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

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

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

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 $[\![S, P]\!]^{\mathrm{c}}\, X = \Sigma s : 1.\ 1 \to X \cong X$.
- 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

$$
\begin{array}{ll}
\multicolumn{2}{l}{[\![S, P]\!]^{\mathrm{c}}\, X} \\
= & \Sigma(s_0, v) : (\Sigma s : S_0 . P_0\, s \to S_1).\ (\Sigma p : P_0\, s_0.\ P_1\,(v\, p)) \to X \\
\cong & \Sigma s_0 : S_0 . \Sigma v : (P_0\, s_0 \to S_1) . \Pi p : P_0\, s_0.\ P_1\,(v\, p) \to X \\
\cong & \Sigma s_0 : S_0.\ P_0\, s_0 \to \Sigma s_1 : S_1.\ P_1\, s_1 \to X \\
= & [\![S_0, P_0]\!]^{\mathrm{c}}\,([\![S_1, P_1]\!]^{\mathrm{c}}\, X)
\end{array}
$$

A monoidal category, monoidal functor

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

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

Containers $\cap$ monads

Monadic containers

A monadic container is given by

$$
\begin{aligned}
&S : \mathbf{Set} \\
&P : S \to \mathbf{Set} \\
&\mathrm{e} : S \\
&\bullet : \Pi s : S.\, (P\ s \to S) \to S \\
&\nwarrow : \Pi s : S.\, \Pi v : P\, s \to S.\, P\,(s \bullet v) \to P\, s \\
&\nearrow : \Pi s : S.\, \Pi v : P\, s \to S.\, \Pi p : P\,(s \bullet v).\, P\,(v\,(v \nwarrow \{s\}\, p))
\end{aligned}
$$

such that

$$
\begin{aligned}
&s \bullet (\lambda\_.\, \mathrm{e}) = s \\
&(\lambda\_.\, \mathrm{e}) \nwarrow \{s\}\, p = p \\
&\mathrm{e} \bullet (\lambda\_.\, s) = s \\
&(\lambda\_.\, s) \nearrow \{\mathrm{e}\}\, p = p \\
&(s \bullet v) \bullet (\lambda p.\, w\,(v \nwarrow \{s\}\, p)\,(v \nearrow \{s\}\, p)) \\
&\quad = s \bullet (\lambda p.\, v\, p \bullet (\lambda p'.\, w\, p\, p'))
\end{aligned}
$$

...

Interpretation into monads

It interprets to a monad $[\![S, P, \mathrm{e}, \bullet, \nwarrow, \nearrow]\!]^{\mathrm{mc}} = (T, \eta, \mu)$ via

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

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

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

List monad

$$
\begin{aligned}
T\,X &= \mathrm{List}\,X \\
\eta\,x &= [x] \\
\mu\,xss &= \mathrm{concat}\,xss
\end{aligned}
$$

$$
\begin{aligned}
S &= \mathrm{Nat} \\
P\,s &= [0..s) \\
\mathrm{e} &= 1 \\
s \bullet v &= \textstyle\sum_{p:[0..s)} v\,p \\
v \nwarrow \{s\}\,p &= \text{greatest } p_0 : [0..s) \text{ such that } \textstyle\sum_{p':[0..p_0)} v\,p' \leq p \\
v \nearrow \{s\}\,p &= p - \textstyle\sum_{p':[0..v \nwarrow \{s\}\,p)} v\,p'.
\end{aligned}
$$

Monadic containers, interpretation into monads (ctd)

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

$$\begin{tikzcd}
\begin{array}{c}\mathbf{MCont} \\ \cong \mathbf{Monoid}(\mathbf{Cont})\end{array} \arrow[r, "U"] \arrow[d, "{[\![-]\!]^{\mathrm{mc}}}"', "\text{f.f.}"'] & \mathbf{Cont} \quad \text{mon.} \arrow[d, "{[\![-]\!]^{\mathrm{c}} \quad \text{mon., f.f.}}"] \\
\begin{array}{c}\mathbf{Monad}(\mathbf{Set}) \\ \cong \mathbf{Monoid}([\mathbf{Set}, \mathbf{Set}])\end{array} \arrow[r, "U"] & {[\mathbf{Set}, \mathbf{Set}]} \quad \text{str. mon.}
\end{tikzcd}$$

Reader monads

U : **Set**

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

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

Writer monads

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

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

$$
\begin{aligned}
S &= V \\
P\ {}_{\_} &= 1 \\
\mathrm{e} &= \mathrm{o} \\
s \bullet (\lambda *.\, s') &= s \oplus s' \\
(\lambda *.\, s') \nwarrow \{s\}\, * &= * \\
(\lambda *.\, s') \nearrow \{s\}\, * &= *
\end{aligned}
$$

State monads

$U : \mathbf{Set}$

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

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

Update monads

$(V, \mathsf{o}, \oplus) : \mathsf{Monoid}$
$(U, \downarrow) : \mathsf{Act}(V, \mathsf{o}, \oplus)$

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

$$\begin{aligned}
S &= U \to V \\
P\ _ &= U \\
\mathsf{e} &= \lambda_.\,\mathsf{o} \\
s \bullet v &= \lambda p.\, s\,p \oplus v\,p\,(s\,p) \\
v \nwarrow \{s\}\,p &= p \\
v \nearrow \{s\}\,p &= p \downarrow s\,p
\end{aligned}$$

Algebras of monadic containers

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

$$
\begin{aligned}
&X : \mathbf{Set} \\
&* : \Pi s : S.\, (P\, s \to X) \to X
\end{aligned}
$$

such that

$$
\begin{aligned}
&\mathsf{e} * (\lambda p.\, x) = x \\
&(s \bullet v) * (\lambda p.\, w\, (v \nwarrow \{s\}\, p)\, (v \nearrow \{s\}\, p)) \\
&\quad = s * (\lambda p.\, v\, p * (\lambda p'.\, w\, p\, p'))
\end{aligned}
$$

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