Containers

Containers: What and why?

- Containers (Abbott, Altenkirch, Ghani, McBride) are a representation for a large class of collection types (set functors) and polymorphic functions (natural transformations) between them.
- All polymorphic functions between collection types with a container representation are uniquely represented as container maps.
- Many constructions on collection types can be done on the level of containers.
- Hence containers and container maps make a useful syntax for representing, manipulating, reasoning about collection types and polymorphic functions.

Containers, interpretation into set functors

- A container is given by

  $S : \mathbf{Set}$ (shapes)
  $P : S \to \mathbf{Set}$ (positions)

- It interprets into a set functor $[\![S, P]\!]^{\mathrm{c}} = F$ by

  $F : \mathbf{Set} \to \mathbf{Set}$
  $F\,X = \Sigma s : S.\, P\ s \to X$

  $F : \forall\{X, Y\}.\,(X \to Y) \to F\,X \to F\,Y$
  $\quad \forall\{X, Y\}.\,(X \to Y)$
  $\qquad \to (\Sigma s : S.\, P\,s \to X) \to \Sigma s' : S.\, P\,s' \to Y$
  $F\,f = \lambda(s, v).\,(s, \lambda p.\, f\,(v\,p))$

Container morphisms, interp. to nat. transfs.

- A container morphism between (S, P) and (S', P') is given by

  $$t : S \to S'$$
  $$q : \Pi\{s : S\}.\, P'\,(t\,s) \to P\,s$$

- It interprets into a nat. transf. $[\![t, q]\!]^{\mathrm{c}} = \tau$ between $[\![S, P]\!]^{\mathrm{c}} = F$ and $[\![S', P']\!]^{\mathrm{c}} = G$ by

  $$\tau : \forall X.\, F\,X \to G\,X$$
  $$\quad \forall\{X\}.\, (\Sigma s : S.\, P\,s \to X) \to \Sigma s' : S'.\, P'\,s' \to X$$
  $$\tau\,(s, v) = (t\,s, \lambda p.\, v\,(q\,\{s\}\,p))$$

Lists and list reversal

$$F\,X = \mathsf{List}\,X \qquad \cong \Sigma s : \mathsf{Nat}.\,[0..s) \to X$$

$$S = \mathsf{Nat}$$
$$P\,s = [0..s)$$

$$\tau : \forall\{X\}.\,\mathsf{List}\,X \to \mathsf{List}\,X$$
$$\tau = \mathsf{reverse}$$

$$t : \mathsf{Nat} \to \mathsf{Nat}$$
$$t\,s = s$$
$$q : \Pi\{s : \mathsf{Nat}\}.\,[0..t\,s) \to [0..s)$$
$$q\,\{s\}\,p = s - p$$

Identity container, composition of containers

- For $S = 1$, $P * = 1$, we have $[\![S, P]\!]^{c}\, X = \Sigma s : 1.\ 1 \to X \cong X$.
- Given (S_0, P_0), (S_1, P_1), for

$$\begin{array}{rcl} S & = & \Sigma s : S_0.\ P_0\, s \to S_1 \\ P\,(s, v) & = & \Sigma p : P_0\, s.\ P_1\,(vp) \end{array}$$

we have

$$\begin{array}{rl} & [\![S, P]\!]^{c}\, X \\ = & \Sigma(s_0, v) : (\Sigma s : S_0 . P_0\, s \to S_1).\ (\Sigma p : P_0\, s_0.\ P_1\,(v\, p)) \to X \\ \cong & \Sigma s_0 : S_0 . \Sigma v : (P_0\, s_0 \to S_1) . \Pi p : P_0\, s_0.\ P_1\,(v\, p) \to X \\ \cong & \Sigma s_0 : S_0.\ P_0\, s_0 \to \Sigma s_1 : S_1.\ P_1\, s_1 \to X \\ = & [\![S_0, P_0]\!]^{c}\,([\![S_1, P_1]\!]^{c}\, X) \end{array}$$

A monoidal category, monoidal functor

- Containers and container morphisms form a monoidal category **Cont**.
- Interpretation $[\![-]\!]^{c}$ of containers and container morphisms into set functors (and natural transformations) is a fully-faithful monoidal functor.

$$
\begin{array}{ccc}
\mathbf{Cont} & & \text{mon.} \\
\downarrow [\![-]\!]^{c} & & \text{mon., f.f.} \\
[\mathbf{Set}, \mathbf{Set}] & & \text{strct.mon.}
\end{array}
$$

Containers $\cap$ monads

Monadic containers

- A monadic container is given by a container (S, P) with
  - $\mathrm{e} : S$
  - $\bullet : \Pi s : S.\,(P\ s \to S) \to S$
  - $q_0 : \Pi\{s : S\}.\,\Pi v : P\,s \to S.\,P\,(s \bullet v) \to P\,s$
  - $q_1 : \Pi s : S.\,\Pi\{v : P\,s \to S\}.\,\Pi p : P\,(s \bullet v).\,P\,(v\,(v \nwarrow_s p))$

  where we write
  - $q_0\,\{s\}\,v\,p$ as $v \nwarrow_s p$
  - $q_1\,s\,\{v\}\,p$ as $p \nearrow_v s$

  such that
  - $s \bullet (\lambda\_.\,\mathrm{e}) = s$
  - $\mathrm{e} \bullet (\lambda\_.\,s) = s$
  - $(s \bullet v) \bullet (\lambda p''.\,w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)) = s \bullet (\lambda p'.\,v\,p' \bullet w\,p')$

  and ...

- ...
  - $(\lambda\_.\,\mathrm{e}) \nwarrow_s p = p$
  - $p \nearrow_{\lambda\_.\,s} \mathrm{e} = p$
  - $v \nwarrow_s ((\lambda p''.\, w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)) \nwarrow_{s \bullet v} p) = (\lambda p'.\, v\, p' \bullet w\, p') \nwarrow_s p$
  - $((\lambda p''.\, w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)) \nwarrow_{s \bullet v} p) \nearrow_v s = \mathrm{let}\ p_0 \leftarrow (\lambda p'.\, v\, p' \bullet w\, p') \nwarrow_s p\ \mathrm{in}\ w\, p_0 \nwarrow_{v\, p_0} (p \nearrow_{\lambda p'.\, v\, p' \bullet w\, p'} s)$
  - $p \nearrow_{\lambda p''.\, w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)} (s \bullet v) = \mathrm{let}\ p_0 \leftarrow (\lambda p'.\, v\, p' \bullet w\, p') \nwarrow_s p\ \mathrm{in}\ (p \nearrow_{\lambda p'.\, v\, p' \bullet w\, p'} s) \nearrow_{w\, p_0} v\, p_0$
- Laws 1-3 resemble those of monoid, laws 4-8 those of a biaction.

Interpretation into monads

- A monadic container interprets to a monad $[\![S, P, \mathsf{e}, \bullet, \nwarrow, \nearrow]\!]^{\mathrm{mc}} = (T, \eta, \mu)$ where

$$
\begin{aligned}
T\,X &= [\![S, P]\!]^{\mathrm{c}}\,X \\
T\,f &= [\![S, P]\!]^{\mathrm{c}}\,f
\end{aligned}
$$

$$
\begin{aligned}
&\eta : \forall\{X\}.\, X \to T\,X \\
&\quad \forall\{X\}.\, X \to \Sigma s : S.\, P\,s \to X \\
&\eta\,x = (\mathsf{e}, \lambda\_.\, x)
\end{aligned}
$$

$$
\begin{aligned}
&\mu : \forall\{X\}.\, T\,(T\,X) \to T\,X \\
&\quad \forall\{X\}.\, (\Sigma s : S.\, P\,s \to \Sigma s' : S.\, P\,s' \to X) \to \Sigma s : S.\, P\,s \to X \\
&\mu\,(s, v) = \mathsf{let}\ v_0\,p = \mathsf{fst}\,(v\,p) \\
&\qquad\qquad\qquad\ \ v_1\,p = \mathsf{snd}\,(v\,p) \\
&\qquad\qquad\ \ \mathsf{in}\ (s \bullet v_0, \lambda p.\, v_1\,(v_0 \nwarrow_s p)\,(p \nearrow_{v_0} s))
\end{aligned}
$$

Monadic containers, interpretation into monads (ctd)

- Monadic containers form a category **MCont**.
- $[\![-]\!]^{\mathrm{mc}}$ forms a fully faithful functor from **MCont** to **Monad**(**Set**).
- $[\![-]\!]^{\mathrm{mc}}$ is the pullback of $[\![-]\!]^{\mathrm{c}} : \mathbf{Cont} \to [\mathbf{Set}, \mathbf{Set}]$ along $U : \mathbf{Monad}(\mathbf{Set}) \to [\mathbf{Set}, \mathbf{Set}]$.

$$
\begin{array}{ccccc}
\begin{array}{c}\mathbf{MCont}\\ \cong \mathbf{Monoid}(\mathbf{Cont})\end{array} & \xrightarrow{U} & \mathbf{Cont} & & \text{mon.} \\
\text{f.f.} \quad [\![-]\!]^{\mathrm{mc}} \big\downarrow & & \big\downarrow [\![-]\!]^{\mathrm{c}} & & \text{mon., f.f.} \\
\begin{array}{c}\mathbf{Monad}(\mathbf{Set})\\ \cong \mathbf{Monoid}([\mathbf{Set}, \mathbf{Set}])\end{array} & \xrightarrow{U} & [\mathbf{Set}, \mathbf{Set}] & & \text{str. mon.}
\end{array}
$$

List monad

$$T\,X = \mathsf{List}\,X$$
$$\eta\,x = [x]$$
$$\mu\,xss = \mathsf{concat}\,xss$$

$$S = \mathsf{Nat}$$
$$P\,s = [0..s)$$
$$\mathsf{e} = 1$$
$$s \bullet v = \textstyle\sum_{p:[0..s)} v\,p$$
$$v \nwarrow_s p = \text{greatest } p_0 : [0..s) \text{ such that } \textstyle\sum_{p':[0..p_0)} v\,p' \leq p$$
$$p \nearrow_v s = p - \textstyle\sum_{p':[0..v \nwarrow_s p)} v\,p'.$$

Reader monads

$U : \mathsf{Set}$

$$T\,X = U \to X \qquad \cong 1 \times (U \to X)$$
$$\eta\,x = \lambda u.\,x$$
$$\mu\,f = \lambda u.\,f\;u\;u$$

$$S = 1$$
$$P\,* = U$$
$$\mathsf{e} = *$$
$$* \bullet (\lambda\_.\,*) = *$$
$$(\lambda\_.\,*) \nwarrow_{*} p = p$$
$$p \nearrow_{\lambda\_.\,*} * = p$$

Writer monads

$(V, \mathrm{o}, \oplus)$: Monoid

$$T\,X = V \times X \qquad\qquad \cong V \times (1 \to X)$$
$$\eta\,x = (\mathrm{o}, x)$$
$$\mu\,(p, (p', x)) = (p \oplus p', x)$$

$$S = V$$
$$P\,\_ = 1$$
$$\mathrm{e} = \mathrm{o}$$
$$s \bullet (\lambda{*}.\, s') = s \oplus s'$$
$$(\lambda{*}.\, s') \nwarrow_s * = *$$
$$* \nearrow_{\lambda *.\, s'} s = *$$

State monads

$U : \mathrm{Set}$

$$
\begin{aligned}
T\,X &= U \to U \times X \qquad\qquad \cong (U \to U) \times (U \to X) \\
\eta\,x &= \lambda u.\,(u, x) \\
\mu\,f &= \lambda u.\,\mathrm{let}\ (u', g) \leftarrow f\ u'\ \mathrm{in}\ g\ u'
\end{aligned}
$$

$$
\begin{aligned}
&S = U \to U \\
&P\ \_ = U \\
&\mathrm{e} = \lambda p.\,p \\
&s \bullet v = \lambda p.\,v\,p\,(s\,p) \\
&v \nwarrow_s p = p \\
&p \nearrow_v s = s\,p
\end{aligned}
$$

Update monads

$(V, \mathrm{o}, \oplus)$: Monoid
$(U, \downarrow)$: $(V, \mathrm{o}, \oplus)$-Set

$$T\,X = U \to V \times X \qquad \cong (U \to V) \times (U \to X)$$
$$\eta\,x = \lambda u.(\mathrm{o}, x)$$
$$\mu\,f = \lambda u.\,\mathrm{let}\ (p, g) \leftarrow f\ u; (p', x) \leftarrow g\,(u \downarrow p)\ \mathrm{in}\ (p \oplus p', x)$$

$$S = U \to V$$
$$P\ \_ = U$$
$$\mathrm{e} = \lambda\_.\,\mathrm{o}$$
$$s \bullet v = \lambda p.\, s\,p \oplus v\,p\,(s\,p)$$
$$v \nwarrow_s p = p$$
$$p \nearrow_v s = p \downarrow s\,p$$

Algebras of monadic containers

- An algebra of the monad $[\![S, P, \mathsf{e}, \bullet, \nwarrow, \nearrow]\!]^{\mathrm{mc}}$ is given by

  $$X : \mathsf{Set}$$
  $$* : \Pi s : S.\, (P\, s \to X) \to X$$

  such that

  $$\mathsf{e} * (\lambda\_.\, x) = x$$
  $$(s \bullet v) * (\lambda p''.\, w\, (v \nwarrow_s p'')\, (p'' \nearrow_v s))$$
  $$= s * (\lambda p'.\, v\, p' * w\, p')$$

- I.e., an algebra is a set with S many operations, with $P\ s$ the arity of the operation for s.