Stateful runners for effectful computations

Tarmo Uustalu, Institute of Cybernetics, Tallinn

FMCS 2015, Hamilton, NY, 4-7 June 2015
Effectful computations in functional programming

- Singe Moggi 1989, in functional programming we use monads to organize effectful (eg, interactive, stateful, nondeterministic) computations.
- Effectful functions are maps in the Kleisli category of the appropriate monad.
- For a set $X$ of values, effectful computations are elements of $TX$ for some monad $(T, \eta, \mu)$.
- An effectful function from $X$ to $Y$ is a function from $X$ to $TY$.
- The identity effectful function is $X \xrightarrow{\eta_X} TX$.
- Composition of $k : X \to TY$, $\ell : Y \to TZ$ is given by

$$X \xrightarrow{k} TY \xrightarrow{T\ell} T(TZ) \xrightarrow{\mu_Z} TZ$$

- Plotkin, Power 2002 emphasized that (finitary) monads arise uniquely from Lawvere theories.
How to run an effectful computation?

- How can one run (in the sense of extracting a value from) an effectful computation?
- For a general \((T, \eta, \mu)\) we cannot achieve
  \[
  \theta_X : T X \to X
  \]
  but it makes perfect sense to look for sets \(C\) such that we can have
  \[
  \theta_X : T X \to C \Rightarrow C \times X
  \]
  \[
  T^C X
  \]
- If we are serious, we want \(\theta\) to be a monad morphism.
- Such \((C, \theta)\) serve as “platforms” for running computations. I call them runners.
- Runners as not the same as Plotkin and Pretnar’s handlers, but they are related.
Outline

- Three example characterisations of runners
- A general theorem (refers to Lawvere theories/models/comodels)
- 3 slides about Lawvere theories
- The examples derived
Interactive I/O

- An interactive computation for sets $I$ and $O$ (input and output alphabets) over a set $X$ is an element of $T X = \mu Z. X + (I \Rightarrow Z) + (O \times Z)$, i.e., a communication tree.

- A runner for such computations is a responder/listener for $I$ and $O$, i.e.,

  a set $C$ with functions

  \[
  \text{inp} = \langle \text{inps}, \text{inpn} \rangle : C \rightarrow I \times C \\
  \text{upd} : C \times O \rightarrow C
  \]
A stateful computation for a set $S$ (of states) over a set $X$ is an element of $T X = S \Rightarrow S \times X$.

A runner for such computations is a lens for $S$ as the set of views, i.e.,

a set $C$ (of sources) with functions

$$lkps : C \rightarrow S$$
$$upd : C \times S \rightarrow C$$

such that

$$lkps(upd(c, s)) = s$$
$$upd(c, lkps c) = c$$
$$upd(upd(c, s), s') = upd(c, s')$$
Updates

- An *update* computation for a set $S$ (of states), a monoid $(P, \circ, \oplus)$ (of updates) and a right action $\downarrow : S \times P \to S$ (update application) is an element of $T X = S \Rightarrow P \times X$.

- A runner for such computations is an *update lens* for $S$, $(P, \circ, \oplus)$ and $\downarrow$, i.e.,

  a set $C$ with functions

  \[
  lkp : C \to S \\
  upd : C \times P \to C
  \]

  such that

  \[
  lkp ( upd (c, p)) = lkp c \downarrow p \\
  \text{upd} (c, \circ) = c \\
  \text{upd} (\text{upd} (c, p), p') = \text{upd} (c, p \oplus p')
  \]
Nondeterminism (1)

- A *nondeterministic* computation is an element of $T X = \mu Z. X + Z \times Z$, i.e., a binary leaf tree of values.
- A runner for nondeterministic computations is a *resolver*, i.e.,

  a set $C$ with a function

  $$ch : C \rightarrow C + C$$

  or, equivalently

  $$ch = \langle chn, chs \rangle : C \rightarrow C \times 2$$
Nondeterminism (2)

- Or maybe we have a different idea of nondeterminism.
- A nondeterministic computation is an element of $T \times X = \mu Z. X \times (1 + Z) = X^+$, i.e., a list of values.
- Now a runner is

  a set $C$ with a function

  $$ch = \langle chn, chs \rangle : C \rightarrow C \times 2$$

  such that

  $$chs (chn c) = chs c$$

  $$chn (chn c) = chn c$$

  (so only the first and last element of a list can be extracted)
Nondeterminism (3)

- A nondeterministic computation is an element of $T \mathcal{X} = \mathcal{X}^+ | \text{squarefree}$, i.e., a (nonempty) lists of values where no sublist occurs twice in a row.
- Now a runner is a set $C$ with a function $ch = \langle chn, chs \rangle : C \to C \times 2$

  such that

  $chs\ (chn\ c) = chs\ c$

  $chn\ (chn\ c) = chn\ c$

  $chn\ c = c$

  which is the same as simply having a set $C$ with a function $chs : C \to 2$. 
We could also want the following:

A nondeterministic computation is an element of $T X = X^+/\text{perm}$, i.e., a (nonempty) multiset of values.

Now a runner is

a set $C$ with a function

$$ch = \langle chn, chs \rangle : C \to C \times 2$$

such that

$$chs (chn c) = chs c$$
$$chn (chn c) = chn c$$
$$chs c = \neg chs c$$

which really means that $C$ must be 0
Nondeterminism (5)

- We could also want the following:
- A nondeterministic computation is an element of $TX = \mu Z. X + 1 + Z \times Z$, i.e., a nullary-binary leaf tree over $X$.
- A resolver is now a set $C$ with functions:

\[
    \begin{align*}
        die & : C \to 0 \\
        ch & : C \to C + C
    \end{align*}
\]

- This time the presence of $die$ forces $C$ to be 0, i.e., there is only a trivial runner.
- (NB! The same is true for exceptions and more generally for any $T$ such that $T0 \not\cong 0$.)
The general case

- Given a (finitary) monad \((T, \eta, \mu)\), there is an associated comonad \((D, \varepsilon, \delta)\) such that, for any set \(C\),

  monad morphisms

  \[ \theta_X : T X \rightarrow C \Rightarrow C \times X \]

  between \((T, \eta, \mu)\) and the state monad for \(C\)

  are in a bijective correspondence with

  comonad-coalgebra structures

  \[ \gamma : C \rightarrow D C \]

  of \((D, \varepsilon, \delta)\) on \(C\)
Identifying the relevant comonad

- Given a (finitary) monad,
- we identify the corresponding Lawvere theory (whose models are the same as algebras of the monad),
- then work out the corresponding comonad (whose coalgebras are the same as comodels of the Lawvere theory)
Lawvere theories

- A (finitary) Lawvere theory is given by a small category $\mathbb{L}$ with finite products and a functor
  \[ L : \mathbb{F}^{\text{op}} \to \mathbb{L} \]
  that is identity on objects and strictly preserves the finite products of $\mathbb{F}^{\text{op}}$.
  Here $\mathbb{F}$ is the category of finite cardinals.
- A theory can be specified by a presentation, i.e., by some subset of the maps
  \[ \text{OP}_j : I_j \to O_j \]
  of $\mathbb{L}$ (operations) from which all other maps are definable together with some subset of the commuting diagrams
  \[ \text{LHS}_k = \text{RHS}_k \]
  of $\mathbb{L}$ (equations) from which all other commuting diagrams follow.
Models

- A model of a theory \((\mathbb{L}, L)\) is given by a functor
  \[
  [-] : \mathbb{L} \to \text{Set}
  \]
  that preserves the finite products of \(\mathbb{L}\) (non-strictly).
- To give a model, it suffices to specify a set
  \[
  A = [1]
  \]
  since, for any other object \(Y\), we have
  \[
  [Y] = \bigsqcup_{y \in Y} 1 \cong \prod_{y \in Y} [1] = Y \Rightarrow [1],
  \]
  together with functions
  \[
  op_j = [OP_j] : l_j \Rightarrow A \Rightarrow O_j \Rightarrow A
  \]
  since, for any other map \(f\), \([f]\) is uniquely determined by functoriality and preservation of finite products.
Models (ctd)

- Any theory defines a (necessarily unique) monad whose algebras are the same as the models of the theory.
- This monad is finitary.
- Every finitary monad corresponds to exactly one theory in this way.
Comodels

- A comodel of a theory \((\mathbb{L}, L : \mathbb{F}^{\text{op}} \to \mathbb{L})\) is given by a functor
  \[
  \langle\langle - \rangle\rangle : \mathbb{L}^{\text{op}} \to \text{Set}
  \]
  that preserves the finite coproducts of \(\mathbb{L}^{\text{op}}\).
- To give a comodel, it suffices to specify a set
  \[
  C = \langle\langle 1 \rangle\rangle
  \]
  since \(\langle\langle X \rangle\rangle = \langle\langle \bigsqcup_{x \in X} 1 \rangle\rangle \cong \bigsqcup_{x \in X} \langle\langle 1 \rangle\rangle = \langle\langle 1 \rangle\rangle \times X\), together with functions
  \[
  \text{op}_j = \langle\langle \text{op}n_j, \text{ops}_j \rangle\rangle = \langle\langle \text{OP}_j \rangle\rangle : C \times O_j \to C \times I_j
  \]
- A theory also defines a (unique) comonad whose coalgebras are the same as comodels of the theory. This comonad is generally not finitary. Also one comonad can correspond to many theories.
Interactive I/O

- An *algebra* (*model* of the appropriate theory) is a set $A$ with functions $\text{inp}$ and $\text{outp}$

  ![Diagram]

  $I \Rightarrow A$ \hspace{1cm} $A$
  $\downarrow \text{inp} \hspace{1cm} \downarrow \text{outp}$
  $A \hspace{1cm} O \Rightarrow A$

- The *monad* is $T X = \mu Z. X + (I \Rightarrow Z) + O \times Z$. This is the free monad on the functor $F$ defined by $F X = (I \Rightarrow X) + O \times X$. 
Interactive I/O ctd

- A coalgebra (comodel for the same theory) is a set $C$ with two functions $inp$ and $outp$

$$
\begin{array}{ccc}
C \times I & \xrightarrow{\text{inp}} & C \\
| & & | \\
\uparrow & & \uparrow \\
C & \xrightarrow{\text{outp}} & C \times O
\end{array}
$$

- The comonad is by $D X = \nu Z. X \times (Z \times I) \times (O \Rightarrow Z)$. It is the cofree comonad on the functor $G$ defined by $G X = (X \times I) \times (O \Rightarrow X)$. 
State

- A algebra is a set $A$ with two functions $lkp$ and $upd$ such that

$$\begin{align*}
S &\Rightarrow A \\
A &\Rightarrow S \Rightarrow A \\
A &\Rightarrow S \Rightarrow A \\
S &\Rightarrow A \Rightarrow S \Rightarrow (S \Rightarrow A) \Rightarrow S \times S \Rightarrow A
\end{align*}$$

- The monad is $T X = T_0 X / \sim_X$ where

$$T_0 X = \mu Z. X + (S \Rightarrow Z) + S \times Z$$

(the free monad on $F X = (S \Rightarrow X) + S \times X$)

- $T X \cong S \Rightarrow S \times X$
A coalgebra is a set $C$ with functions $\text{lkp}$ and $\text{upd}$ such that

$$
\begin{align*}
\text{C} \times \text{S} & \xrightarrow{\text{lkp}} \text{C} \quad \text{C} \quad \text{C} \xleftarrow{\text{upd}} \text{C} \times \text{S} \\
\text{C} & \xrightarrow{\text{upd}} \text{C} \times \text{S} \\
\text{C} \times \text{S} & \xrightarrow{\text{lkp}} \text{C} \quad \text{C} \leftarrow \text{C} \times \text{S} \\
\text{C} \times \text{S} & \leftarrow (\text{C} \times \text{S}) \times \text{S} \\
\text{C} \times \text{S} & \xrightarrow{\text{upd}} \text{C} \times \text{S} \\
\text{C} \times \text{S} & \leftarrow \text{C} \times (\text{S} \times \text{S}) \\
\end{align*}
$$
Alternatively, we can say that a coalgebra is a set with functions \( lkps \) and \( upd \) such that

\[
\begin{align*}
S & \xrightarrow{lkps} C \\
C & \xrightarrow{upd} C \times S \\
C \times S & \xleftarrow{\langle id, lkps \rangle} C \\
C \times S & \xleftarrow{upd} C \\
C \times S & \xleftarrow{upd \times S} (C \times S) \times S \\
C \times S & \xleftarrow{C \times snd} (C \times S) \times S \\
\end{align*}
\]

The comonad is \( DX = D_0 X \mid ok_X \) where

\[
D_0 X = \nu Z. X \times (Z \times S) \times (S \Rightarrow Z) \quad \text{(the cofree comonad on } G X = (X \times S) \times (S \Rightarrow X))
\]

\( DX \cong S \times (S \Rightarrow X) \)
Nondeterminism (1)

- An *algebra* is a set $A$ equipped with a function $\text{ch}$

  $$
  \begin{array}{ccc}
  A \times A & \xrightarrow{\text{ch}} & A
  \end{array}
  $$

- The monad is $T X = \mu Z. X + Z \times Z$
  (finite binary leaf trees).

  This is the free monad on $F X = X \times X$. 
A coalgebra is a set $C$ with a function $ch$

$$
\begin{array}{c}
C + C \\
\uparrow ch \\
C
\end{array}
$$

The comonad is

$$D X = \nu Z.X \times (Z + Z) \cong \nu Z.X \times (Z \times 2)$$

(streams over $X \times 2$).

This is the cofree comonad on $G X = X + X \cong X \times 2$. 
Nondeterminism (2)

- An *algebra* is a set $A$ equipped with a function $ch$ such that

\[
\begin{align*}
A \times A & \quad (A \times A) \times A \quad \xrightarrow{\quad} \quad A \times (A \times A) \quad \xrightarrow{\quad A \times ch \quad} \quad A \times A \\
\downarrow ch & \quad \downarrow ch \times A & \quad \downarrow A \times ch & \quad \downarrow ch \\
A & \quad A \times A & \quad A & \quad A
\end{align*}
\]

- The monad is $T X = T_0 X / \sim_X$ where
  - $T_0 X = \mu Z. X + Z \times Z$ (finite binary leaf trees)
  - $\sim_X$ relates those trees that flatten to the same nonempty list
  - $T X \cong X^+ = \mu Z. X \times (1 + Z)$ (non-empty lists)
Nondeterminism (2) ctd

- A coalgebra is a set \( C \) with a function \( ch \) satisfying

\[
\begin{align*}
C + C & \quad (C + C) + C & \quad C + (C + C) & \quad C + C \\
\quad & \quad \quad \quad & \quad \quad \quad & \quad \quad \quad \\
\quad & \quad \quad \quad & \quad \quad \quad & \quad \quad \quad \\
\quad & \quad \quad \quad & \quad \quad \quad & \quad \quad \quad \\
C & \quad \quad C + C & \quad \quad C & \quad \quad C
\end{align*}
\]

- The comonad is \( DX = D_0 X \mid ok_X \) where
  - \( D_0 X = \nu Z. X \times (Z + Z) \cong \nu Z. X \times (Z \times 2) \) (streams over \( X \times 2 \))
  - \( ok_X \) restricts \( D_0 X \) to streams where the bits are all zeros or all ones and all values except the head value are the same
  - \( DX \cong X \times (X \times 2) \).
Nondeterminism (3)

- An algebra must also satisfy

  \[ A \xrightarrow{\Delta} A \times A \xrightarrow{ch} A \]

- The monad is \( T X \cong X^+ \mid \text{squarefree} \) (nonempty lists with no sublist twice in a row).

- A coalgebra must also satisfy

  \[ C \leftarrow C + C \xrightarrow{\triangledown} C \]

- The comonad is \( D X \cong X \times 2 \).
Nondeterminism (4)

- An algebra must also satisfy

\[ A \times A \xrightarrow{\sigma^x} A \times A \]

- The monad is \( T X \cong X^+ / \text{perm} \) (nonempty lists modulo permutations, i.e., nonempty multisets).

- A coalgebra must also satisfy

\[ C + C \xleftarrow{\sigma^+} C + C \]

- The comonad is \( D X \cong 0 \).
Handling vs running

- Handling: Given a set $B$, a map $f : A \to B$ and an algebra structure $(B, g : T B \to B)$, the handler is defined as the unique map $h : TA \to B$ such that

$$
\begin{align*}
A & \xrightarrow{\eta_A} TA & \xleftarrow{\mu_A} & T(TA) \\
& \downarrow f & \downarrow h & \downarrow T h \\
B & \xleftarrow{g} & T B
\end{align*}
$$

- Running: Given a coalgebra $(C, \gamma : C \to D C)$, the runner $\theta$ is a unique natural transformation such that

$$
\begin{align*}
X & \xrightarrow{\eta_X} TX & \xleftarrow{\mu_X} & T(TX) \\
& \downarrow \eta^C_X & \downarrow \theta_X & \downarrow T \theta_X \\
T^C X & \xleftarrow{\mu^C_X} & T^C(T^C X) & \xrightarrow{\theta_{T^C X}} & T(T^C X)
\end{align*}
$$

where $(T^C, \eta^C, \mu^C)$ is the state monad for $C$. 
Stateful running vs running with continuations

- Given a monad \((T, \eta, \mu)\), for any set \(R\), monad morphisms

  \[ \theta_X : T X \to (X \Rightarrow R) \Rightarrow R \]

  between \((T, \eta, \mu)\) and the continuations monad for \(R\) are in a bijective correspondence with monad-algebra structures

  \[ \alpha : T R \to R \]

  of \((T, \eta, \mu)\) on \(R\)

- For \((R, \alpha) = (T X, \mu_X)\), \(\theta_X \eta_X = \text{id}_{T X}\).