Total (Co)Programming with Guarded Recursion

Andrea Vezzosi

Department of Computer Science and Engineering
Chalmers University of Technology, Gothenburg, Sweden

Types for Proofs and Programs Annual Meeting 2015
Tallinn, Estonia
18 May 2015

Guarded Recursion

- Guarded coinductive types
- Coinductive types
- Guarded fixed point operator as only source of recursion
- Recursive types as fixed points on the universe

What about Induction?

Main Combinators

- $\triangleright\, A$, "later A", modality as an applicative functor:

$$next : A \to \triangleright\, A$$
$$\circledast \quad : \triangleright\,(A \to B) \to \triangleright\, A \to \triangleright\, B$$

Guarded fixpoint combinator:

$$fix : (\triangleright\, A \to A) \to A$$
$$fix\ f = f\ (next\ (fix\ f))$$

Corecursion Example

$$gStr\ A \cong A \times \triangleright\ gStr\ A$$

$$\begin{aligned} ghead &: gStr\ A \to A \\ ghead &= fst \end{aligned}$$

$$\begin{aligned} gtail &: gStr\ A \to \triangleright\ gStr\ A \\ gtail &= snd \end{aligned}$$

$$\begin{aligned} map &: (A \to B) \to gStr\ A \to gStr\ B \\ map\ f &= fix\ (\lambda\ map'\ .\ \lambda\ xs\ .\ ghead\ xs, map' \circledast gtail\ xs) \end{aligned}$$

Recursion Example?

$$gList\ A \cong \top + A \times gList\ A$$

$$\begin{aligned}
&all : (A \to Bool) \to gList\ A \to Bool \\
&all\ p = fix\ (\lambda\,(all' : \triangleright\,(gList\ A \to Bool))\,.\,\lambda\,xs\,. \\
&\qquad case\ xs\ of \\
&\qquad\quad [\,] \qquad\quad \to True \\
&\qquad\quad (x :: xs) \to p\ x \wedge\ ?
\end{aligned}$$

We need a way to call all' with xs as argument and obtain $Bool$.

Recursion Example, take 2, with diamonds

$$gList\ A \cong \top + A \times \Diamond\ gList\ A$$

$$\begin{aligned} &extract : \Diamond\ Bool\ \to\ Bool \\ &\star\ \ \ \ \ \ : \triangleright\,(A\ \to\ B)\ \to\ \Diamond\ A\ \to\ \Diamond\ B \end{aligned}$$

$$\begin{aligned} &all : (A\ \to\ Bool)\ \to\ gList\ A\ \to\ Bool \\ &all\ p\ =\ fix\ (\lambda\,(all' : \triangleright\,(gList\ A\ \to\ Bool))\,.\,\lambda\,xs\,. \\ &\qquad\qquad case\ xs\ of \\ &\qquad\qquad\quad [\,]\ \ \ \ \ \ \ \ \to\ True \\ &\qquad\qquad\quad (x :: xs)\ \to\ p\ x \wedge extract\ (all'\ \star\ xs) \end{aligned}$$

Problem: we lose next

For $\lozenge\ A$ we cannot have $next$, e.g.:

$$next : \lozenge\ \top\ \to\ \triangleright\ (\lozenge\ \top)$$

$\lozenge\ \top\ \to\ \triangleright\ (\lozenge\ \top)$ means
"if there is time left now, there will be time left later too"

Semantics

The standard model for Guarded Recursion is the topos of trees i.e. functors $\omega^{op} \to \mathsf{Set}$

$$
\begin{aligned}
&A : \mathbb{N} \to \mathit{Set} \\
&A\ (n \leqslant m) : A\ m \to A\ n
\end{aligned}
$$

$$
\begin{aligned}
(\triangleright\ A)\ 0 &= \top \\
(\triangleright\ A)\ (\mathit{suc}\ n) &= A\ n \\
\mathit{next}_0 &= ! \\
\mathit{next}_{\mathit{suc}\ n} &= A\ (n \leqslant \mathit{suc}\ n)
\end{aligned}
$$

next uses the functoriality of A

Alternative Semantics: Relators

$$A : \mathbb{N} \to Set$$
$$A\ (n \leqslant m) : A\ m \to A\ n \to Set$$
$$A\ (n \leqslant n) \cong\ =_{A\,n}$$

Any functor $A : \omega^{op} \to Set$ is also a relator:

$$A\ (n \leqslant m)\ a_n\ a_m\ =\ a_n\ =_{A\,n}\ A\ (n \leqslant m)\ a_m$$

ala Sized Types

$$\triangleright, \Diamond : (Time \rightarrow Set) \rightarrow (Time \rightarrow Set)$$
$$\triangleright\ A\ i\ =\ \forall\ j < i.\ A\ j$$
$$\Diamond\ A\ i\ =\ \exists\ j < i.\ A\ j$$

$$\star : \forall\ i.\ (\forall\ j < i.\ A\ j\ \rightarrow\ B\ j)\ \rightarrow\ (\exists\ j < i.\ A\ j)\ \rightarrow\ \exists\ j < i.\ B\ j$$
$$f\ \star\ (j, a)\ =\ (j, f\ j\ a)$$

ala Sized Types (contd.)

$$\mathit{fix} : (\forall\ i.\ (\forall\ j < i.\ A\ j)\ \to\ A\ i)\ \to\ \forall\ i.\ A\ i$$

$$\begin{aligned}
&\mathit{unfold} : (\forall\ i\ .\ S\ i\ \to\ \top + (A \times \exists\ j < i.\ S\ j)) \\
&\qquad\qquad \to\ \forall\ i\ .\ S\ i\ \to\ \mathit{List}\ A \\
&\mathit{unfold}\ f\ =\ \mathit{fix}\ \lambda\ i\ \mathit{unfold}'\ s\ .\ \mathit{case}\ f\ i\ s\ \mathit{of} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathit{inl}\ _\ \to\ [\,] \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathit{inr}\ (a, (j, s'))\ \to\ a :: \mathit{unfold}'\ j\ s'
\end{aligned}$$

Recursive Types through fixed points

$$\hat{\triangleright} : \triangleright\ U \ \rightarrow \ U$$
$$gStr\ A \ = \ \mathit{fix}\ \lambda\ X.\ A \times \hat{\triangleright}\ X$$

$$gStr\ A \ = \ \mathit{fix}\ \lambda\ i\ (X : \forall\ j < i.\ U).\ A \times \forall\ j < i.\ X\ j$$

Coinductive Types with $\triangleright$

$$gStr^{\kappa}\ A \cong A \times \triangleright^{\kappa}\ gStr^{\kappa}\ A$$
$$Str\ A \cong \forall\ \kappa\,.\ gStr^{\kappa}\ A$$

$$force : (\forall\ \kappa\,.\ \triangleright^{\kappa}\ A) \cong (\forall\ \kappa\,.\ A)$$
$$tail : Str\ A\ \rightarrow\ Str\ A$$
$$tail\ xs\ =\ force\ (\lambda\ \kappa\,.\ gtail\ (xs\ \kappa))$$

Coinductive Types with $\forall\, j < i$

$$gStr\ A\ i \cong A \times \forall\, j < i.\ gStr\ A\ j$$
$$Str\ A \cong \forall\, i.\ gStr\ A\ i$$

$$force_{\triangleright}\ :\ (\forall\, i.\, \forall\, j < i.\ A\ j)\ \rightarrow\ \forall\, i.\ A\ i$$
$$force_{\triangleright}\ f\ \ i\ \ \ =\ f\ (suc\ i)\ i$$
$$guard_{\triangleright} : (\forall\, i.\ A\ i)\ \rightarrow\ \forall\, i.\, \forall\, j < i.\ A\ j$$
$$guard_{\triangleright}\ f\ i\ j\ =\ f\ j$$

$$guard_{\triangleright}\ (force_{\triangleright}\ f)\ i\ j\ =\ f\ (suc\ j)\ j$$

Inductive Types with $\exists\, j < i$

$$gNat\ i \cong \top + \exists\, j < i.\ A\ j$$
$$Nat \cong \exists\, i.\ gNat\ i$$

$$force_\Diamond\ : (\exists\, i.\ \exists\, j < i.\ A\ j)\ \to\ \exists\, i.\ A\ i$$
$$force_\Diamond\ (i, j, a)\ =\ (j, a)$$
$$guard_\Diamond : (\forall\, i.\ A\ i)\ \to\ \forall\, i.\ \forall\, j < i.\ A\ j$$
$$guard_\Diamond\ (j, a)\ \ =\ (suc\ j, j, a)$$

$$guard_\Diamond\ (force_\Diamond\ (i, j, a))\ =\ suc\ j, j, a$$

$\exists\ i$ as a weak existential

$$gNat\ i \cong \top + \exists\ j < i.\ A\ j$$
$$Nat \cong \exists\ i.\ gNat\ i$$

Want all "zeros" to be equal:

$$(i, inl\ tt)\ =\ (j, inl\ tt)$$

We cannot project times out:

$$fst : (\exists\ i.\ A\ i)\ \rightarrow\ Time$$
$$fst\ (i, a)\ =\ i$$
$$\begin{aligned} i\ &=\ fst\ (i, inl\ tt) \\ &=\ fst\ (j, inl\ tt) \\ &=\ j \end{aligned}$$

$\exists\ i$ as a weak existential

$$\frac{P : (\exists\ i.\ A\ i)\ \to\ U \qquad f : (\forall\ i.\ (a : A\ i)\ \to\ P\ (i, a))}{uncurry\ f : (x : \exists\ i\ .\ A\ i)\ \to\ P\ x}$$

where U is a type theoretic universe such that $Time \notin U$

Summary

- Ordered type $Time : Type$ which supports well-founded induction
- A universe $U : Type$ such that $Time \notin U$
- Parametric time quantifiers $\forall\ i.\ A\ i$ and $\exists\ i.\ A\ i$

Reflexive Graph Model of Martin Löf Type Theory

$$
\begin{aligned}
\Gamma_O &: Set \\
\Gamma_R &: \Gamma_O \to \Gamma_O \to Set \\
\Gamma_{refl} &: (\gamma_O : \Gamma_O) \to \Gamma_R\, \gamma_O\, \gamma_O
\end{aligned}
$$

$Time$

$$
\begin{aligned}
Time_O &= \mathbb{N} \\
Time_R\ i\ j &= \top
\end{aligned}
$$

Any two time values are related.

Types depending on $Time$

$$i : Time \vdash A : Type$$

$$A_O : \mathbb{N} \to Set$$
$$A_R : (n\ m : \mathbb{N}) \to A\ n \to A\ m \to Set$$
$$A_{refl} : (n : \mathbb{N}) \to (a : A\ n) \to A_R\ n\ n\ a\ a$$

$$A_R\ n\ n =?\ =_{A_O}$$

Universe of Small Discrete Reflexive Graphs

$$
\begin{aligned}
U_O &= \{(A_O, A_R) \mid A_O \ small\ set, A_R \cong eqAo\} \\
U_R\ A\ B &= \{Rel \mid Rel\ small\ proof\ irrelevant\ relation \\
&\qquad between\ A_O\ and\ B_O\} \\
U_{refl}\ (A_O, A_R) &= A_R
\end{aligned}
$$

$$
\frac{\Gamma \vdash A : U}{\Gamma \vdash El\ A : Type}
$$

$$
(El\ A)_R\ (\Gamma_{refl}\ \gamma) \cong =_{El A_O}
$$

$Time$ dependency for discrete reflexive graphs

Given A such that $i \notin fv\ A$

$$i : Time \vdash t : El\ A$$

$$t_O : \mathbb{N} \to (El\ A)_O$$
$$t_R : (n\ m : \mathbb{N}) \to t_O\ n =_{El A_O} t_O\ m$$

$$[\![\forall\, i\,.\, \forall\, j < i.\ El\ (A\ j)]\!] \cong [\![\forall\, i\,.\ El\ (A\ i)]\!]$$

Discretization

Given any small reflxive graph A
we can form its free discrete reflexive graph $\int A : U$

$$(El\ (\int\ A))_O\ =\ A_O\ /\ \textit{symmetric transitive closure of}\ A_R$$
$$(El\ (\int\ A))_R \cong\ =_{(\int A)_O}$$

Discretization, Universal property

$$(El\ (\int\ A)\ \to\ El\ B) \cong (A\ \to\ El\ B)$$

$$\frac{P : El\ (\int\ A)\ \to\ U \qquad f : (a : A)\ \to\ P\ (\int\ a)}{elim\ f : (x : El\ (\int\ A))\ \to\ El\ (P\ x)}$$

$$\exists\ i\ .\ A\ i\ =\ \int\ (\Sigma\ (i : Time)\ .\ A\ i)$$

Future work

How to internalize the parametricity properties of $\forall\ i$ and $\exists\ i$?

- Very interested in the talks about parametricity in the following days!
- Cohesive Homotopy Type Theory has something like discretization $\int$

How to preserve strong normalization?

$$\begin{aligned} \mathit{fix}\ f\ i &= f\ i\ (\mathit{fix}\ f) \\ &= f\ i\ (\lambda\, j.\, \mathit{fix}\ f\ j) \\ &= f\ i\ (\lambda\, j.\, f\ j\ (\mathit{fix}\ f)) \\ &= \ldots \end{aligned}$$