Partiality as an Effect, Made Rigourous

Tarmo Uustalu and Niccolò Veltri

Institute of Cybernetics at Tallinn University of Technology

Joint Estonian-Latvian Theory Days in Lilaste, 16 October 2016

Introduction

- Topic of this talk: modeling possibly non-terminating computations in type theory (Agda).
- Agda is a dependently typed functional programming language.
- E.g. the type of vectors of a given length:

$$\frac{}{[] : \mathsf{Vec}\, X\, 0} \qquad \frac{x : X \quad xs : \mathsf{Vec}\, X\, n}{x :: xs : \mathsf{Vec}\, X\, (n+1)}$$

- Agda is a foundational language for the development of constructive mathematics (based on Martin-Löf type theory).

$$+\mathsf{comm} : \forall n, m : \mathbb{N}.\ n + m \equiv m + n$$

- types $\leftrightsquigarrow$ propositions, terms $\leftrightsquigarrow$ proofs.

Introduction

- Agda is a total language, non-terminating programs are not allowed.
- E.g. Kleene's minimization operator.

$$\begin{aligned}&\text{minim} : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}\\ &\text{minim}\ f = \text{“the smallest } n \text{ such that } f\ n \equiv 0\text{”}\end{aligned}$$

- We can treat possible non-termination as an effect and use a monad to deal with it.
  $\Rightarrow$ Capretta's delay monad
- We are interested in termination of computations and not the exact computation time.
  $\Rightarrow$ Weak bisimilarity
- We quotient the delay monad by weak bisimilarity, we obtain another monad (???) $D_{\approx}$.

Introduction

- What does it mean that $D_\approx$ is a "monad for partiality", or a "monad for non-termination"?
- In order to make these statements precise, we give a precise category theoretical characterization of $D_\approx$.
- Building on top of work by Cockett, Lack and Guo, we introduce a new class of monads for partiality.
  $\Rightarrow$ ω-join classifying monads
- We proved in Agda that $D_\approx$ is the initial join classifying monad.
- In this sense, the monad $D_\approx$ provides a canonical solution for introducing non-termination in type theory.

Monads

- A monad is a map $T : Set \to Set$ together with operations:
  - a unit $\eta_X : X \to TX$;
  - a substitution (bind) operation

$$\frac{f : X \to TY}{f^* : TX \to TY}$$

  subjects to conditions

$$\begin{array}{rcl} \eta_X^* & \equiv & \mathrm{id}_{TX} \\ f^* \circ \eta_X & \equiv & f \\ g^* \circ f^* & \equiv & (g^* \circ f)^* \end{array}$$

- Intuition: think at $TX = \mathrm{Term}_\Sigma X$ terms over a signature Σ.

Monads and effects

- Exception: $\mathrm{Exc}\, X = X + E$.
- (Maybe: $\mathrm{Maybe}\, X = X + 1$.)
- Non-determinism: $\mathrm{NonDet}\, X = \mathrm{List}\, X$.
- State: $\mathrm{State}\, X = (S \to S \times X)$.
- ...
- Effectful computations: $f : X \to TY$.
- Pure functions: $f : X \to Y$.
- $\eta_X : X \to TX$ identity on X thought of as trivially effectful.
- Composition of effectful computations $f : X \to TY$ and $g : Y \to TZ$:
$$g \diamond f = g^* \circ f$$
- What about non-termination?

Delay monad

- For a given type X, each element of $\mathsf{D}X$ is a possibly infinite computation that returns a value of X, if it terminates. We define $\mathsf{D}X$ as a coinductive type by the rules

$$\frac{}{\mathsf{now}\, x : \mathsf{D}X} \qquad \frac{c : \mathsf{D}X}{\mathsf{later}\, c : \mathsf{D}X}$$

- Examples: $\mathsf{now}\, x$, $\mathsf{later}^n\,(\mathsf{now}\, x)$, $\mathsf{never} = \mathsf{later}\,\mathsf{never}$.
- The delay datatype is a monad: unit $\eta : X \to \mathsf{D}\, X$ is now; bind operation:

$$\begin{array}{rcl} f^*\,(\mathsf{later}^n\,(\mathsf{now}\, x)) & = & \mathsf{later}^n\,(f\, x) \\ f^*\,\mathsf{never} & = & \mathsf{never} \end{array}$$

Equality of computations: weak bisimilarity

- Weak bisimilarity is defined in terms of convergence. This binary relation between $\mathsf{D}X$ and X relates a terminating computation to its value and is inductively defined by the rules

$$\frac{}{\mathsf{now}\, x \downarrow x} \qquad \frac{c \downarrow x}{\mathsf{later}\, c \downarrow x}$$

- Two computations are weakly bisimilar if they differ by a finite number of application of the constructor later, i.e., they either converge to the same value or diverge. Weak bisimilarity is defined coinductively by the rules

$$\frac{c_1 \downarrow x \quad c_2 \downarrow x}{c_1 \approx c_2} \qquad \frac{c_1 \approx c_2}{\mathsf{later}\, c_1 \approx \mathsf{later}\, c_2}$$

- Examples: $\mathsf{later}^n\,(\mathsf{now}\, x) \approx \mathsf{later}^k\,(\mathsf{now}\, x)$, $\mathsf{never} \approx \mathsf{never}$.

Quotiented delay monad

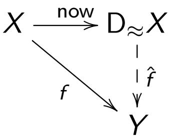
- We quotient the delay monad by weak bisimilarity.

$$\mathsf{D}_{\approx}\, X = \mathsf{D}\, X/{\approx}$$

- Agda does not have quotient types. We extend the type theory with Hofmann's quotient types.
- Previous work: is $\mathsf{D}_{\approx}$ a monad? Yes, but we have to postulate additional principles, such as the axiom of countable choice.

$\mathrm{D}_\approx$ delivers free ωcppos

- A ω-complete pointed partial order (ωcppo) is a poset $(X, \leq)$ with a bottom element, $\bot : X$, in which every increasing sequence $s : \mathbb{N} \to X$ has a least upper bound (lub) $\sqcup s : X$.
- $\mathrm{D}_\approx X$ is the free ωcppo over X.
- Let Y be a ωcppo and $f : X \to Y$. Then:



($\hat{f}$ structure preserving.)

Classifying monads

- A monad T is a classifying monad if there exists an operation

$$\frac{f : X \to TY}{\overline{f} : X \to TX}$$

called restriction, satisfying certain conditions.

$$\begin{array}{rcl} f \diamond \overline{f} & \equiv & f \\ \overline{g} \diamond \overline{f} & \equiv & \overline{f} \diamond \overline{g} \\ \overline{\eta_Y \circ f} & \equiv & \eta_X \\ & \vdots & \end{array}$$

- Idea:
  - $\overline{f}$ identifies the domain of definedness of f. It is the partial identity function associated to f.
  - The pure functions are total.

The additional condition CM6

- Cockett and Lack, condition CM6:

$$\overline{\mathrm{id}_{TX}} \equiv T\eta_X \qquad TX \underset{T\eta_X}{\overset{\overline{\mathrm{id}_{TX}}}{\rightrightarrows}} TTX$$

- Fundamental requirement for connecting classifying monads with partial map categories and partial map classifiers.
- We do not use it in our initiality result.
- Some consequences of CM6:
    - Not all monads are classifying monads: we could choose $\overline{f} = \eta_X$ (i.e. every function is total), but generally $\overline{\mathrm{id}_{TX}} = \eta_{TX} \not\equiv T\eta_X$.
    - It excludes e.g. non-determinism.

Restriction order

- Given T classifying monad, every function space $X \to TY$ is a poset.

$$f \leq g \quad = \quad g \diamond \overline{f} \equiv f$$

- $\leq$ is called restriction order.
- Idea: g is defined on the domain of definedness of f, and it coincides with f on it. g is more defined than f.

Join classifying monads

- A classifying monad T is a join classifying monad if there exist two operations

$$\frac{}{\perp_{X,Y} : X \to TY} \qquad \frac{s : \mathbb{N} \to (X \to TY) \quad \mathsf{isIncr}_{\leq}\, s}{\sqcup s : X \to TY}$$

  satisfying the following conditions:

  BOT1 $\perp_{X,Y} \leq f$

  LUB1 $s\, n \leq \sqcup s$

  LUB2 if $s\, n \leq t$ for all $n : \mathbb{N}$, then $\sqcup s \leq t$

  (Function spaces are ωcppos.)

  BOT2 $\perp_{Y,Z} \diamond f \equiv \perp_{X,Z}$

  LUB3 $\sqcup s \diamond f \equiv \sqcup(\lambda n.\, s\, n \diamond f)$

  (Precomposition is a structure preserving operation).
- Important property: join classifying monads admit unguarded iteration:

$$\frac{f : X \to T(Y + X)}{f^{\dagger} : X \to TY} \qquad f^{\dagger} \equiv [\eta_Y, f^{\dagger}] \diamond f$$

Join classifying monads: examples

- The quotiented delay monad $\mathsf{D}_{\approx}$ is the initial such monad!
- Main result: given a join classifying monad T, there exists a unique join classifying monad morphism $\alpha : \mathsf{D}_{\approx} \Rightarrow T$. ($\alpha$ structure preserving).
- Non-example: maybe monad $\mathsf{Maybe}\, X = X + 1$.
- Non-trivial example:

$$\mathsf{Prop}/X = \sum_{P:Set} \mathsf{isProp}\, P \times (P \to X)$$

(A "proposition" is a type with at most one inhabitant.)

Conclusions

- We defined join classifying monads, a class of monads for non-termination in type theory.
- Capretta's (quotiented) delay monad is canonical among such monads.
- Everything I presented has been fully formalized in Agda.
- Future work:
  - Condition CM6?
  - Other examples of join classifying monads?
  - Join classifying monads, delay monad and weak bisimilarity in a general category.
  - Counterpart of join classifying monads for partial map categories and partial map classifiers.