3(POPL), 2019.