The Delay Monad and Restriction Categories

James Chapman, Tarmo Uustalu, Niccolò Veltri

Institute of Cybernetics, Tallinn

Theory Days 2013, Saka, 27 October 2013

Motivation

- The delay monad is a viable constructive alternative to the maybe monad.
- It was introduced by Capretta for representing general recursive functions in type theory and it is useful for modeling non-terminating behaviours.
- It has been studied only from a type theoretical point of view. What about a more general (categorical) analysis?
- Restriction categories are an axiomatic framework by Cockett and Lack for reasoning about partiality.

This Talk

- We formalize some parts of the categorical theory of restriction categories (and partial map categories) in Agda.
- We develop the theory of the Delay type in the restriction category setting.
- We set up the basis for the study of the Kleisli category of the delay monad on **Set** (e.g. partial products, joins, meets, iteration operation)

Restriction Categories

- A *restriction category* is a category $\mathbb{X}$ together with an operation (*restriction*) that associates to every $f : A \to B$ a map $\overline{f} : A \to A$ such that

  R1. $f \circ \overline{f} = f$
  R2. $\overline{g} \circ \overline{f} = \overline{f} \circ \overline{g}$ with $f : A \to B, g : A \to C$
  R3. $\overline{g} \circ \overline{f} = \overline{g \circ \overline{f}}$ with $f : A \to B, g : A \to C$
  R4. $\overline{g} \circ f = f \circ \overline{g \circ f}$ with $f : A \to B, g : B \to C$

- A map $f : A \to B$ is called *total*, if $\overline{f} = \mathrm{id}$.
- Intuition : $\overline{f}$ is the "partial identity function" on A specifying the domain of definedness of $f : A \to B$.
- $f \leq g$ if and only if $f = g \circ \overline{f}$.
- f is 'less defined' than g if f coincides with g on f's domain of definedness.

Restriction Categories: Examples

- **Set** (and more generally any category $\mathbb{X}$) is a restriction category with the trivial restriction $\overline{f} = id$
- **Pfn** = "sets and partial functions" is a restriction category with the restriction

$$\overline{f}(x) = \begin{cases} x & \text{if } f(x) \text{ is defined} \\ \text{undefined} & \text{otherwise} \end{cases}$$

- "A single object $\mathbb{N}$ and all partial recursive functions" is a restriction category with restriction as above (for a partially recursive function, it is partially recursive)

The Delay Type

- For a type A, we define $\mathsf{Delay}\ A$ as a coinductive type by the rules

$$\frac{}{\mathsf{now}\ a : \mathsf{Delay}\ A} \qquad \frac{c : \mathsf{Delay}\ A}{\mathsf{later}\ c : \mathsf{Delay}\ A}$$

- We define convergence $\downarrow$ as a binary relation between $\mathsf{Delay}\ A$ and A inductively by the rules

$$\frac{}{\mathsf{now}\ a \downarrow a} \qquad \frac{c \downarrow a}{\mathsf{later}\ c \downarrow a}$$

Equality for the Delay Type : Strong Bisimilarity

- We define strong bisimilarity $\sim$ coinductively via by the rules

$$\frac{}{\text{now } a \sim \text{now } a} \qquad \frac{c \sim c'}{\text{later } c \sim \text{later } c'}$$

- Two computations are 'equal' if they contain the same (possibly infinite) number of later applications.

Equality for the Delay Type : Weak Bisimilarity

- We define weak bisimilarity $\approx$ coinductively via convergence by the rules

$$\frac{c \downarrow a \quad c' \downarrow a}{c \approx c'} \qquad \frac{c \approx c'}{\text{later } c \approx \text{later } c'}$$

- Two computations are 'equal' if they differ for a finite number of applications of the constructor later.

The Delay Monad

- Delay (quotiented by strong/weak bisimilarity) is a strong monad.

$$\eta : A \to \text{Delay } A$$
$$\eta = \text{now}$$

$$\text{bind} : (A \to \text{Delay } B) \to \text{Delay } A \to \text{Delay } B$$
$$\text{bind } f \ (\text{now } a) = f \ a$$
$$\text{bind } f \ (\text{later } c) = \text{later } (\text{bind } f \ c)$$

$$\text{str} : (A \times \text{Delay } B) \to \text{Delay } (A \times B)$$
$$\text{str } (a, c) = \text{map } (\lambda b \to (a, b)) \ c$$

where map is the action of the endofunctor Delay on maps.

Restriction in the Kleisli Category

- The Kleisli category of the delay monad quotiented by weak bisimilarity ($\mathbf{Kl}(\mathsf{Delay}/\approx)$) is a restriction category. Restriction is given in terms of the strength

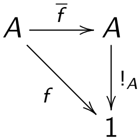
$$\overline{f} = A \xrightarrow{\langle \mathrm{id}, f \rangle} A \times \mathsf{Delay}\ B \xrightarrow{\mathsf{str}} \mathsf{Delay}\ (A \times B) \xrightarrow{\mathsf{Delay}\ \pi_0} \mathsf{Delay}\ A$$

- The Kleisli category of the delay monad quotiented by strong bisimilarity ($\mathbf{Kl}(\mathsf{Delay}/\sim)$) is <u>not</u> a restriction category

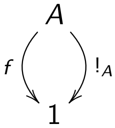
$$f \circ \overline{f} \not\sim f$$

Cartesian Restriction Categories

- A *cartesian restriction category* $\mathbb{X}$ is a restriction category with a partial final object and partial products between any pair of objects.
- A restriction category $\mathbb{X}$ has a *partial final object* if there is an object 1 such that for any map $f : A \to 1$ there is unique map $!_A : A \to 1$ such that

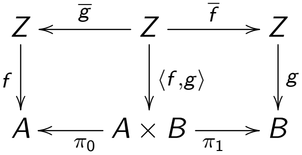


- Compare with the ordinary final object

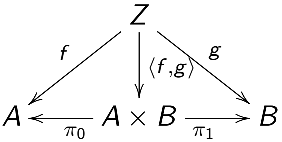


Cartesian Restriction Categories

- A restriction category $\mathbb{X}$ has *binary partial products* if for each pair of objects A and B there is an object $A \times B$ with total maps $\pi_0 : A \times B \to A$, $\pi_1 : A \times B \to B$ such that for any pair of maps $f : Z \to A$, $g : Z \to B$ there is a unique map $\langle f, g \rangle : Z \to A \times B$ such that



- Compare with the ordinary binary product



Partial Final Object in $\mathbf{Kl}(\text{Delay}/\approx)$

- The partial final object is 1. Given an object A the unique good map pointing into 1 is $\text{now} \circ !_A$.
- The ordinary final object is 0. The only map of type $A \to \text{Delay } 0$ is the always undefined one.

Partial Products in $\mathbf{Kl}(\mathsf{Delay}/\approx)$

- The partial product of A and B is $A \times B$ with projections $\mathsf{now} \circ \pi_0$ and $\mathsf{now} \circ \pi_1$, which are total.
- Pairing:

$$
\begin{aligned}
&\langle \_, \_ \rangle : \mathsf{Delay}\ A \to \mathsf{Delay}\ B \to \mathsf{Delay}\ (A \times B) \\
&\langle \mathsf{now}\ a, \mathsf{now}\ b \rangle = \mathsf{now}\ (a, b) \\
&\langle \mathsf{now}\ a, \mathsf{later}\ c \rangle = \mathsf{later}\ \langle \mathsf{now}\ a, c \rangle \\
&\langle \mathsf{later}\ c, \mathsf{now}\ b \rangle = \mathsf{later}\ \langle c, \mathsf{now}\ b \rangle \\
&\langle \mathsf{later}\ c, \mathsf{later}\ c' \rangle = \mathsf{later}\ \langle c, c' \rangle
\end{aligned}
$$

- The pairing is extended to functions pointwise.
- The ordinary product of A and B is $A + B + A \times B$ (as in the category of sets and partial functions).

Restriction Joins and Meets

- In a restriction category a map $f : A \to B$ is the *join* of parallel maps f_1 and f_2 if

  (*i*) $f_1 \leq f$, $f_2 \leq f$

  (*ii*) for any other map g such that $f_1 \leq g$, $f_2 \leq g$ we have $f \leq g$

  i.e. f is the join of f_1 and f_2 in $\mathbb{X}(A, B)$.
- Similarly $f : A \to B$ is the *meet* of f_1 and f_2 if it is the meet of f_1 and f_2 in $\mathbb{X}(A, B)$.

Joins in $\mathbf{Kl}(\mathrm{Delay}/\approx)$

- We define a function join:

$$\begin{aligned}
&\mathrm{join} : \mathrm{Delay}\ A \to \mathrm{Delay}\ A \to \mathrm{Delay}\ A\\
&\mathrm{join}\ (\mathrm{now}\ a)\ c = \mathrm{now}\ a\\
&\mathrm{join}\ (\mathrm{later}\ c)\ (\mathrm{now}\ a) = \mathrm{now}\ a\\
&\mathrm{join}\ (\mathrm{later}\ c)\ (\mathrm{later}\ c') = \mathrm{later}\ (\mathrm{join}\ c\ c')
\end{aligned}$$

- It is extended pointwise to maps.
- The function join above is the join of f and g in $\mathbf{Kl}(\mathrm{Delay}/\approx)$ only if f and g are compatible maps.
- Two maps are compatible if they return the same value whenever they are defined.

Meets in $\mathbf{Kl}(\text{Delay}/\approx)$

- We define a function meet:

$$\begin{aligned}
&\text{meet} : \text{Delay } A \to \text{Delay } A \to \text{Delay } A \\
&\text{meet } (\text{now } a) \; c = \text{now } a \\
&\text{meet } (\text{later } c) \; (\text{now } a) = \text{later } (\text{meet } c \; (\text{now } a)) \\
&\text{meet } (\text{later } c) \; (\text{later } c') = \text{later } (\text{meet } c \; c')
\end{aligned}$$

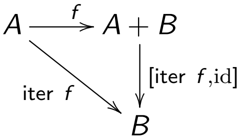
- It is extended pointwise to maps.
- The meet function above is the meet of $f, g : A \to \text{Delay } B$ in $\mathbf{Kl}(\text{Delay}/\approx)$ if B is a semidecidable set.

Iteration Operator

- An *iteration operator* in a category $\mathbb{X}$ is an operation

$$\frac{f : A \to A + B}{\text{iter } f : A \to B}$$

which satisfies



and other axioms.

Iteration in $\mathbf{Kl}(\text{Delay}/\approx)$

- Iteration is defined as

$$\begin{aligned}
&\text{iter}' : (A \to \text{Delay } (A + B)) \to \text{Delay } (A + B) \to \text{Delay } B \\
&\text{iter}' \ f \ (\text{now } (\text{inl } a)) = \text{later } (\text{iter}' \ f \ (f \ a)) \\
&\text{iter}' \ f \ (\text{now } (\text{inr } b)) = \text{now } b \\
&\text{iter}' \ f \ (\text{later } c) = \text{later } (\text{iter}' \ f \ c)
\end{aligned}$$

$$\begin{aligned}
&\text{iter} : (A \to \text{Delay } (A + B)) \to A \to \text{Delay } B \\
&\text{iter } f \ a = \text{iter}' \ f \ (\text{now } (\text{inl } a))
\end{aligned}$$

Conclusion and Future Work

- The Kleisli category of the delay monad on **Set** expresses computability in the sense that it is a cartesian restriction category with joins, meets and iteration.
- It is the starting point for the development of the delay monad theory in general categories.
- We claim that the Kleisli category of the delay monad has more interesting properties (e.g. initial algebra-final coalgebra (limit-colimit) coincidence, Kleisli exponentials, Turing category structure).