Monads and More: Part 4

Tarmo Uustalu, Tallinn

Nottingham, 14-18 May 2007

Coeffectful computation and comonads

For coeffectful notions of computation, we have a comonad (D, ε, δ) on the base category $\mathcal{C}$ of pure functions such that the category of impure functions is $\mathbf{CoKl}(D)$, i.e.,

- an impure function between object A and B of $\mathcal{C}$ can be viewed as a map $A \to^D B$ of $\mathbf{CoKl}(D)$, i.e., a map $DA \to B$ of $\mathcal{C}$,
- the identity impure functions are $\mathrm{id}^D =_{\mathrm{df}} \varepsilon$,
- and the composition of impure functions is $\ell \circ^D k =_{\mathrm{df}} \ell \circ k^\dagger$.

Pure functions are a special case of impure functions via the inclusion $J : \mathcal{C} \to \mathbf{CoKl}(D)$, given by $Jf =_{\mathrm{df}} f \circ \varepsilon$.

Intuition: DA – values from A in a context.

Simplest example: $DA =_{\mathrm{df}} A \times E$ for dependency on environment, but $\mathbf{CoKl}(D) \cong \mathbf{Kl}(T)$ for $TA =_{\mathrm{df}} E \Rightarrow A$.

Dataflow computations

Dataflow computation = discrete-time signal transformations = stream functions.

The output value at a time instant (stream position) is determined by the input value at the same instant (position) plus further input values.

Example dataflow programs

$$
\begin{aligned}
pos &= 0 \text{ fby } (pos + 1) \\
sum\ x &= x + (0 \text{ fby } (sum\ x)) \\
fact &= 1 \text{ fby } (fact * (pos + 1)) \\
fibo &= 0 \text{ fby } (fibo + (1 \text{ fby } fibo))
\end{aligned}
$$

`pos`	0	1	2	3	4	5	6	...
`sum pos`	0	1	3	6	10	15	21	...
`fact`	1	1	2	6	24	120	720	...
`fibo`	0	1	1	2	3	5	8	...

We want to consider functions Str $A \rightarrow$ Str B as impure functions from A to B.

Streams are naturally isomorphic to functions from natural numbers: $\mathsf{Str}A =_{\mathrm{df}} \nu X.A \times X \cong \mathsf{Nat} \Rightarrow A$.

General stream functions $\mathsf{Str}A \rightarrow \mathsf{Str}B$ are thus in natural bijection with maps $\mathsf{Str}A \times \mathsf{Nat} \rightarrow B$.

Comonad for general stream functions

- Functor:

$$DA =_{\mathrm{df}} (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} \cong \mathsf{List}A \times \mathsf{Str}A$$

- Input streams with past/present/future:

$$a_0, a_1, \ldots, a_{n-1}, \boxed{a_n}, a_{n+1}, a_{n+2}, \ldots$$

- Counit:

$$\begin{array}{rcccl} \varepsilon_A & : & (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} & \rightarrow & A \\ & & (a, n) & \mapsto & a(n) \end{array}$$

- Co-Kleisli extension:

$$\frac{k \;:\; (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} \;\rightarrow\; B}{\begin{array}{rcccl} k^\star & : & (\mathsf{Nat} \Rightarrow A) \times \mathsf{Nat} & \rightarrow & (\mathsf{Nat} \Rightarrow B) \times \mathsf{Nat} \\ & & (a, n) & \mapsto & (\lambda m\, k(a, m), n) \end{array}}$$

Comonad for causal stream functions

- Functor: $DA =_{\mathrm{df}} \mathsf{NEList} \cong \mathsf{List}A \times A$
- Input streams with past and present but no future
- Counit:

$$\begin{array}{rcccl} \varepsilon_A & : & \mathsf{NEList}A & \longrightarrow & A \\ & & [a_0, \ldots, a_n] & \mapsto & a_n \end{array}$$

- Co-Kleisli extension:

$$\frac{k \ : \ \mathsf{NEList}A \ \longrightarrow \ B}{\begin{array}{rcccl} k^\star & : & \mathsf{NEList}A & \longrightarrow & \mathsf{NEList}B \\ & & [a_0, \ldots, a_n] & \mapsto & [k[a_0], k[a_0, a_1], \ldots, k[a_0, \ldots, a_n]] \end{array}}$$

Comonad for anticausal stream functions

- Input streams with present and future but no past
- Functor: $DA =_{\mathrm{df}} \mathsf{Str}A \cong \ A \times \mathsf{Str}A$

Relabelling tree transformations

Let $F : \mathcal{C} \to \mathcal{C}$. Define $\mathsf{Tree}A =_{\mathrm{df}} \mu X.A \times FX$. We are interested in functions $\mathsf{Tree}A \to \mathsf{Tree}B$.

(Alt. we can define $\mathsf{Tree}^\infty A =_{\mathrm{df}} \nu X.A \times FX$ and interest ourselves in functions $\mathsf{Tree}^\infty A \to \mathsf{Tree}^\infty B$.)

Comonad for general relabelling functions:

$$DA =_{\mathrm{df}} \mathsf{Path}A \times \mathsf{Tree}A$$

(Huet's zipper) where $\mathsf{Path}A =_{\mathrm{df}} \mu X.1 + A \times F'(\mathsf{Tree}A) \times X$.

Comonad for bottom-up relabelling functions:

$$DA =_{\mathrm{df}} \mathsf{Tree}A$$

Co-Kleisli categories and Cartesian closed structure

Let D be a comonad on a Cartesian closed cat. $\mathcal{C}$.

Since J is right adjoint and preserves limits, **CoKl**(D) has products. Explicitly, we can define

$$\begin{array}{rcl} A \times^D B & =_{\mathrm{df}} & A \times B \\ \pi_0^D & =_{\mathrm{df}} & \mathsf{fst} \circ \varepsilon \\ \pi_1^D & =_{\mathrm{df}} & \mathsf{snd} \circ \varepsilon \\ \langle k_0, k_1 \rangle^D & =_{\mathrm{df}} & \langle k_0, k_1 \rangle \end{array}$$

If D is *strong/lax symmetric semimonoidal* wrt. $(1, \times)$, i.e., comes with a nat. iso./transf. $\mathsf{m} : DA \times DB \to D(A \times B)$, then we can also define

$$\begin{array}{rcl} A \Rightarrow^D B & =_{\mathrm{df}} & DA \Rightarrow B \\ \mathsf{ev}^D & =_{\mathrm{df}} & \mathsf{ev} \circ \langle \varepsilon \circ D\mathsf{fst}, D\mathsf{snd} \rangle \\ \Lambda^D(k) & =_{\mathrm{df}} & \Lambda(k \circ \mathsf{m}) \end{array}$$

$$D((DA \Rightarrow B) \times A) \xrightarrow{\langle \varepsilon \circ D\mathsf{fst}, D\mathsf{snd} \rangle} (DA \Rightarrow B) \times DA \xrightarrow{\mathsf{ev}} B$$

$$\frac{DC \times DA \xrightarrow{\mathsf{m}} D(C \times A) \xrightarrow{k} B}{DC \xrightarrow{\Lambda(k \circ \mathsf{m})} DA \Rightarrow B}$$

Using a strength (if available) is not a good idea: We have no multiplication

$$DC \times DA \xrightarrow{\mathsf{sl}} D(C \times DA) \xrightarrow{D\mathsf{sr}} DD(C \times A) \xrightarrow{?} D(C \times A)$$

and applying ε or $D\varepsilon$ gives a solution where the order of arguments of a function is important and coeffects do not combine:

$$DC \times DA \xrightarrow{\mathsf{id} \times \varepsilon} DC \times A \xrightarrow{\mathsf{sl}} D(C \times A)$$

or

$$DC \times DA \xrightarrow{\varepsilon \times \mathsf{id}} C \times DA \xrightarrow{\mathsf{sr}} D(C \times A)$$

If D is strong semimonoidal (in which case it is automatically strong symmetric semimonoidal), then $A \Rightarrow^D -$ is right adjoint to $- \times^D A$ and hence $\Rightarrow^D$ is an exponent functor:

$$\frac{\dfrac{D(C \times A) \to B}{DC \times DA \to B}}{DC \to DA \Rightarrow B}$$

This is the case, e.g., if $DA =_{\mathrm{df}} \nu X.A \times (K \Rightarrow X)$ for some K (e.g., $DA =_{\mathrm{df}} \mathsf{Str} A$).

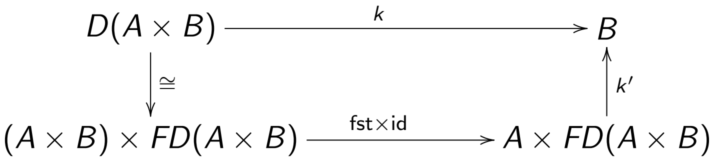
More typically, D is only lax symmetric semimonoidal.

Then it suffices to have m satisfying $\mathrm{m} \circ \Delta = D\,\Delta$, where $\Delta = \langle \mathrm{id}, \mathrm{id} \rangle : A \to A \times A$ is part of the comonoid structure on the objects of $\mathcal{C}$, to get that $m \circ \langle D\mathrm{fst}, D\mathrm{snd} \rangle = \mathrm{id}$ and that $\Rightarrow^D$ is a weak exponent operation on objects. It is not functorial (not even in each argument separately).

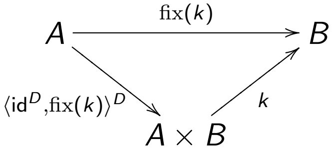
Partial uniform parameterized fixpoint operator

Let $F : \mathcal{C} \to \mathcal{C}$. Define $DA =_{\mathrm{df}} \nu Z.A \times FZ$.

Call a coKleisli map $k : A \times B \to^D B$ *guarded* if for some k' we have



For any guarded $k : A \times B \to^D B$, there is a unique map $\mathrm{fix}(k) : A \to^D B$ satisfying



fix is a partial *Conway operator* defined on guarded maps, i.e., besides the *fixpoint property*, for any guarded $k : A \times^D B \to^D B$,

$$\mathrm{fix}(k) = k \circ^D \langle \mathsf{id}^D, \mathrm{fix}(k)\rangle^D$$

it satisfies *naturality* in A, *dinaturality* in B, and the *diagonal property*: for any guarded $k : A \times^D B \times^D B \to^D B$,

$$\mathrm{fix}(k \circ^D (\mathsf{id}^D \times^D \Delta^D)) = \mathrm{fix}(\mathrm{fix}(k))$$

Wrt. pure maps, fix is also *uniform* (i.e., strongly dinatural in B instead of dinatural), i.e., for any guarded $k : A \times^D B \to^D B$, $\ell : A \times^D B' \to^D B'$ and $h : B \to B'$

$$Jh \circ^D k = \ell \circ^D (\mathsf{id}^D \times^D Jh) \quad \Longrightarrow \quad Jh \circ^D \mathrm{fix}(k) = \mathrm{fix}(\ell)$$

Comonadic semantics

As in the case of monadic semantics, we interpret the lambda-calculus into **CoKl**(D) in the standard way, getting

$$
\begin{array}{rcll}
[\![A \times B]\!]^D & =_{\mathrm{df}} & [\![A]\!]^D \times^D [\![B]\!]^D & = [\![A]\!]^D \times [\![B]\!]^D \\
[\![A \Rightarrow B]\!]^D & =_{\mathrm{df}} & [\![A]\!]^D \Rightarrow^D [\![B]\!]^D & = D[\![A]\!]^D \Rightarrow [\![B]\!]^D \\
[\![(\underline{x})\, x_i]\!]^D & =_{\mathrm{df}} & \pi_i^D & = \pi_i \circ \varepsilon \\
[\![(\underline{x})\, \mathit{fst}(t)]\!]^D & =_{\mathrm{df}} & \pi_0^D \circ^D [\![(\underline{x})\, t]\!]^D & = \mathsf{fst} \circ [\![(\underline{x})\, t]\!]^D \\
[\![(\underline{x})\, \mathit{snd}(t)]\!]^D & =_{\mathrm{df}} & \pi_1^D \circ^D [\![(\underline{x})\, t]\!]^D & = \mathsf{snd} \circ [\![(\underline{x})\, t]\!]^D \\
[\![(\underline{x})\, (t_0, t_1)]\!]^D & =_{\mathrm{df}} & \langle [\![(\underline{x})\, t_0]\!]^D, [\![(\underline{x})\, t_1]\!]^D \rangle^D & = \langle [\![(\underline{x})\, t_0]\!]^D, [\![(\underline{x})\, t_1]\!]^D \rangle \\
[\![(\underline{x})\, \lambda x t]\!]^D & =_{\mathrm{df}} & \Lambda^D([\![(\underline{x}, x)\, t]\!]^D) & = \Lambda([\![(\underline{x}, x)\, t]\!]^D \circ \mathsf{m}) \\
[\![(\underline{x})\, t\, u]\!]^D & =_{\mathrm{df}} & \mathsf{ev}^D \circ^D \langle [\![(\underline{x})\, t]\!]^D, [\![(\underline{x})\, u]\!]^D \rangle^D & = \mathsf{ev} \circ \langle [\![(\underline{x})\, t]\!]^D, ([\![(\underline{x})\, u]\!]^D)^\dagger \rangle \\
[\![(\underline{x})\, \mathsf{rec}\, x t]\!]^D & =_{\mathrm{df}} & \mathrm{fix}^D([\![(\underline{x}, x)\, t]\!]^D) & \text{for } (\underline{x}, x)\, t \text{ syntactically guarded}
\end{array}
$$

Coeffect-specific constructs are interpreted specifically.

Again, $\underline{x} : \underline{C} \vdash t : A$ implies $[\![(\underline{x})t]\!]^D : [\![\underline{C}]\!]^D \rightarrow^D [\![A]\!]^D$, but not all equations of the lambda-calculus are validated.

Closed terms: Soundness of typing for $\vdash t : A$ says that $[\![t]\!]^D : 1 \to^D [\![A]\!]^D$, i.e., $D1 \to [\![A]\!]^D$, so closed terms are evaluated relative to a coeffect over 1.

In case of general or causal stream functions, this is a list over 1 (i.e., a natural number), the time elapsed.

If D is properly symmetric monoidal (e.g., Str), we have a canonical choice $\mathrm{e} : 1 \xrightarrow{\sim} D1$.

Comonadic dataflow language semantics: The first-order language agrees perfectly with Lucid and Lustre by its semantics.

The meaning of higher-order dataflow computation has been unclear. We get a neat semantics from mathematical considerations (cf. Colaço, Pouzet's design with two flavors of function spaces).

Related linear/modal logic work

Strong symmetric monoidal comonads are central in the semantics of intuitionistic linear logic and modal logic to interpret the ! and $\Box$ operators.

Linear logic: Benton, Bierman, de Paiva, Hyland; Bierman; Benton; Mellies; Maneggia; etc.

Modal logic: Bierman, da Paiva.

Applications to staged computation and semantics of names: Pfenning, Davies, Nanevski.