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

$$S : \mathbf{Set} \text{ (shapes)}$$

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

ullet It interprets into a set functor $[\![S,P]\!]^c=F$ by

$$F : \mathbf{Set} \to \mathbf{Set}$$

$$F : X = \Sigma s : S. P s \to X$$

$$F : \forall \{X, Y\}. (X \to Y) \to F X \to F Y$$

$$\forall \{X, Y\}. (X \to Y)$$

$$\to (\Sigma s : S. P s \to X) \to \Sigma s' : S. P s' \to Y$$

$$F f = \lambda(s, v). (s, \lambda p. f (v p))$$

Container morphisms, interp. to nat. transfs.

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

$$t: S \rightarrow S'$$

 $q: \Pi\{s: S\}. P'(ts) \rightarrow Ps$

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

$$\tau: \forall X. F X \to G X$$

$$\forall \{X\}. (\Sigma s: S. P s \to X) \to \Sigma s': S'. P' s' \to X$$

$$\tau(s, v) = (t s, \lambda p. v (q \{s\} p))$$

Lists and list reversal

$$FX = \operatorname{List} X$$
 $\cong \Sigma s : \operatorname{Nat.} [0..s) \to X$ $S = \operatorname{Nat} P s = [0..s)$

$$au: orall \{X\}. \operatorname{List} X o \operatorname{List} X$$
 $au = \operatorname{reverse}$
 $t: \operatorname{Nat} o \operatorname{Nat}$
 $ts = s$
 $q: \Pi\{s: \operatorname{Nat}\}. [0..ts) o [0..s)$
 $q\{s\} p = s - p$

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

$$S = \Sigma s : S_0. P_0 s \rightarrow S_1$$

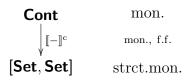
$$P(s, v) = \Sigma p : P_0 s. P_1 (vp)$$

we have

$$\begin{aligned}
& [S, P]^{c} X \\
&= \Sigma(s_{0}, v) : (\Sigma s : S_{0}.P_{0} s \to S_{1}). (\Sigma p : P_{0} s_{0}. P_{1} (v p)) \to X \\
&\cong \Sigma s_{0} : S_{0}.\Sigma v : (P_{0} s_{0} \to S_{1}).\Pi p : P_{0} s_{0}. P_{1} (v p) \to X \\
&\cong \Sigma s_{0} : S_{0}.P_{0} s_{0} \to \Sigma s_{1} : S_{1}.P_{1} s_{1} \to X \\
&= [S_{0}, P_{0}]^{c} ([S_{1}, P_{1}]^{c} X)
\end{aligned}$$

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.



Containers ∩ monads

Monadic containers

A monadic container is given by

```
S:\mathbf{Set}
P:S\to\mathbf{Set}
e \cdot S
• : \Pi s : S. (P s \rightarrow S) \rightarrow S
\uparrow: \sqcap s: S. \sqcap v: Ps \rightarrow S. P(s \bullet v) \rightarrow Ps
\uparrow: \Pi s: S. \Pi v: Ps \rightarrow S. \Pi p: P(s \bullet v). P(v(v \setminus \{s\} p))
such that
s \bullet (\lambda_{-}.e) = s
(\lambda_{-}, e) \setminus \{s\} p = p
e \bullet (\lambda_{-}.s) = s
(\lambda_{-}, s) \uparrow \{e\} p = p
(s \bullet v) \bullet (\lambda p. w (v \setminus \{s\} p) (v \setminus \{s\} p))
    = s \bullet (\lambda p. v p \bullet (\lambda p'. w p p'))
```

Interpretation into monads

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

$$TX = [S, P]^{c} X$$

$$Tf = [S, P]^{c} f$$

$$\eta : \forall \{X\}. X \to TX$$

$$\forall \{X\}. X \to \Sigma s : S. P s \to X$$

$$\eta x = (e, \lambda p. x)$$

$$\mu : \forall \{X\}. T(TX) \to TX$$

$$\forall \{X\}. (\Sigma s : S. P s \to \Sigma s' : S. P s' \to X) \to (\Sigma s : S. P s \to X)$$

$$\mu(s, v) = \text{let } v_{0} p = \text{fst } (v p)$$

$$v_{1} p = \text{snd } (v p)$$

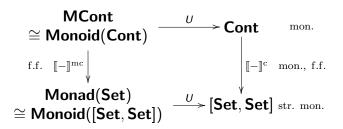
$$\text{in } (s \bullet v_{0}, \lambda p. v_{1} (v_{0} \land \{s\} p) (v_{0} \land \{s\} p))$$

List monad

```
TX = \text{List } X
\eta x = [x]
\mu xss = \operatorname{concat} xss
S = Nat
P s = [0..s)
e = 1
s \bullet v = \sum_{p:[0..s)} v p
v \uparrow \{s\} \, p = 	ext{greatest } p_0 : [0..s) 	ext{ such that } \sum_{p':[0..p_0)} v \, p' \leq p
v 
ewline \{s\} p = p - \sum_{p':[0..v \setminus \{s\},p)} v p'.
```

Monadic containers, interpretation into monads (ctd)

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



Reader monads

 $U:\mathbf{Set}$

$$TX = U \to X$$

$$\eta x = \lambda u. x$$

$$\mu f = \lambda u. f u u$$

$$S = 1$$

$$P *= U$$

$$e = *$$

$$* \bullet (\lambda_{-}. *) \uparrow \{*\} p = p$$

$$(\lambda_{-}. *) f \{*\} p = p$$

Writer monads

$$(V, o, \oplus)$$
: Monoid

 $TX = V \times X \qquad \cong V \times (1 \to X)$
 $\eta x = (o, x)$
 $\mu (p, (p', x)) = (p \oplus p', x)$
 $S = V$
 $P_- = 1$
 $e = o$
 $s \bullet (\lambda *. s') = s \oplus s'$
 $(\lambda *. s') \uparrow \{s\} * = *$
 $(\lambda *. s') \uparrow \{s\} * = *$

State monads

U: **Set**

$$TX = U \rightarrow U \times X \qquad \cong (U \rightarrow U) \times (U \rightarrow X)$$

$$\eta x = \lambda u.(u,x)$$

$$\mu f = \lambda u. \text{ let } (u',g) \leftarrow f u'; (u'',x) \leftarrow g u' \text{ in } (u'',x)$$

$$S = U \rightarrow U$$

$$P_{-} = U$$

$$e = \lambda p. p$$

$$s \bullet v = \lambda p. v p(s p)$$

$$v \uparrow \{s\} p = p$$

$$v \uparrow \{s\} p = s p$$

Update monads

```
(V, o, \oplus): Monoid
(U,\downarrow): Act(V,o,\oplus)
TX = U \rightarrow V \times X
                                                   \cong (U \to V) \times (U \to X)
\eta x = \lambda u.(o, x)
\mu f = \lambda u. let (p,g) \leftarrow f u; (p',x) \leftarrow g (u \downarrow p) in (p \oplus p',x)
S = U \rightarrow V
P = U
e = \lambda o
s \bullet v = \lambda p. s p \oplus v p (s p)
v \setminus \{s\} p = p
v \upharpoonright \{s\} p = p \downarrow s p
```

Algebras of monadic containers

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

X: **Set** $*: \Pi s: S. (Ps \rightarrow X) \rightarrow X$

such that

$$e * (\lambda p. x) = x$$

$$(s \bullet v) * (\lambda p. w (v \land \{s\} p) (v \land \{s\} p))$$

$$= s * (\lambda p. v p * (\lambda p'. w p p'))$$

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