Guarded Cubical Type Theory

Path Equality for Guarded Recursion

Lars Birkedal[1], Aleš Bizjak[1], Ranald Clouston[1],
Hans Bugge Grathwohl[1], Bas Spitters[1], **Andrea Vezzosi[2]**

[1] Aarhus University, Denmark
[2] Chalmers University of Technology, Sweden

CSL 2016
Marseille
30 Aug 2016

In April at FoSSaCS...

We presented: **Guarded Dependent Type Theory**
an extensional type theory with *guarded recursive types.*

Guarded recursion can, e.g., be used for

- programming and reasoning with coinductive types,

- building models of program logics
  (solving recursive domain equations synthetically).

Guarded Recursive Types: Vocabulary

$\triangleright A$	the type of terms that will be *A* at the next tick of the clock
next *t*	the term *t* saved for the next tick of the clock
fix *x*.*t*	guarded fixed point of *t*
$t \circledast u$	function application at the next tick of the clock

Example: Streams

$$\mathsf{Stream} = \mathsf{Nat} \times \triangleright \mathsf{Stream}$$

$$\mathsf{head} = \pi_1$$

$$\mathsf{tail} = \pi_2$$

$$\mathsf{zeros} = \mathsf{fix}\ s.\ (0, s)$$

$$\mathsf{map}\ f = \mathsf{fix}\ m.\ \lambda s.\ (f\ (\mathsf{head}\ s), m \circledast (\mathsf{tail}\ s))$$

Example: Y Combinator

$\mathrm{Rec}_A = \triangleright \mathrm{Rec}_A \to A$

$Y : (\triangleright A \to A) \to A$

$Y = \lambda f.\ \Delta\ (\mathrm{next}\ \Delta)$

where

$\Delta : \triangleright \mathrm{Rec}_A \to A$

$\Delta = \lambda x.\ f\ (x \circledast (\mathrm{next}\ x))$

Going Dependent

$$\frac{\Gamma \vdash t : \triangleright(A \to B) \quad \Gamma \vdash u : \triangleright A}{\Gamma \vdash t \circledast u : \triangleright B}$$

$$\frac{\Gamma \vdash t : \triangleright((x : A) \to B) \quad \Gamma \vdash u : \triangleright A}{\Gamma \vdash t \circledast u : \triangleright ?}$$

Going Dependent

$$\frac{\Gamma \vdash t : \triangleright(A \to B) \quad \Gamma \vdash u : \triangleright A}{\Gamma \vdash t \circledast u : \triangleright B}$$

$$\frac{\Gamma \vdash t : \triangleright((x : A) \to B) \quad \Gamma \vdash u : \triangleright A}{\Gamma \vdash t \circledast u : \triangleright[x \leftarrow u].\, B}$$

Delayed substitutions

Going Dependent

$$\frac{\Gamma \vdash t : \triangleright(A \to B) \quad \Gamma \vdash u : \triangleright A}{\Gamma \vdash t \circledast u : \triangleright B}$$

$$\frac{\Gamma \vdash t : \triangleright((x : A) \to B) \quad \Gamma \vdash u : \triangleright A}{\Gamma \vdash \mathsf{next}\ [t' \leftarrow t,\ u' \leftarrow u].\ t'\ u' \ : \triangleright[x \leftarrow u].\ B}$$

Delayed substitutions

Löb Induction

Guarded fixed points on identity type:

$$\frac{\Gamma, x : \triangleright(t = u) \vdash p : t = u}{\Gamma \vdash \mathrm{fix}\ x.\ p : t = u}$$

This is powerful in combination with the term:

$$\mathrm{com} : \triangleright(t = u) \to \mathrm{next}\ t = \mathrm{next}\ u,$$

to prove e.g. bisimilar streams equal:

$$\vdash \mathrm{fix}\ x.\ ..\mathrm{com}\ x..\ : \mathrm{map}\ \mathrm{id}\ \mathrm{zeros} = \mathrm{zeros}$$

Canonicity Loss

com : $\triangleright(t = u) \to \text{next } t = \text{next } u$

if “=” is intensional Martin-Löf identity type,
“com” comes at the cost of *canonicity:*

$$\vdash \text{fix } x.\ ..\text{com } x..\ :\ \text{map id zeros} = \text{zeros}$$

but

$$\vdash \text{refl} \not: \text{map id zeros} = \text{zeros}$$

Identity Types

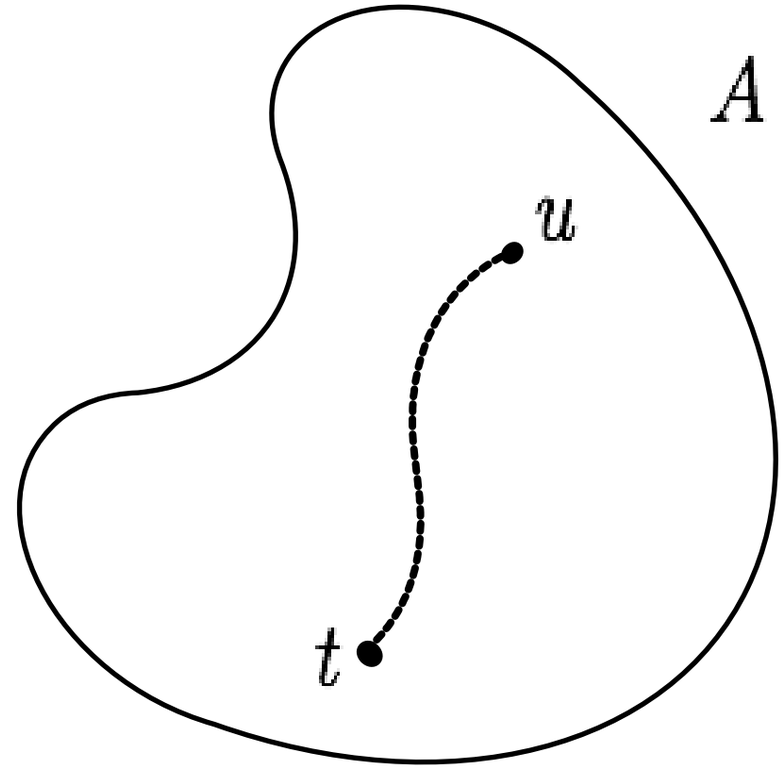
We want:

- a dependent type theory with
- guarded recursive types,
- canonicity,
- decidable type-checking.

So we need another notion of identity than the usual intensional or extensional identity types.

Cubical Type Theory

Dependent type theory influenced by homotopy type theory: Types are **spaces**, identities are **paths**.



$\mathrm{Id}_A\ t\ u$

(Cohen, Coquand, Huber, Mörtberg 2016)

Cubical Type Theory

We have a special *interval object* $\mathbb{I}$, and equalities are like functions from $\mathbb{I}$ into a type:

$$\frac{\Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \langle i \rangle\, t : \mathsf{Id}_A\ t[0/i]\ \ t[1/i]}$$

$$\frac{\Gamma \vdash p : \mathsf{Id}_A\, t\ u \qquad \Gamma \vdash r : \mathbb{I}}{\Gamma \vdash p\, r : A} \qquad \begin{aligned} p\,0 &= t \\ p\,1 &= u \end{aligned}$$

Cubical Type Theory

Theorem:

- Consistency & soundness (w.r.t. *cubical sets model*)

Conjectures:

- Canonicity (https://arxiv.org/abs/1607.04156)
- Decidable type-checking

Implementation:

https://github.com/mortberg/cubicaltt

Guarded Cubical Type Theory

We propose **Guarded Cubical Type Theory**, a combination of GDTT and CTT.

Paths and guarded recursive types play well together.

Compare:

$$\mathsf{funext} : ((a : A) \to \mathsf{Id}\ B\ (f\ a)\ (g\ a)) \to \mathsf{Id}\ (A \to B)\ f\ g$$
$$\mathsf{funext} = \lambda p.\ \langle i \rangle\ \lambda a.\ p\ a\ i$$

$$\mathsf{com} : \triangleright(\mathsf{Id}\ A\ t\ u) \to \mathsf{Id}\ (\triangleright A)\ (\mathsf{next}\ t)\ (\mathsf{next}\ u)$$
$$\mathsf{com} = \lambda p.\ \langle i \rangle\ \mathsf{next}\ [x \leftarrow p].\ x\ i$$

Unfolding Fixed Points

Obvious source of non-termination:

$$\text{fix x.}\,t = t[\text{next(fix x.}\,t)/\text{x}]$$

We have a *path equality* instead of definitional equality

$$\frac{\Gamma \vdash r : \mathbb{I} \qquad \Gamma, x : \triangleright A \vdash t : A}{\Gamma \vdash \text{dfix}^r\ x.t : \triangleright A}$$

"delayed fixed point"

along with

$$\text{dfix}^1\ x.t = \text{next}\ (t[\text{dfix}^0\ x.t/x])$$

We write: fix x.t = t[dfix^0 x.t/x]

Guarded Cubical

Prototype type-checker at
https://github.com/hansbugge/cubicaltt/tree/gcubical
with examples.

Andrea Vezzosi, CSL 2016

Semantics

Lemma. $\mathcal{E}$ presheaf topos with non-trivial internal De Morgan algebra with the disjunction propery ⇒ fibrant types of $\mathcal{E}$ models CTT (*without* glueing and universe)

Lemma. $\mathbb{C}$ category with initial object ⇒ fibrant types of **PSh(**$\square \times \mathbb{C}$) models CTT (*with* glueing and universe). ($\square$: category of cubes).

Theorem. The fibrant types of **PSh(**$\square \times \omega$) models GCTT. Especially: $\triangleright A$ is fibrant.

Conclusion and Future Work

We have

- Consistent type theory – GCTT
- Model – PSh(□ × ω)
- Implementation and examples

Future work

- Algorithmic equality (decidable type checking)
- Canonicity
- Multiple clocks and clock quantification (real coinduction)