Container combinatorics: Monads and more

Tarmo Uustalu, Tallinn University of Technology

SYCO 1, Birmingham, 20–21 September 2018

Containers?

- Containers (Abbott, Altenkirch, Ghani; cf polynomials, Gambino, Hyland, Kock) are an elegant “syntax” in terms of shapes and positions for a wide class of set functors.
- In particular, they are good for enumerative combinatorics, for enumerating structures of a given type on a functor.
- Prior work: Directed containers (Ahman, Chapman, Uustalu) as containers with additional structure denoting comonads.
- This talk: Further specializations of containers corresponding to monads, lax monoidal functors (aka idioms) and more.

Containers

- A *container* is given by
  - a set S (of shapes)
  - and a S-indexed family P of sets (of positions in each shape)

- A container (S, P) interprets into a set functor $[\![S, P]\!]^{\mathrm{c}} = F$ where
  - $F\,X = \Sigma s : S.\, P\,s \to X$
  - $F\,f = \lambda(s, v).\,(s, f \circ v)$

Lists container

- Let
  - $S = \mathbb{N}$
  - $P\,s = [0..s)$
- The container (S, P) represents the list datatype, as
  - $$\begin{aligned} [\![S, P]\!]^{\mathrm{c}}\, X &= \Sigma s : \mathbb{N}.\, [0..s) \to X \\ &\cong \mathsf{List}\, X\,. \end{aligned}$$

Container morphisms

- A *container morphism* between (S, P) and (S', P') is given by operations
  - $t : S \to S'$ (the shape map)
  - and $q : \Pi_{s:S}.\, P'\,(t\,s) \to P\,s$ (the position map)

- A container morphism (t, q) between (S, P) and (S', P') interprets into a natural transformation $[\![t, q]\!]^{\mathrm{c}} = \tau$ between $[\![S, P]\!]^{\mathrm{c}}$ and $[\![S', P']\!]^{\mathrm{c}}$ where
  - $\tau_X : [\![S, P]\!]^{\mathrm{c}}\, X \to [\![S', P']\!]^{\mathrm{c}}\, X$
    $(\Sigma s : S.\, P\,s \to X) \to (\Sigma s' : S'.\, P'\,s' \to X)$

    $\tau\,(s, v) = (t\,s, v \circ q_s)$

Some lists container endomorphisms

- Let $S = \mathbb{N}$, $P\,s = [0..s)$ as before.
- We can define a container endomorphism (t, q) on (S, P) for example by
  - $t\,s = s$
  - $q_s\,p = s - p$

  This denotes the list reversal function.
- But setting
  - $t\,s = s + s$
  - $q_s\,p = p \bmod s$

  we get a representation of the list self-append function.

The category of containers

- Identity on (S, P) is $(\mathrm{id}_S, \lambda_s.\, \mathrm{id}_{P\,s})$.
- Composition of $(t, q) : (S, P) \to (S', P')$ and $(t', q') : (S', P') \to (S'', P'')$ is $(t' \circ t, \lambda_s.\, q_s \circ q'_{t\,s})$.

- Containers form a category **Cont**.

- $[\![-]\!]^{\mathrm{c}}$ makes a fully-faithful functor from **Cont** to $[\mathbf{Set}, \mathbf{Set}]$.

Two monoidal structures

- The identity container is $\mathsf{Id}^{\mathrm{c}} = (1, \lambda *. 1)$.
- Composition of (S, P) and (S', P') is $(S, P) \cdot^{\mathrm{c}} (S', P') = (\Sigma s : S. P\, s \to S', \lambda(s, v). \Sigma p : P\, s. P'\,(v\, p))$.
- $(\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \cdot^{\mathrm{c}})$ is a <u>monoidal</u> category and $[\![-]\!]^{\mathrm{c}}$ a <u>monoidal</u> functor to $([\mathbf{Set}, \mathbf{Set}], \mathsf{Id}, \cdot)$.

- Day convolution of (S, P) and (S', P') is $(S, P) \circledast^{\mathrm{c}} (S', P') = (S \times S', \lambda(s, s'). P\, s \times P'\, s)$.
- $(\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \circledast^{\mathrm{c}})$ is a <u>symmetric monoidal</u> category and $[\![-]\!]^{\mathrm{c}}$ a <u>symmetric monoidal</u> functor to $([\mathbf{Set}, \mathbf{Set}], \mathsf{Id}, \circledast)$.

- For any (S, P), (S', P'), there is a container morphism from $(S, P) \circledast^{\mathrm{c}} (S', P') \to (S, P) \cdot^{\mathrm{c}} (S', P')$.
- This makes $\mathsf{Id}_{\mathbf{Cont}}$ a <u>lax monoidal</u> functor from $(\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \cdot^{\mathrm{c}})$ to $(\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \circledast^{\mathrm{c}})$.

Mnd-containers

- Call an *mnd-container* a container (S, P) with operations
  - $\mathsf{e} : S$
  - $\bullet : \Pi s : S.\, (P\ s \to S) \to S$
  - $q_0 : \Pi s : S.\, \Pi v : P\, s \to S.\, P\,(s \bullet v) \to P\, s$
  - $q_1 : \Pi s : S.\, \Pi v : P\, s \to S.\, \Pi p : P\,(s \bullet v).\, P\,(v\,(v \nwarrow_s p))$

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

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

  and ...

Mnd-containers ctd

- ... and
  - $p = (\lambda\_.\, \mathrm{e}) \nwarrow_s \; p$
  - $p \nearrow_{\lambda\_.\, s} \mathrm{e} = p$
  - $v \nwarrow_s ((\lambda p''.\, w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)) \nwarrow_{s \bullet v} p) =$
    $(\lambda p'.\, v\, p' \bullet w\, p') \nwarrow_s p$
  - $((\lambda p''.\, w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)) \nwarrow_{s \bullet v} p) \nearrow_v s =$
    $\mathrm{let}\ u\, p' \leftarrow v\, p' \bullet w\, p'\ \mathrm{in}\ w\,(u \nwarrow_s p) \nwarrow_{v\,(u \nwarrow_s p)} (p \nearrow_u s)$
  - $p \nearrow_{\lambda p''.\, w\,(v \nwarrow_s p'')\,(p'' \nearrow_v s)} (s \bullet v) =$
    $\mathrm{let}\ u\, p' \leftarrow v\, p' \bullet w\, p'\ \mathrm{in}\ (p \nearrow_u s) \nearrow_{w\,(u \nwarrow_s p)} v\,(u \nwarrow_s p)$

Mnd-containers ctd

- An mnd-container $(S, P, \mathrm{e}, \bullet, \nwarrow, \nearrow)$ interprets into a <u>monad</u> $[\![S, P, \mathrm{e}, \bullet, \nwarrow, \nearrow]\!]^{\mathrm{mc}} = (T, \eta, \mu)$ where
  - $T = [\![S, P]\!]^{\mathrm{c}}$
  - $\eta_X : X \to T\,X$

    $X \to \Sigma s : S.\, P\,s \to X$

    $\eta\,x = (\mathrm{e}, \lambda\_.\,x)$
  - $\mu_X : T\,(T\,X) \to T\,X$

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

    $\mu\,(s, v) =$

    let $(v_0\,p, v_1\,p) \leftarrow v\,p$ in $(s \bullet v_0, \lambda p.\, v_1\,(v_0 \nwarrow_s p)\,(p \nearrow_{v_0} s))$

The category of mnd-containers

- Mnd-containers form a category **MCont**, with identities and composition inherited from **Cont**.
- Mnd-container interpretation $[\![-]\!]^{\mathrm{mc}}$ makes a <u>fully-faithful</u> functor between **MCont** and **Monad**(**Set**).

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

Exception container

- Let $S = 1 + E$ for some set E and $P\,(\mathsf{inl}\,*) = 1,\ P\,(\mathsf{inr}\,\_) = 0$.
- Then $T\,X =$
  $$\Sigma s : 1 + E. \left(\text{case } s \text{ of } \begin{array}{lcl} \mathsf{inl}\,* & \mapsto & 1 \\ \mathsf{inr}\,\_ & \mapsto & 0 \end{array} \right) \to X \;\cong\; X + E.$$
- If, in a hypotetical mnd-container structure on (S, P), $\mathsf{e} = \mathsf{inr}\,e_0$ for some $e_0 : E$, then $P\,\mathsf{e} = 0$ and therefore $\mathsf{inl}\,* = \mathsf{e} \bullet (\lambda\_.\,\mathsf{inl}\,*) = \mathsf{e} \bullet (\lambda\_.\,\mathsf{inr}\,e_0) = \mathsf{inr}\,e_0$, which is absurd.
- If $\mathsf{e} = \mathsf{inl}\,*$, then necessarily $\mathsf{inl}\,* \bullet v = \mathsf{e} \bullet (\lambda *.\, v\,*) = v\,*$ and $\mathsf{inr}\,e \bullet v = \mathsf{inr}\,e \bullet (\lambda\_.\,\mathsf{e}) = \mathsf{inr}\,e$.
- This choice of e and $\bullet$ satisfies the conditions of an mnd-container.
- So there is exactly one mnd-container structure on (S, P) and exactly one monad structure on T.

Lists container

- Let $S = \mathbb{N}$, $P\,s = [0..s)$.
- Then $T\,X = \Sigma s : \mathbb{N}.\,[0..s) \to X \;\cong\; \mathrm{List}\,X$.

- The following is an mnd-container structure:
  - $\mathrm{e} = 1$
  - $s \bullet v = \sum_{p:[0..s)} v\,p$
  - $v \nwarrow_s p =$ greatest $p_0 : [0..s)$ st $\sum_{p':[0..p_0)} v\,p' \leq p$
  - $p \nearrow_v s = p - \sum_{p':[0..v \nwarrow_s p)} v\,p'$
- The corresponding monad structure is $\eta_X\,x = [x]$, $\mu_X\,xss = \mathrm{concat}\,xss$.

- But these are not the only mnd-container structure on (S, P) and not the only monad structure on T.

Mnd-containers as generalized operads

- The (standard) lists mnd-container generalizes for non-symmetric operads.
- Given an operad, i.e., a set O (of operations) and functions $\# : O \to \mathbb{N}$ (fixing the arities) and $\mathrm{id} : O$ (the identity) and $\circ : \Pi o : O.\,(\#\, o \to O) \to O$ (composition) satisfying $\#\,\mathrm{id} = 1$ and $\#\,(o \circ v) = \sum_{i:[0,\#\, o)} \#\,(v\; i)$ and a number of further equations.
- We can take $S = O$, $P\, o = [0..\#\, o)$, $\mathrm{e} = \mathrm{id}$, $s \bullet v = s \circ v$ and $\nwarrow$, $\nearrow$ as in the lists mnd-container.
- The lists mnd-container corresponds to the operad Assoc with exactly one operation of every arity.
- General mnd-containers are like operads, but arities may be infinite, identification of the arguments of an operation is nominal, and the arguments of a composition may be used non-linearly by the operations involved (as specified by $\nwarrow$, $\nearrow$).

Mnd-containers as lax $(1, \Sigma)$-universes

- Altenkirch, Pinyo have observed that an mnd-container defines a "lax" $(1, \Sigma)$-universe.
  - S is the set of "(codes for) types",
  - $P\,s$ is the "denotation" of s,
  - e is the type 1,
  - $\bullet$ is the Σ-type former,
  - $\nwarrow$, $\nearrow$ are projections from denotations of Σ-types
- The laxity is that 1 need not really denote the singleton set and Σ-types need not really denote dependent products, we only have functions $P\,e \to 1$ and $P\,(s \bullet v) \to \Sigma p : P\,s.\, P\,(v\,p)$, not isomorphisms.

Lmf-containers

- Call an *lmf-container* a container (S, P) with operations
  - $\mathrm{e} : S$
  - $\bullet : S \to S \to S$
  - $q_0 : \Pi s : S. \Pi s' : S. P\,(s \bullet s') \to P\,s$
  - $q_1 : \Pi s : S. \Pi s' : S. P\,(s \bullet s') \to P\,s'$

  where we write
  - $q_0\,s\,s'\,p$ as $s' \nwarrow_s p$ and $q_1\,s\,s'\,p$ as $p \nearrow_{s'} s$

  satisfying
  - $\mathrm{e} \bullet s = s$
  - $s = s \bullet e$
  - $(s \bullet s') \bullet s'' = s \bullet (s' \bullet s'')$
  - $\mathrm{e} \nwarrow_s p = p$
  - $p \nearrow_s \mathrm{e} = p$
  - $s' \nwarrow_s (s'' \nwarrow_{s \bullet s'} p) = (s' \bullet s'') \nwarrow_s p$
  - $(s'' \nwarrow_{s \bullet s'} p) \nearrow_{s'} s = s'' \nwarrow_{s'} (p \nearrow_{s' \bullet s''} s)$
  - $p \nearrow_{s''} (s \bullet s') = (p \nearrow_{s' \bullet s''} s) \nearrow_{s''} s'$

Lmf-containers ctd

An lmf-container $(S, P, \mathrm{e}, \bullet, \nwarrow, \nearrow)$ interprets into a lax monoidal functor $[\![S, P, \mathrm{e}, \bullet, \nwarrow, \nearrow]\!]^{\mathrm{lc}} = (F, \mathrm{m}^0, \mathrm{m})$ where

- $F = [\![S, P]\!]^{\mathrm{c}}$
- $\mathrm{m}^0 : 1 \to T\,1$

  $1 \to (\Sigma s : S.\, P\,s \to 1)$

  $\mathrm{m}^0\,* = (\mathrm{e}, \lambda\_.\,*)$
- $\mathrm{m}_{X,Y} : T\,X \times T\,Y \to T\,(X \times Y)$

  $(\Sigma s : S.\, P\,s \to X) \times (\Sigma s : S.\, P\,s \to Y) \to (\Sigma s : S.\, P\,s \to X \times Y)$
- $\mathrm{m}_{X,Y}\,((s, v), (s', v')) = (s \bullet s', \lambda p.\,(v\,(s' \nwarrow_s p), v'\,(p \nearrow_{s'} s)))$

The category of lmf-containers

- Lmf-containers form a category **LCont**, with identities and composition inherited from **Cont**.

- $[\![-]\!]^{\mathrm{lc}}$ is a fully-faithful functor between **LCont** and **LMF**(**Set**).

$$\begin{tikzcd}
\begin{array}{c}\mathbf{LCont} \\ \cong \mathbf{Monoid}(\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \circledast^{\mathrm{c}})\end{array} \arrow[r, "U"] \arrow[d, "\text{f.f.} \quad [\![-]\!]^{\mathrm{lc}}"'] & \mathbf{Cont} \arrow[d] \\
\begin{array}{c}\mathbf{LMF}(\mathbf{Set}) \\ \cong \mathbf{Monoid}([\mathbf{Set}, \mathbf{Set}], \mathsf{Id}, \circledast)\end{array} \arrow[r, "U"] & [\mathbf{Set}, \mathbf{Set}]
\end{tikzcd}$$

$$\begin{tikzcd}
(\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \circledast^{\mathrm{c}}) \arrow[d, "{[\![-]\!]^{\mathrm{c}} \quad \text{f.f.}}"] \\
([\mathbf{Set}, \mathbf{Set}], \mathsf{Id}, \circledast)
\end{tikzcd}$$

Mnd-containers vs lmf-containers

- Any mnd-container $(S, P, \mathrm{e}, \bullet, \nwarrow, \nearrow)$ defines an lmf-container $(S, P, \mathrm{e}, \bullet', \nwarrow', \nearrow')$ by $s \bullet' s' = s \bullet (\lambda\_.\, s')$.
- Any mnd-container morphism is an lmf-container morphism.

- This gives a faithful functor from **MCont** to **LCont**. This is the functor induced by the lax monoidal functor $\mathsf{Id}_{\mathbf{Cont}} : (\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \cdot^{\mathrm{c}}) \to (\mathbf{Cont}, \mathsf{Id}^{\mathrm{c}}, \circledast^{\mathrm{c}})$.

Exception container

- Let $S = 1 + E$ for some set E and $P\,(\mathsf{inl}\,*) = 1$, $P\,(\mathsf{inr}\,\_) = 0$.
- Then $T\,X \cong X + E$.

- If, in an lmf-container structure on (S, P), we had $\mathrm{e} = \mathsf{inr}\,e_0$ for some $e_0 : E$, then $\mathsf{inr}\,e_0 \bullet \mathsf{inl}\,* = \mathsf{inl}\,*$. But then $q_0\,(\mathsf{inr}\,e_0)\,(\mathsf{inl}\,*) : 1 \to 0$, which cannot be.
- If $\mathrm{e} = \mathsf{inl}\,*$, then $\mathsf{inl}\,* \bullet s = s$, $\mathsf{inr}\,e \bullet \mathsf{inl}\,* = \mathsf{inr}\,e$, $\mathsf{inr}\,e \bullet \mathsf{inr}\,e' = e \otimes e'$ where $\otimes$ must be <u>some</u> semigroup structure on E.

- The unique mnd-container structure on (S, P) corresponds to the particular case of the <u>left zero</u> semigroup, i.e., the semigroup where $e \otimes e' = e$.

Lists container

- Let $S = \mathbb{N}$, $P\,s = [0..s)$. Then $T\,X \cong \mathsf{List}\,X$.
- The standard mnd-container structure on (S, P) gives this lmf-container structure:
  - $\mathsf{e} = 1$
  - $s \bullet s' = s * s'$
  - $s' \nwarrow_s p = p \operatorname{div} s'$, $p \nearrow_{s'} s = p \bmod s'$
- The corresponding lax monoidal functor structure on T is $\mathsf{m}^0 * = [*]$, $\mathsf{m}_{X,Y}\,(xs, ys) = [(x, y) \mid x \leftarrow xs, y \leftarrow ys]$.
- But we also have, eg, this lmf-container structure:
  - $\mathsf{e} = 1$
  - $s \bullet s' = s \min s'$
  - $s' \nwarrow_s p = p$, $p \nearrow_{s'} s = p$
- The corresponding lax monoidal functor structure is $\mathsf{m}^0 * = [*]$, $\mathsf{m}_{X,Y}\,(xs, ys) = \mathsf{zip}\,(xs, ys)$.

Lmf-containers as operads with restricted composition

- Similarly to the mnd-containers case, the list container example can be generalized.
- The appropriate generalization is a relaxation of non-symmetric operads where parallel composition is only defined when the given n operations composed with the given n-ary operation are all the same, ie, we have $\circ : O \to O \to O$ and $\#\,(o \circ o') = \#\,o * \#\,o'$.

Lmf-containers as lax $(1, \times)$-universes

- While an mnd-container defines a lax $(1, \Sigma)$-universe, an lmf-container defines a lax $(1, \times)$-universe.
- $\bullet$ is the $\times$-type former.

Containers $\cap$ commutative monads

- The monad interpreting an mnd-container is commutative (which reduces to the corresponding lax monoidal functor being symmetric) iff
  - $s \bullet (\lambda\_.\, s') = s' \bullet (\lambda\_.\, s)$
  - $(\lambda\_.\, s') \nwarrow_s p = p \nearrow_{\lambda\_.\, s} s'$

Containers $\cap$ Cartesian monads

- The monad interpreting an mnd-container is Cartesian (in the sense that all naturality squares of η, μ are pullbacks) iff
  - the function $\lambda\_.\,* : P\,\mathsf{e} \to 1$ is an isomorphism,
  - for any $s : S$, $v : P\,s \to S$, the function $\lambda p.\,(v \nwarrow_s p, p \nearrow_v s) : P\,(s \bullet v) \to \Sigma p : P\,s.\,P\,(v\,p)$ is an isomorphism.
- Such mnd-containers are proper $(1, \Sigma)$-universes.
- With Veltri, we also analyzed a number of other specializations of monads—copy monads, equational lifting monads etc.

Takeaway

- Containers whose interpretation carries a monad or a lax monoidal functor structure admit insightful explicit characterizations as mnd-containers and lmf-containers.

- These explain why set monads and lax monoidal endofunctors have very similar properties (the former also being a special case of the latter).

- Mnd-containers generalize operads, lmf-containers operads with restricted composition.
- Mnd-containers are lax $(1, \Sigma)$ universes, lmf-containers are lax $(1, \times)$ universes.