Guarded recursion in type theory

Rasmus Ejlers Møgelberg

IT University of Copenhagen

December 11, 2014

Joint work with Lars Birkedal, Jan Schwinghammer, Kristian Støvring

Overview

- Guarded recursion applications
  - Computing with streams
  - Modelling higher order store
- Model: the topos of trees
- Recursive types
- Intensional models
- Coinduction via guarded recursion

Motivation

Motivation 1: Computing with streams

- Which of the following streams are well-defined:

```
zeros = 0::zeros
xs = xs
```

- Much less obvious when higher types are involved

$$\texttt{mergef}:(\texttt{int} \to \texttt{int} \to \texttt{S(int)} \to \texttt{S(int)}) \to \texttt{S(int)} \to \texttt{S(int)} \to \texttt{S(int)}$$

```
mergef f x::xs y::ys = f x y (mergef f xs ys)
```

- A nonproductive example:

```
badf x y xs = xs
mergef badf x::xs y::ys = mergef badf xs ys
```

- (example due to Bob Atkey)

Capturing productivity in types

- Introduce modal operator $\blacktriangleright$

$$\texttt{S(int)} \;=\; \mu\texttt{X.int} \times \blacktriangleright X$$
$$\texttt{hd}\colon \texttt{S(int)} \to \texttt{int}$$
$$\texttt{tail}\colon \texttt{S(int)} \to \blacktriangleright \texttt{S(int)}$$
$$\texttt{cons}\colon \texttt{int} \times \blacktriangleright \texttt{S(int)} \to \texttt{S(int)}$$

- Fixed points

$$\texttt{fix}\colon (\blacktriangleright \texttt{S(int)} \to \texttt{S(int)}) \to \texttt{S(int)}$$
$$\texttt{zeros} \;=\; \texttt{fix}(\lambda\texttt{xs.0::xs})$$

Capturing productivity in types

- $\blacktriangleright$ is an applicative functor

$$\texttt{next} : \texttt{X} \to \blacktriangleright \texttt{X}$$
$$\circledast : \blacktriangleright (\texttt{X} \to \texttt{Y}) \to \blacktriangleright \texttt{X} \to \blacktriangleright \texttt{Y}$$

- Typing `mergef`

$$\texttt{mergef} : (\texttt{int} \to \texttt{int} \to \blacktriangleright \texttt{S(int)} \to \texttt{S(int)}) \to \texttt{S(int)} \to \texttt{S(int)} \to \texttt{S(int)}$$

$$\texttt{mergef f = fix}(\lambda \texttt{g}.\lambda(\texttt{x::xs})\lambda(\texttt{y::ys}).\texttt{f x y (g} \circledast \texttt{xs} \circledast \texttt{ys))}$$

- where

$$\texttt{g} : \blacktriangleright(\texttt{S(int)} \to \texttt{S(int)} \to \texttt{S(int)})$$

- Note: `mergef badf` is not well typed

Motivation 2: Modelling higher-order store

- Would like to solve (but can not)

$$\mathcal{W} = \mathsf{N} \to_{\mathrm{fin}} \mathcal{T} \qquad \mathcal{T} = \mathcal{W} \to_{\mathrm{mon}} \mathcal{P}(\mathrm{Value})$$

- Suffices to solve this equation

$$\widehat{\mathcal{T}} \cong \blacktriangleright((\mathsf{N} \to_{\mathrm{fin}} \widehat{\mathcal{T}}) \to_{\mathrm{mon}} \mathcal{P}(\mathrm{Value}))$$

- Can model higher-order store in expressive type theory with guarded recursion
- Synthetic presentation of step-indexed model!

The topos of trees

The topos of trees

- $\mathcal{S} = \mathbf{Set}^{\omega^{\mathrm{op}}}$
- Objects

$$X(1) \xleftarrow{r_1} X(2) \xleftarrow{r_2} X(3) \longleftarrow \dots$$

- Morphism

$$\begin{array}{ccccccc} X(1) & \longleftarrow & X(2) & \longleftarrow & X(3) & \longleftarrow & \dots \\ \downarrow f_1 & & \downarrow f_2 & & \downarrow f_3 & & \\ Y(1) & \longleftarrow & Y(2) & \longleftarrow & Y(3) & \longleftarrow & \dots \end{array}$$

- Example: object of streams of integers `S(int)`

$$\mathbb{Z} \xleftarrow{\pi} \mathbb{Z}^2 \xleftarrow{\pi} \mathbb{Z}^3 \xleftarrow{\pi} \dots$$

An endofunctor

- Define $\blacktriangleright X$

$$\{*\} \longleftarrow X(1) \longleftarrow X(2) \longleftarrow \dots$$

- Note that $\texttt{S(int)} \cong \mathbb{Z} \times \blacktriangleright \texttt{S(int)}$:

$$\mathbb{Z} \times 1 \xleftarrow{\mathbb{Z}\times !} \mathbb{Z} \times \mathbb{Z} \xleftarrow{\mathbb{Z} \times \pi} \mathbb{Z} \times \mathbb{Z}^2 \xleftarrow{\mathbb{Z} \times \pi} \dots$$

- Define $\mathrm{next} : X \to \blacktriangleright X$

$$\begin{array}{ccccccc}
X(1) & \xleftarrow{r_1} & X(2) & \xleftarrow{r_2} & X(3) & \longleftarrow & \dots \\
\downarrow & & \downarrow r_1 & & \downarrow r_2 & & \\
\{*\} & \longleftarrow & X(1) & \xleftarrow{r_1} & X(2) & \longleftarrow & \dots
\end{array}$$

Fixed points

- Fixed point operator

$$\mathrm{fix}_X : (\blacktriangleright X \to X) \to X$$

- Fixpoint property

$$f(\mathrm{next}(\mathrm{fix}_X(f))) = \mathrm{fix}_X(f)$$

- This fixed point is *unique*.

Fixed points

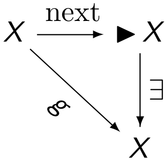
- Fixed point operator

$$\mathrm{fix}_X : (\blacktriangleright X \to X) \to X$$

- Fixpoint property

$$f(\mathrm{next}(\mathrm{fix}_X(f))) = \mathrm{fix}_X(f)$$

- This fixed point is *unique*.
- A morphism factoring through next is called *contractive*



- Contractive morphisms have *unique* fixed points

Construction of fixed points

- Given $f : \blacktriangleright X \to X$:

$$\begin{array}{ccccccc}
\{*\} & \longleftarrow & X(1) & \xleftarrow{r_1} & X(2) & \longleftarrow & \dots \\
\downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f_2} & & \downarrow{\scriptstyle f_3} & & \\
X(1) & \xleftarrow{r_1} & X(2) & \xleftarrow{r_2} & X(3) & \longleftarrow & \dots
\end{array}$$

- Construct $\mathrm{fix}_X(f) : 1 \to X$:

$$\begin{array}{ccccccc}
1 & \longleftarrow & 1 & \longleftarrow & 1 & \longleftarrow & \dots \\
\downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f_2 \circ f_1} & & \downarrow{\scriptstyle f_3 \circ f_2 \circ f_1} & & \\
X(1) & \xleftarrow{r_1} & X(2) & \xleftarrow{r_2} & X(3) & \longleftarrow & \dots
\end{array}$$

Guarded recursive types

Example

- Consider type constructor F

$$FX = \blacktriangleright X \times \mathbb{Z}$$

- $F(\texttt{S(int)}) \cong \texttt{S(int)}$

$$\begin{array}{lllllll}
X & X(1) \longleftarrow & X(2) \longleftarrow & X(3) \longleftarrow & X(4) \ldots \\
FX & 1 \times \mathbb{Z} \longleftarrow & X(1) \times \mathbb{Z} \longleftarrow & X(2) \times \mathbb{Z} \longleftarrow & X(3) \times \mathbb{Z} \ldots \\
F^2X & 1 \times \mathbb{Z} \longleftarrow & 1 \times \mathbb{Z}^2 \longleftarrow & X(1) \times \mathbb{Z}^2 \longleftarrow & X(2) \times \mathbb{Z}^2 \ldots
\end{array}$$

- If $n \geq k$ then $F^n(X)(k) = 1 \times \mathbb{Z}^k \cong \texttt{S(int)}(k)$
- F is a *productive* type constructor!

Productive type constructors

- Plan: capture productive type constructors as contractive morphisms on universe

$$V \xrightarrow{\text{next}} \blacktriangleright V \xrightarrow{F} V$$

- Alternative approach: use that F's action on morphisms is contractive

$$Y^X \xrightarrow{\text{next}} \blacktriangleright(Y^X) \xrightarrow{G_{X,Y}} FY^{FX}$$

- (say F *locally contractive*)

Recursive domain equations

- Recall $F : \mathcal{S} \to \mathcal{S}$ *strong* (enriched) if exists

$$F_{X,Y} : Y^X \to FY^{FX}$$

- Say F locally contractive if each $F_{X,Y}$ contractive:

$$Y^X \xrightarrow{\text{next}} \blacktriangleright(Y^X) \xrightarrow{G_{X,Y}} FY^{FX}$$

  and $G_{X,Y}$ respects composition and identity
- Generalises to mixed variance functors of many variables
- **Theorem:** If $F : \mathcal{S}^{\mathrm{op}} \times \mathcal{S} \to \mathcal{S}$ is locally contractive then there exists X such that $F(X,X) \cong X$. Moreover, X unique up to isomorphism
- Solutions are initial dialgebras

Proof of algebraic compactness (sketch)

- First: existence of solutions to covariant equations
- **Lemma:** Suppose $F : \mathcal{S} \to \mathcal{S}$ is locally contractive and $n \geq k$. Then $F^n(X)(k) \cong F^n(Y)(k)$ for all X, Y.
- Construct solution to F as diagonal of

$$
\begin{array}{ccccccccc}
F(1)(1) & \xleftarrow{F!_1} & F^2(1)(1) & \xleftarrow{F^2(!)_1} & F^3(1)(1) & \xleftarrow{F^3(!)_1} & F^4(1)(1) & \longleftarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
F(1)(2) & \xleftarrow{F!_2} & F^2(1)(2) & \xleftarrow{F^2(!)_2} & F^3(1)(2) & \xleftarrow{F^3(!)_2} & F^4(1)(2) & \longleftarrow & \cdots \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
F(1)(3) & \xleftarrow{F!_3} & F^2(1)(3) & \xleftarrow{F^2(!)_3} & F^3(1)(3) & \xleftarrow{F^3(!)_3} & F^4(1)(3) & \longleftarrow & \cdots \\
\vdots & & \vdots & & \vdots & & \vdots & &
\end{array}
$$

Universal properties for locally contractive functors

- Let F be locally contractive
- All solutions to $F(X) \cong X$ are initial algebras and final coalgebras
- because $h : X \to Y$ is algebra map from $F(X) \cong X$ to g iff

$$\begin{array}{ccc} F(X) & \xleftarrow{\;\cong\;} & X \\ {\scriptstyle F(h)}\downarrow & & \downarrow{\scriptstyle h} \\ F(Y) & \xrightarrow{\;g\;} & Y \end{array}$$

i.e. iff h fixed point of contractive map

$$Y^X \to Y^X$$

Guarded recursive types via universes

Universes in type theory

- Universe type U : Type

$$\frac{\Gamma \vdash A : U}{\Gamma \vdash \mathrm{El}(A) : \mathsf{Type}}$$

- Basic elements $\mathbb{Z} : U$, $\mathrm{El}(\mathbb{Z}) = \mathbb{Z}$

$$\frac{\Gamma \vdash A : U \qquad \Gamma \vdash B : U}{\Gamma \vdash A \times B : U}$$

$$\mathrm{El}(A \times B) = \mathrm{El}(A) \times \mathrm{El}(B)$$

Guarded recursion

- Universe closed under $\blacktriangleright$

$$\frac{\Gamma \vdash A : U}{\Gamma \vdash \triangleright A : U}$$

$$\mathrm{El}(\triangleright(A)) = \blacktriangleright \mathrm{El}(A)$$

Guarded recursion

- Universe closed under $\blacktriangleright$

$$\frac{\Gamma \vdash A : \blacktriangleright U}{\Gamma \vdash \triangleright A : U}$$

$$\mathrm{El}(\triangleright(\mathrm{next}(A))) = \blacktriangleright \mathrm{El}(A)$$

Guarded recursion

- Universe closed under $\blacktriangleright$

$$\frac{\Gamma \vdash A : \blacktriangleright U}{\Gamma \vdash \triangleright A : U}$$

$$\mathrm{El}(\triangleright(\mathrm{next}(A))) = \blacktriangleright \mathrm{El}(A)$$

- Type of streams as fixed point for universe map

$$\mathtt{S(int)} = \mathrm{fix}(\lambda X : \blacktriangleright U . \mathbb{Z} \times \triangleright X)$$

- Then

$$\begin{aligned} \mathrm{El}(\mathtt{S(int)}) &= \mathrm{El}(\mathbb{Z} \times \triangleright(\mathrm{next}(\mathtt{S(int)}))) \\ &= \mathbb{Z} \times \blacktriangleright \mathrm{El}(\mathtt{S(int)}) \end{aligned}$$

Universes in the topos of trees

A universe in $\mathbf{Set}^{\omega^{\mathrm{op}}}$

- Assume given universe U in **Set**
- Construct universe V in $\mathbf{Set}^{\omega^{\mathrm{op}}}$

$$V(1) \xleftarrow{r_1^V} V(2) \xleftarrow{r_2^V} V(3) \xleftarrow{r_3^V} V(4) \longleftarrow \dots$$

- Define $V(1) = U$

$$V(n+1) = \{X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_n} X_{n+1} \mid \forall i. X_i \in U\}$$

$$r_n^V(X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_n} X_{n+1}) = (X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_{n-1}} X_n)$$

- (Construction due to Hofmann and Streicher)

Global elements of the universe

- Maps $1 \to V$

$$\begin{array}{ccccccccc} 1 & \longleftarrow & 1 & \longleftarrow & 1 & \longleftarrow & 1 & \longleftarrow & \dots \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\ V(1) & \longleftarrow & V(2) & \longleftarrow & V(3) & \longleftarrow & V(4) & \longleftarrow & \dots \end{array}$$

- Correspond to objects

$$X(1) \longleftarrow X(2) \longleftarrow X(3) \longleftarrow X(4) \longleftarrow \dots$$

- Such that $X(n) \in U$
- (Generalises to statement about dependent types)

Later operator

- Need $\triangleright : \blacktriangleright V \to V$

$$\begin{array}{ccccccccc}
1 & \longleftarrow & V(1) & \longleftarrow & V(2) & \longleftarrow & V(3) & \longleftarrow & \dots \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
V(1) & \longleftarrow & V(2) & \longleftarrow & V(3) & \longleftarrow & V(4) & \longleftarrow & \dots
\end{array}$$

- Define

$$\underset{1}{\triangleright}(\star) = 1$$

$$\underset{n+1}{\triangleright}(X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_{n-1}} X_n) = (1 \xleftarrow{!} X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_{n-1}} X_n)$$

Later operator

- Need $\triangleright : \blacktriangleright V \to V$

$$\begin{array}{ccccccccc}
1 & \longleftarrow & V(1) & \longleftarrow & V(2) & \longleftarrow & V(3) & \longleftarrow & \dots \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
V(1) & \longleftarrow & V(2) & \longleftarrow & V(3) & \longleftarrow & V(4) & \longleftarrow & \dots
\end{array}$$

- Define

$$\underset{1}{\triangleright}(\star) = 1$$

$$\underset{n+1}{\triangleright}(X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_{n-1}} X_n) = (1 \xleftarrow{!} X_1 \xleftarrow{f_1} X_2 \dots \xleftarrow{f_{n-1}} X_n)$$

- If A corresponds to $\bar{A} : 1 \to V$ then

$$1 \xrightarrow{\quad \bar{A} \quad} V \xrightarrow{\quad \text{next} \quad} \blacktriangleright V \xrightarrow{\quad \triangleright \quad} V$$

corresponds to $\blacktriangleright A$.

Intensional models

Intensional type theory

- Identity types

$$\frac{\Gamma \vdash M, N : A}{\Gamma \vdash \mathrm{Id}_A(M, N) : \mathsf{Type}}$$

- Extensional type theory

$$\frac{\Gamma \vdash - : \mathrm{Id}_A(M, N)}{\Gamma \vdash M \equiv N}$$

Guarded recursion in intensional type theory

- Fixed point property is judgemental equality

$$\frac{\Gamma \vdash f : \blacktriangleright A \to A}{\Gamma \vdash f(\mathrm{next}(\mathrm{fix}\, x.f(x))) \equiv \mathrm{fix}\, x.f(x)}$$

- Uniqueness of fixed points is propositional

$$\frac{\Gamma \vdash f : \blacktriangleright A \to A \quad \Gamma \vdash M : A \quad \Gamma \vdash p : \mathrm{Id}_A(M, f(\mathrm{next}(M)))}{\Gamma \vdash \mathrm{UFP}(p) : \mathrm{Id}_A(M, \mathrm{fix}\, x.f(x))}$$

Intensional models

- **Theorem** (Shulman): If $\mathbb{C}$ is a model of intensional type theory, so is $\mathbb{C}^{\omega^{\mathrm{op}}}$.
- **Theorem**: $\mathbb{C}^{\omega^{\mathrm{op}}}$ models guarded recursion plus UFP

Intensional models

- **Theorem** (Shulman): If $\mathbb{C}$ is a model of intensional type theory, so is $\mathbb{C}^{\omega^{\mathrm{op}}}$.
- **Theorem**: $\mathbb{C}^{\omega^{\mathrm{op}}}$ models guarded recursion plus UFP
- Model construction uses Reedy model structure on $\mathbb{C}^{\omega^{\mathrm{op}}}$
- Closed types are sequences

$$A(1) \xleftarrow{r_1^A} A(2) \xleftarrow{r_2^A} A(3) \xleftarrow{r_3^A} A(4) \longleftarrow \dots$$

  where each r_n^A is a *fibration*
- i.e. r_n^A models

$$x : A(n-1) \vdash A(n)(x) : \mathsf{Type}$$

Univalence

- Voevodsky's univalence axiom

$$\mathrm{Id}_U(A, B) \simeq (A \simeq B)$$

- **Theorem** (Shulman): If U in $\mathbb{C}$ is univalent, so is V in $\mathbb{C}^{\omega^{\mathrm{op}}}$
- Univalence plus UFP implies

$$(F(A) \simeq A) \longleftrightarrow (A \simeq \mathrm{fix}\, X.F(X))$$

Coinductive types via guarded recursive types

Computing with guarded recursive streams

- Recall guarded recursive streams

$$\mathtt{S(int)} \;\mathtt{=}\; \mu\mathtt{X.int} \times \blacktriangleright X$$

$$\mathtt{hd} \colon \mathtt{S(int)} \to \mathtt{int}$$

$$\mathtt{tail} \colon \mathtt{S(int)} \to \blacktriangleright \mathtt{S(int)}$$

$$\mathtt{cons} \colon \mathtt{int} \times \blacktriangleright \mathtt{S(int)} \to \mathtt{S(int)}$$

- Encoding productivity in types!

Computing with guarded recursive streams

- Recall guarded recursive streams

$$
\begin{aligned}
\texttt{S(int)} &= \mu\texttt{X.int} \times \blacktriangleright X \\
\texttt{hd}&\colon \texttt{S(int)} \to \texttt{int} \\
\texttt{tail}&\colon \texttt{S(int)} \to \blacktriangleright \texttt{S(int)} \\
\texttt{cons}&\colon \texttt{int} \times \blacktriangleright \texttt{S(int)} \to \texttt{S(int)}
\end{aligned}
$$

- Encoding productivity in types!
- Computing the second element

$$\texttt{snd} = (\lambda\texttt{(x:xs).next(hd)} \circledast \texttt{xs})\colon \texttt{S(int)} \to \blacktriangleright \texttt{int}$$

- How to get rid of $\blacktriangleright$?

Guarded recursion vs coinduction in model

- Guarded recursive streams

$$\mathbb{Z} \xleftarrow{\pi} \mathbb{Z}^2 \xleftarrow{\pi} \mathbb{Z}^3 \xleftarrow{\pi} \dots$$

- Coinductive streams

$$\mathbb{Z}^{\mathbb{N}} \xleftarrow{id} \mathbb{Z}^{\mathbb{N}} \xleftarrow{id} \mathbb{Z}^{\mathbb{N}} \xleftarrow{id} \dots$$

- All maps `f : S(int)` $\rightarrow$ `S(int)` are *causal*

$$\begin{array}{ccccccc}
\mathbb{Z} & \xleftarrow{\pi} & \mathbb{Z}^2 & \xleftarrow{\pi} & \mathbb{Z}^3 & \xleftarrow{\pi} & \dots \\
\downarrow f_1 & & \downarrow f_2 & & \downarrow f_3 & & \dots \\
\mathbb{Z} & \xleftarrow{\pi} & \mathbb{Z}^2 & \xleftarrow{\pi} & \mathbb{Z}^3 & \xleftarrow{\pi} & \dots
\end{array}$$

Guarded recursion vs coinduction in model

- Guarded recursion useful when constructing streams
- Would like to use coinductive streams when taking them apart
- Observation: Limit of guarded recursive streams is *set* of real streams!

$$\mathbb{Z} \xleftarrow{\pi} \mathbb{Z}^2 \xleftarrow{\pi} \mathbb{Z}^3 \xleftarrow{\pi} \dots$$

- One type theory for both **Set** and $\mathbf{Set}^{\omega^{\mathrm{op}}}$

$$\mathbf{Set} \underset{\lim}{\overset{D}{\rightleftarrows}} \mathbf{Set}^{\omega^{\mathrm{op}}} \quad (D \dashv \lim)$$

Multiple clocks

- Idea due to Atkey and McBride (simply typed setting only)
- I extended it to dependent types using topos of trees model
- Clock variable context $\Delta = \kappa_1, \ldots, \kappa_n$

$$\frac{\Delta; \Gamma \vdash A : \mathrm{Type} \qquad \kappa \in \Delta}{\Delta; \Gamma \vdash \overset{\kappa}{\blacktriangleright} A : \mathrm{Type}}$$

$$\mathtt{fix}^{\kappa} : (\overset{\kappa}{\blacktriangleright} X \to X) \to X$$

- etc

Universal quantification over clocks

$$\frac{\Delta, \kappa; \Gamma \vdash A : \mathsf{Type} \qquad \kappa \notin \mathsf{fc}(\Gamma)}{\Delta; \Gamma \vdash \forall \kappa . A : \mathsf{Type}}$$

$$\frac{\Delta, \kappa; \Gamma \vdash t : A \qquad \kappa \notin \mathsf{fc}(\Gamma)}{\Delta; \Gamma \vdash \Lambda \kappa . t : \forall \kappa . A}$$

$$\frac{\kappa, \kappa' \notin \mathsf{fc}(\Gamma, \forall \kappa . A) \qquad \Delta, \kappa; \Gamma \vdash A : \mathsf{Type} \qquad \Delta, \kappa'; \Gamma, \Gamma' \vdash t : \forall \kappa . A}{\Delta, \kappa'; \Gamma, \Gamma' \vdash t[\kappa'] : A[\kappa'/\kappa]}$$

- Clock quantification is right adjoint to clock weakening

Type isomorphisms

$$\begin{aligned}
\forall\kappa.A &\cong A && (\kappa \notin \mathrm{fc}(A)) \\
\forall\kappa.\forall\kappa'.A &\cong \forall\kappa'.\forall\kappa.A \\
\forall\kappa.(A \times B) &\cong (\forall\kappa.A) \times (\forall\kappa.B) \\
(\forall\kappa.A) + (\forall\kappa.B) &\cong \forall\kappa.(A + B) \\
\sum x : A.\forall\kappa.B &\cong \forall\kappa.\sum x : A.B && (\kappa \notin \mathrm{fc}(A)) \\
\forall\kappa.\prod x : A.B &\cong \prod x : A.\forall\kappa.B && (\kappa \notin \mathrm{fc}(A)) \\
\overset{\kappa'}{\blacktriangleright}\forall\kappa.A &\cong \forall\kappa.\overset{\kappa'}{\blacktriangleright} A && (\kappa \neq \kappa') \\
\forall\kappa.A &\cong \forall\kappa.\overset{\kappa}{\blacktriangleright} A
\end{aligned}$$

- Can all be proved sound wrt model

Coinduction via guarded recursion

$$
\begin{aligned}
\mathtt{S}^{\kappa}(\mathtt{int}) &= \mathtt{int} \times \overset{\kappa}{\blacktriangleright} \mathtt{S}^{\kappa}(\mathtt{int}) \\
\mathtt{S}(\mathtt{int}) &= \forall \kappa . \mathtt{S}^{\kappa}(\mathtt{int})
\end{aligned}
$$

- Type isomorphism provable using isomorphisms from last slide:

$$
\begin{aligned}
\mathtt{S}(\mathtt{int}) &= \forall \kappa . (\mathtt{int} \times \overset{\kappa}{\blacktriangleright} \mathtt{S}^{\kappa}(\mathtt{int})) \\
&\cong (\forall \kappa . \mathtt{int}) \times (\forall \kappa . \overset{\kappa}{\blacktriangleright} \mathtt{S}^{\kappa}(\mathtt{int})) \\
&\cong \mathtt{int} \times (\forall \kappa . \mathtt{S}^{\kappa}(\mathtt{int})) \\
&= \mathtt{int} \times \mathtt{S}(\mathtt{int})
\end{aligned}
$$

- Can prove this is a final coalgebra

Coinduction via guarded recursion

$$\mathtt{S}^{\kappa}(\mathtt{int}) = \mathtt{int} \times \overset{\kappa}{\blacktriangleright} \mathtt{S}^{\kappa}(\mathtt{int})$$
$$\mathtt{S}(\mathtt{int}) = \forall\kappa.\mathtt{S}^{\kappa}(\mathtt{int})$$
$$\mathtt{S}(\mathtt{int}) \cong \mathtt{int} \times \mathtt{S}(\mathtt{int})$$

- Example: define function

$$\mathtt{odd} \colon \mathtt{S}(\mathtt{int}) \to \mathtt{S}(\mathtt{int})$$

selecting the elements at odd indices of input stream

Example

- Need indexed fixed point operator

$$\mathtt{pfix}^{\kappa} \colon ((\mathtt{A} \to \overset{\kappa}{\blacktriangleright} \mathtt{B}) \to (\mathtt{A} \to \mathtt{B})) \to (\mathtt{A} \to \mathtt{B})$$

- Defined as

$$\mathtt{pfix}^{\kappa}(\mathtt{f}) = \mathtt{fix}^{\kappa}(\lambda \mathtt{g} \colon \overset{\kappa}{\blacktriangleright}(\mathtt{A} \to \mathtt{B}).\mathtt{f}(\lambda \mathtt{x} \colon \mathtt{A}.\mathtt{g} \circledast^{\kappa}(\mathtt{next}^{\kappa}(\mathtt{x}))))$$

- Satisfies equation

$$\mathtt{f}(\mathtt{next}^{\kappa} \circ \mathtt{pfix}^{\kappa}(\mathtt{f})) = \mathtt{pfix}^{\kappa}(\mathtt{f})$$

Example

- Define

$$\mathtt{oddrec}^{\kappa} \colon (\mathtt{S(int)} \to \overset{\kappa}{\blacktriangleright} \mathtt{S}^{\kappa}\mathtt{(int)}) \to (\mathtt{S(int)} \to \mathtt{S}^{\kappa}\mathtt{(int)})$$

$$\mathtt{oddrec}^{\kappa}\ \mathtt{g\ x::y::xs\ =\ x::}^{\kappa}\mathtt{g(xs)}$$

$$\mathtt{odd}^{\kappa} \colon \mathtt{S(int)} \to \mathtt{S}^{\kappa}\mathtt{(int)}$$

$$\mathtt{odd}^{\kappa}\ \mathtt{=\ pfix}^{\kappa}\mathtt{(oddrec}^{\kappa}\mathtt{)}$$

$$\mathtt{odd} \colon \mathtt{S(int)} \to \mathtt{S(int)}$$

$$\mathtt{odd\ xs\ =\ } \Lambda\kappa.\mathtt{odd}^{\kappa}\mathtt{(xs)}$$

Coinduction via guarded recursion, general theorem

Theorem. if $\mathtt{F}$ functorial and

$$\mathtt{F}(\forall\kappa.\mathtt{X}) \cong \forall\kappa.\mathtt{F}(\mathtt{X}) \qquad \kappa \notin \Delta$$

Then

$$\nu\mathtt{X}.\mathtt{F}(\mathtt{X}) =_{\text{def}} \forall\kappa.\mathrm{Fix}^{\kappa}\mathtt{X}.\mathtt{F}(\overset{\kappa}{\blacktriangleright}\mathtt{X})$$

carries a final coalgebra structure for $\mathtt{F}$.

Semantics

Multidimensional topos of trees

$$
\begin{aligned}
\mathrm{GR}[\Delta] &=_{\mathrm{def}} \mathbf{Set}^{(\omega^{\Delta})^{\mathrm{op}}} \\
\mathrm{GR}[-] &\cong \mathbf{Set} \\
\mathrm{GR}[\kappa] &\cong \mathbf{Set}^{\omega^{\mathrm{op}}}
\end{aligned}
$$

- Object of $\mathrm{GR}[\kappa, \kappa']$

$$
\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
X_{1,2} & \longleftarrow & X_{2,2} & \longleftarrow & X_{3,2} & & \dots \\
\downarrow & & \downarrow & & \downarrow & & \\
X_{1,1} & \longleftarrow & X_{2,1} & \longleftarrow & X_{3,1} & & \dots
\end{array}
$$

Interpretation of clock quantification

$$\mathrm{GR}[\Delta] \underset{\lim}{\overset{D}{\rightleftarrows}} \mathrm{GR}[\Delta, \kappa] \quad (D \dashv \lim)$$

- Interpretation of clock quantification

$$[\![\Delta; - \vdash \forall\kappa.A : \mathsf{Type}]\!] = \lim([\![\Delta, \kappa; - \vdash A : \mathsf{Type}]\!])$$

- Crucial invariant

$$[\![\Delta, \kappa; - \vdash A : \mathsf{Type}]\!] \cong D([\![\Delta; - \vdash A : \mathsf{Type}]\!]) \qquad \kappa \notin \mathsf{fc}(A)$$

A category subsuming all $\mathrm{GR}[\Delta]$

- Let CV countable set of all clock variables

$$\mathrm{GR}[\Delta] \simeq \overline{\mathrm{GR}[\Delta]} \hookrightarrow \mathrm{GR}[\mathrm{CV}] =_{\mathrm{def}} \mathbf{Set}^{(\omega^{\mathrm{CV}})^{\mathrm{op}}}$$

- Interpret types, contexts and terms in $\mathrm{GR}[\mathrm{CV}]$
- Prove that $[\![\Delta; - \vdash A : \mathsf{Type}]\!]$ in $\overline{\mathrm{GR}[\Delta]}$
- Crucial invariant becomes equality

$$[\![\Delta, \kappa; - \vdash A : \mathsf{Type}]\!] = [\![\Delta; - \vdash A : \mathsf{Type}]\!] \qquad \kappa \notin \mathsf{fc}(A)$$

Universes

- Assume universe U in **Set**
- Can construct universe V_Δ in $\mathrm{GR}[\Delta]$ (Hofmann-Streicher)
- But

$$\llbracket \Delta; - \vdash U : \mathsf{Type} \rrbracket = V_\Delta$$
$$\llbracket \Delta, \kappa; - \vdash U : \mathsf{Type} \rrbracket \not\cong D\llbracket \Delta; - \vdash U : \mathsf{Type} \rrbracket$$

- Instead

$$\frac{\Delta' \subseteq \Delta}{\Delta; \Gamma \vdash \mathrm{U}_{\Delta'} : \mathsf{Type}}$$

- Semantics

$$\llbracket \Delta; \Gamma \vdash \mathrm{U}_{\Delta'} : \mathsf{Type} \rrbracket = D_{\Delta', \Delta}(V_{\Delta'})$$

Conclusions

- Guarded recursive types useful for
  - modelling higher-order store
  - computing with streams (encoding productivity in types)
- Model: topos of trees
- Intensional variant shows g.r. consistent with univalence
- Coinductive types can be encoded using guarded recursive types

Thanks!