Impurity and modularity from monads and coproducts

Tarmo Uustalu

(based on joint work with Neil Ghani)

Teooriapäevad Kokel, 30.1.–1.2.2004

Motivation

- Monads are an excellent machinery to represent and reason about the semantics of impure languages, i.e., languages with side-effects (exceptions, state, continuations), systematically in a uniform fashion.
- For modular modelling and reasoning, systematic ways of combining monads are therefore desirable.
- The methods based on distributivity laws and monad transformers are not entirely satisfactory.
- Taking the coproduct of two monads (in the category of monads) is the perfect solution in more than one sense, but this is hard to construct.

This paper

- A general construction of colimits of finitary monads / monads with rank on an lfp category / accessible category has been given by Kelly (Bull. Austr. Math. Soc. 1980).
- Lüth, Ghani (FroCoS 2002) gave three simpler constructions of the coproduct of two finitary *ideal* monads, but these involve colimits of chains and quotienting and are thus not directly implementable.
- We give a fixed point formula for calculating the coproduct of two *ideal* monads (making no rank assumptions).

Outline

- Notions of computation and monads
- Modularity and coproducts of monads
- Ideal monads and how to calculate their coproducts

Monads

- A monad on a category $\mathcal{C}$ is an endofunctor T on $\mathcal{C}$ together with nat. transfs. $\eta : \mathsf{Id} \to T$ and $m : T \cdot T \to T$ s.t.

$$\begin{array}{ccc} T & \xrightarrow{\eta \cdot T} & T \cdot T \\ {\scriptstyle T\cdot\eta}\downarrow & \searrow^{=} & \downarrow{\scriptstyle m} \\ T \cdot T & \xrightarrow[m]{} & T \end{array} \qquad \begin{array}{ccc} T \cdot T \cdot T & \xrightarrow{m \cdot T} & T \cdot T \\ {\scriptstyle T\cdot m}\downarrow & & \downarrow{\scriptstyle m} \\ T \cdot T & \xrightarrow[m]{} & T \end{array}$$

- Intuition: T is a notion of computation: For A a type, TA is the correspoding type of computations, η is insertion of values into computations, m flattens nested computations.

- A monad morphism from $T = (T, \eta, m)$ to $T' = (T', \eta', m')$ is a nat. transf. $f : T \to T'$ s.t.

$$
\begin{array}{ccc}
\mathsf{Id} & \xrightarrow{\eta} & T \\
\Vert & & \downarrow{\scriptstyle f} \\
\mathsf{Id} & \xrightarrow[\eta']{} & T'
\end{array}
\qquad\qquad
\begin{array}{ccc}
T \cdot T & \xrightarrow{m} & T \\
{\scriptstyle f \cdot f}\downarrow & & \downarrow{\scriptstyle f} \\
T' \cdot T' & \xrightarrow[m']{} & T'
\end{array}
$$

- Intuition: f converts computations according to notion T to computations according to notion T' respecting values and flattening of nested computations.

Kleisli category of a monad

- A monad $T = (T, \eta, m)$ on a category $(\mathcal{C}, \mathsf{id}, \circ)$ determines a category $(\mathcal{C}_T, \mathsf{jd}, \bullet)$, called its Kleisli category:
  - $|\mathcal{C}_T| = |\mathcal{C}|$,
  - $\mathcal{C}_T(A, B) = \mathcal{C}(A, TB)$,
  - for $A \in |\mathcal{C}_T|$, $\mathsf{jd}_A = \eta_A$,
  - for $f \in \mathcal{C}_T(A, B)$, $g \in \mathcal{C}_T(B, C)$, $g \bullet f = m_C \circ Tg \circ f$.
- Intuition: If T captures the notion of computation appropriate for some impure extension of a pure language for which $\mathcal{C}$ is model, then $\mathcal{C}_T$ is a model of the extended language.

- Some examples:
  - Exceptions: $\mathsf{Exc}_E\ A = A + E$.
  - Output: $\mathsf{Outp}_C\ A = A \times \mathsf{List}\ C = \mu X.\ A + C \times X$.
  - Non-determinism: $\mathsf{NDet}\ A = \mathcal{P}A$.
  - Probabilistic choice: $\mathsf{PrCh}\ A = \mathcal{P}(A \times \mathbb{R}^+)/{\sim}$.
  - Time: $\mathsf{Time}\ A = A \times \mathsf{Nat} = \mu X.\ A + X$.
  - Non-termination: $\mathsf{NTerm}\ A = \nu X.\ A + X$.
  - State: $\mathsf{State}_S\ A = S \Rightarrow A \times S$.
  - Continuations: $\mathsf{Cont}_R\ A = (A \Rightarrow R) \Rightarrow R$.
  - Free monads (term algebras): $F^\mu = \mu X.\ \mathsf{Id} + F \cdot X$.
  - Completely free monads (non-wellfounded term algebras): $F^\nu = \nu X.\ \mathsf{Id} + F \cdot X$.

Distributive laws

- Given two monads (R, η^R, m^R) and (S, η^S, m^S), a distributive law of the first over the second is a nat. transf. $\lambda : R \cdot S \to S \cdot R$ subject to four coherence conditions.
- Given a distributive law, there is the compatible monad $(S \cdot R, \eta, m)$ with $\eta = \eta_S \cdot \eta_R$, $m = (m^S \cdot m^R) \circ (S \cdot \lambda \cdot R)$.
- If the distributive law satisfies some additional conditions, then the compatible monad is the coproduct.

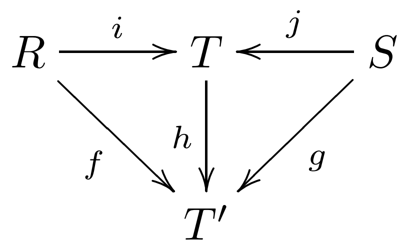
Monad transformers

- A monad transformer is a pointed functor F on $\mathbf{Monad}(\mathcal{C})$.
- For many monads there are natural accompanying monad transformers, e.g.
  - Exceptions monad transformer: $(\mathsf{ExcT}_E\ R)\ A = R\ (A + E)$.
  - State monad transformer: $(\mathsf{StateT}_S\ R)\ A = S \Rightarrow R\ (A \times S)$.

Coproducts of monads

- A coproduct of two monads on $\mathcal{C}$ is a coproduct of them as objects of $\mathbf{Monad}(\mathcal{C})$.

  I.e.: a coproduct of two monads R, S is a monad T together with monad morphisms $i : R \to T$, $j : S \to T$ s.t., for any monad T' and monad morphisms $f : R \to T'$, $g : S \to T'$, there exists a unique monad morphism $h : T \to T'$ satisfying

  If R, S have a coproduct, we denote it $R \oplus S$.

  It is certainly <u>not</u> the case that $R \oplus S = R + S$: in general, there is no way to get a nat. transf. $m : (R + S) \cdot (R + S) \to R + S$.

- Intuition: $R \oplus S$ is the least notion of computation than contains (in disjoint fashion) both R and S.

- Example: Let $R \oplus \mathsf{Exc}_E = R \cdot \mathsf{Exc}_E$.
- Coproducts are one canonical construction delivering monad transformers: given some monad S on $\mathcal{C}$, the functor $F : \mathbf{Monad}(\mathcal{C}) \to \mathbf{Monad}(\mathcal{C})$ given by $FR = R \oplus S$ is a monad transformer.

Free monads

- The free monad of an endofunctor F on a category $\mathcal{C}$ is the universal arrow from F to the forgetful functor $U : \mathbf{Monad}(\mathcal{C}) \to [\mathcal{C}, \mathcal{C}]$.
- Given an endofunctor F on a category $\mathcal{C}$, the underlying functor of its free monad is $F^\mu = \mu X.\ \mathsf{Id} + F \cdot X$. i.e. the least solution of

$$X \cong \mathsf{Id} + F \cdot X$$

- For $\mathcal{C} = \mathbf{Set}$ and F polynomial, this is the term algebra monad induced by F as a signature.

Coproducts of free monads

- The coproduct of two free monads is easy to construct:

$$F^\mu \oplus G^\mu = (F + G)^\mu = \mu X.\ \mathsf{Id} + F \cdot X + G \cdot X$$

- Hyland, Plotkin, Power (IFIP TCS 2002) have also given a construction of the coproduct of any monad with a free monad:

$$R \oplus F^\mu = R \cdot (F \cdot R)^\mu = R \cdot (\mu X.\ \mathsf{Id} + F \cdot R \cdot X) = \mu X.\ R \cdot (\mathsf{Id} + F \cdot X)$$

Ideal monads

- A monad (T, η, m) is said to be *ideal* (Aczel, Adámek et al., CMCS 2001) if there exist T_0, $\tau : T_0 \to T$, $m_0 : T_0 \cdot T \to T_0$ s.t. $[\,\eta, \tau\,] : \mathsf{Id} + T_0 \to T$ is iso and

$$\begin{array}{ccc} T_0 \cdot T & \xrightarrow{\tau \cdot T} & T \cdot T \\ {\scriptstyle m_0}\downarrow & & \downarrow{\scriptstyle m} \\ T_0 & \xrightarrow[\tau]{} & T \end{array}$$

- Without loss of generality, we assume $T = \mathsf{Id} + T_0$, so $\eta = \mathsf{inl}$, $\tau = \mathsf{inr}$.

- Intuitively: Every computation is either a value or a non-value and flattening of a non-value computation of a computation must give a non-value computation.

- If $f : T \to T'$ is a monad morphism from (T, η, m) to (T', η', m') with an ideal source, then $f = [\,\eta', f_0\,]$ for some $f_0 : T_0 \to T'$.

Ideal monads: examples

- Exceptions: $\mathsf{Exc}_E\ A = A + E$.
- Output: $\mathsf{Outp}_C\ A = A \times \mathsf{List}\ C \cong A + A \times \mathsf{NEList}\ C$.
- Non-deadlocking non-determinism: $\mathsf{NDet}_{\geq 1}\ A = \mathcal{P}_{\geq 1} A \cong A + \mathcal{P}_{\geq 2} A$.
- Non-deadlocking probabilistic choice:
  $\mathsf{PrCh}_{\geq 1}\ A = \mathcal{P}_{\geq 1}(A \times \mathbb{R}^+)/{\sim} \cong A + \mathcal{P}_{\geq 2}(A \times \mathbb{R}^+)/{\sim}$.
- Time: $\mathsf{Time}\ A = A \times \mathsf{Nat} \cong A + A \times \mathsf{Nat} = A + \mathsf{Time}\ A$.
- Non-termination: $\mathsf{NTerm}\ A = \nu X.\ A + X \cong A + \mathsf{NTerm}\ A$.
- Free monads: $F^\mu = \mu X.\ \mathsf{Id} + F \cdot X \cong \mathsf{Id} + F \cdot F^\mu$.
- Free completely iterative monads: $F^\nu = \mu X.\ \mathsf{Id} + F \cdot X \cong \mathsf{Id} + F \cdot F^\nu$.

Coproducts of ideal monads

- Our result: The coproduct of two ideal monads $R = \mathsf{Id} + R_0$ and $S = \mathsf{Id} + S_0$ is

  $$T = \mathsf{Id} + (T_1 + T_2)$$

  where

  $$(T_1, T_2) = \mu(X, Y).\ (R_0 \cdot (\mathsf{Id} + Y), S_0 \cdot (\mathsf{Id} + X))$$

  i.e., (T_1, T_2) is the least solution of the system

  $$X \cong R_0 \cdot (\mathsf{Id} + Y)$$
  $$Y \cong S_0 \cdot (\mathsf{Id} + X)$$

- Intuitively: T is given by (*strictly*) alternating R_0 and S_0 on top of Id in a wellfounded way:
  $T \cong \mathsf{Id} + (R_0 + S_0) + (R_0 \cdot S_0 + S_0 \cdot R_0) + (R_0 \cdot S_0 \cdot R_0 + S_0 \cdot R_0 \cdot S_0) + \ldots.$

- This first guess is <u>wrong</u>:
  $T = T_1 + T_2$ where $(T_1, T_2) = \mu(X_1, X_2).\ (R \cdot (\mathsf{Id} + Y), S \cdot (\mathsf{Id} + X)$.

- Constructions: The unit is $\eta = \mathsf{inl} : \mathsf{Id} \to \mathsf{Id} + (T_1 + T_2)$.

- The multiplication is $m = [\,T, \mathsf{inr} \circ (m_1 + m_2)\,] : T + (T_1 \cdot T + T_2 \cdot T) \to T$ where $m_1 : T_1 \cdot T \to T_1$ and $m_2 : T_2 \cdot T \to T_2$ are constructed by mutual iteration

$$
\begin{array}{ccc}
R_0 \cdot (\mathsf{Id} + T_2) \cdot T & \xrightarrow{\mathsf{in}_1 \cdot T} & T_1 \cdot T \\
{\scriptstyle R_0 \cdot (T + m_2)} \downarrow & & \downarrow {\scriptstyle m_1} \\
R_0 \cdot (T + T_2) & \xrightarrow[p_1]{} & T_1
\end{array}
\qquad
\begin{array}{ccc}
T_2 \cdot T & \xleftarrow{\mathsf{in}_2 \cdot T} & S_0 \cdot (\mathsf{Id} + T_1) \cdot T \\
{\scriptstyle m_2} \downarrow & & \downarrow {\scriptstyle S_0 \cdot (T + m_1)} \\
T_2 & \xleftarrow[p_2]{} & S_0 \cdot (T + T_1)
\end{array}
$$

and p_1, p_2 are the composites

$$R_0 \cdot (T + T_2) \longrightarrow R_0 \cdot ((\mathsf{Id} + T_2) + T_1) \xrightarrow{R_0 \cdot ((\mathsf{Id}+T_2)+\mathsf{in}_1^{-1})} R_0 \cdot R \cdot (\mathsf{Id} + T_2) \xrightarrow{m_0^R \cdot (\mathsf{Id}+T_2)} R_0 \cdot (\mathsf{Id} + T_2) \xrightarrow{\mathsf{in}_1} T_1$$

$$S_0 \cdot (T + T_1) \longrightarrow S_0 \cdot ((\mathsf{Id} + T_1) + T_2) \xrightarrow{S_0 \cdot ((\mathsf{Id}+T_1)+\mathsf{in}_2^{-1})} S_0 \cdot S \cdot (\mathsf{Id} + T_1) \xrightarrow{m_0^S \cdot (\mathsf{Id}+T_1)} S_0 \cdot (\mathsf{Id} + T_1) \xrightarrow{\mathsf{in}_2} T_2$$

- The injections are $\mathsf{Id} + (\mathsf{inl} \circ i_1) : \mathsf{Id} + R_0 \to T$ and $\mathsf{Id} + (\mathsf{inr} \circ i_2) : \mathsf{Id} + S_0 \to T$ where i_1, i_2 are the composites

$$R_0 \xrightarrow{R_0 \cdot \mathsf{inl}} R_0 \cdot (\mathsf{Id} + T_2) \xrightarrow{\mathsf{in}_1} T_1$$

and

$$S_0 \xrightarrow{S_0 \cdot \mathsf{inl}} S_0 \cdot (\mathsf{Id} + T_1) \xrightarrow{\mathsf{in}_2} T_2$$

- The copair of two monad morphisms $f = [\,\eta', f_0\,] : \mathsf{Id} + R_0 \to T'$, $g = [\,\eta', g_0\,] : \mathsf{Id} + S_0 \to T'$ is $h = [\,\eta', [\,h_1, h_2\,]\,] : \mathsf{Id} + (T_1 + T_2) \to T'$ where $h_1 : T_1 \to T'$ and $h_2 : T_2 \to T'$ are constructed by mutual iteration

$$
\begin{array}{ccccccc}
R_0 \cdot (\mathsf{Id} + T_2) & \xrightarrow{\mathsf{in}_1} & T_1 & \qquad & T_2 & \xleftarrow{\mathsf{in}_2} & S_0 \cdot (\mathsf{Id} + T_1) \\
{\scriptstyle R_0 \cdot (\mathsf{Id} + h_2)} \downarrow & & \downarrow {\scriptstyle h_1} & & {\scriptstyle h_2} \downarrow & & \downarrow {\scriptstyle S_0 \cdot (\mathsf{Id} + h_1)} \\
R_0 \cdot (\mathsf{Id} + T') & \xrightarrow[q_1]{} & T' & & T' & \xleftarrow[q_2]{} & S_0 \cdot (\mathsf{Id} + T')
\end{array}
$$

in which q_1, q_2 are the composites

$$R_0 \cdot (\mathsf{Id} + T') \xrightarrow{f_0 \cdot [\,\eta', T'\,]} T' \cdot T' \xrightarrow{m'} T'$$

and

$$S_0 \cdot (\mathsf{Id} + T) \xrightarrow{g_0 \cdot [\,\eta', T'\,]} T' \cdot T' \xrightarrow{m'} T'$$

Coproducts of ideal monads: examples

- Non-deadlocking non-determinism and probabilistic choice: finite alternations of non-trivial non-determinism and non-trivial probabilistic choice on top of values.
- Output and non-termination: finite alternations of non-zero time and non-empty output on top of non-termination.
- $F^\nu \oplus G^\nu \ncong (F+G)^\nu$.

  ($(F+G)^\nu$ allows for infinite alternations between $F \cdot F^\nu$ and $G \cdot G^\nu$ whereas the coproduct only permits finite alternation.)

Conclusions

- Although the general construction of coproducts (colimits) of monads is highly complex, coproducts (colimits) of monads can be constructed relatively simply in special cases using fixed point techniques.
- This said, one should still be very careful when seeking help from intuition, it is easy to arrive at wrong solutions.