Higher Order Containers

Thorsten Altenkirch
(joint work with Sam Staton and Paul Levy)

School of Computer Science
University of Nottingham

July 4, 2010

This talk

- Containers $\approx$ strictly positive datatypes.
- Type Theory as functional programming
  with expressive types (Agda).
- Category Theory to keep things organized.
- Proof Theory and Computation?
- Propositions as Types!

Overview

1. Tutorial on containers
2. The category of containers (Cont) is cartesian closed (hence *higher order containers*).
3. Past and future of containers

Functorial Semantics of Datatypes

```
data List (A : Set) : Set where
  nil : List A
  cons : A → List A → List A
```

$\texttt{List}\, A = \mu X.1 + A \times X$

```
data RTree : Set where
  node : List RTree → RTree
```

$\texttt{RTree} = \mu\texttt{List} = \mu Y.\mu X.1 + Y \times X$

```
data BTree : Set where
  leaf : BTree
  node : BTree → BTree → BTree
```

$\texttt{BTree} = \mu X.1 + X \times X$

Exercise: Show that `RTree` and `BTree` are isomorphic.

Examples of generic constructions on functors

- For any Functor $F : \mathrm{Set} \to \mathrm{Set} \to \mathrm{Set}$ there is a natural isomorphism:

$$\alpha : \mu X.\mu Y.F\,X\,Y \simeq \mu X.F\,X\,X$$

- For any functor $G : \mathrm{Set} \to \mathrm{Set}$ we can construct the free monad

$$\mathrm{FMon}\,G : \mathrm{Set} \to \mathrm{Set}$$
$$\mathrm{FMon}\,G\,A = \mu X.A + G\,X$$

Strict positivity

- Are we permitted to write μF for any functor $F : \mathrm{Set} \to \mathrm{Set}$?
- Not every functor $F : \mathrm{Set} \to \mathrm{Set}$ has an initial algebra, e.g.

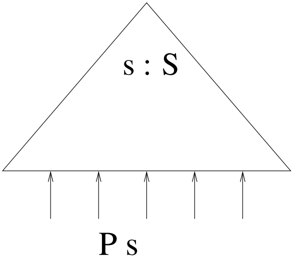
$$
\begin{aligned}
F_1\,X &= (X \to \mathrm{Bool}) \to \mathrm{Bool} \\
F_2\,X &= T(TX) \\
\mathrm{where}\, T\,X &= \mu Y.1 + X \to Y
\end{aligned}
$$

- In general we require a signature functor to be strictly positive.
  E.g. $T\,\mathbb{N} = \mu Y.1 + \mathbb{N} \to Y$.
- Containers: capture strict positivity semantically.

Definition: Container

A container $S \lhd P$ is given by $S : \mathrm{Set}$ (shapes) and $P : S \to \mathrm{Set}$ (positions).
The extension of a container is an endofunctor:

$$\llbracket S \lhd P \rrbracket : \mathrm{Set} \to \mathrm{Set}$$
$$\llbracket S \lhd P \rrbracket\, X = \Sigma s : S . P\, s \to X$$



Examples

- $(A+)X = A + X$ is given by $A+ = [\![S \lhd P]\!]$

$$\begin{aligned} S &= A + 1 \\ P\, s &= (s = \mathrm{inr}\,()) \end{aligned}$$

- $\mathrm{List}\, A = \mu X.1 + A \times X$ is given by $\mathrm{List} = [\![\mathbf{N} \lhd \mathrm{Fin}]\!]$ where

$$\mathrm{Fin}\, n = \{0, 1, \ldots, n - 1\}$$

W-types

The initial algebra of a container $S \lhd P$ always exists and is called the W-type ($W\,S\,P : \mathrm{Set}$) in Type Theory.

Container morphisms

Given containers $S \lhd P$, $T \lhd Q$, a morphism
$f \lhd r : \mathrm{Cont}\,(S \lhd P)\,(T \lhd Q)$ is given by:

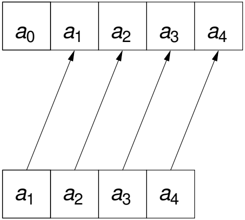
$$
\begin{aligned}
f &: S \to T \\
r &: \Pi s : S.Q\,(f\,s) \to P\,s
\end{aligned}
$$

Its extension $[\![f \lhd r]\!]$ is a natural transformation given by

$$
\begin{aligned}
[\![f \lhd r]\!]\,A &: [\![S \lhd P]\!]\,A \to [\![T \lhd Q]\!]\,A \\
&: (\Sigma s : S.P\,s \to A) \to (\Sigma t : T.Q\,t \to A) \\
[\![f \lhd r]\!]\,(s, a) &= (f\,s, a \circ r\,s)
\end{aligned}
$$

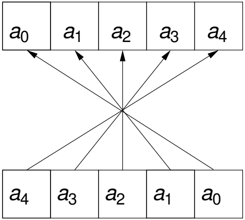
Example: tail

$\mathrm{tl}_A : \mathrm{List}\, A \to \mathrm{List}\, A$ with $\mathrm{hd}_A\, [a_0, a_1, \ldots, a_n] = [a_1, \ldots, a_n]$.
$\mathrm{hd} = [\![\lambda n.n - 1 \lhd \lambda n, i.i - 1]\!]$



Example: reverse

$\mathrm{rev}_A : \mathrm{List}\, A \to \mathrm{List}\, A$ with $\mathrm{rev}_A\, [a_0, a_1, \ldots, a_n] = [a_n, \ldots, a_1, a0]$.
$\mathrm{rev} = [\![\lambda n.n \lhd \lambda n, i.n - i]\!]$



Theorem

Every natural transformation between containers

$$\begin{aligned}\alpha_A &: [\![S \lhd P]\!]\, A \to [\![T \lhd Q]\!]\, A \\ &: (\Sigma s : S.P\, s \to A) \to (\Sigma t : T.Q\, t \to A)\end{aligned}$$

is given by a container morphism $\alpha = [\![f \lhd r]\!]$.

Given $s : S$ define

$$\begin{aligned}h_s &: \Sigma t : T.Q\, t \to P\, s \\ h_s &= \alpha_{P\, s}(s, \lambda p.p)\end{aligned}$$

then set

$$\begin{aligned}f &: S \to T \\ f\, s &= \pi_0\, h_s \\ r &: \Pi s : S.Q\,(f\, s) \to P\, s \\ r\, s\, q &= \pi_1\, h_s\end{aligned}$$

Constructions on containers

Given $S \lhd P, T \lhd Q$ we define

coproduct

$$(S \lhd P) + (T \lhd Q) = S + T \lhd \left[\begin{array}{lll} \mathrm{inl}\, s & \mapsto & P\, s \\ \mathrm{inr}\, t & \mapsto & Q\, t \end{array} \right]$$

product

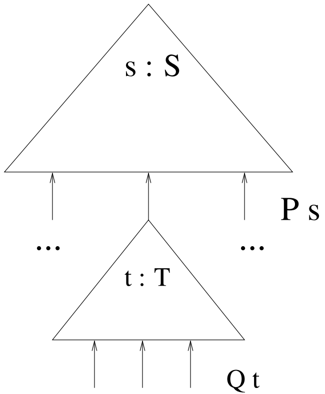
$$(S \lhd P) \times (T \lhd Q) = (s, t) : S \times T \lhd P\, s + Q\, t$$

composition

$$(S \lhd P) \circ (T \lhd Q) = (s, f) : \Sigma s : S.P\, s \to T \lhd \Sigma p : P\, s.Q(fp)$$

Coproduct and product generalize (easily) to the infinite cases.

Composition



Example: λ-terms

$\Lambda : \mathrm{Set} \to \mathrm{Set}$ is the initial solution to the equation

$$\Lambda \simeq I + \Lambda \times \Lambda + (I \to \Lambda)$$

We can eliminate the function space by:

$$\Lambda \simeq I + \Lambda \times \Lambda + \Lambda \circ (+1)$$

where $(+1)\, X = X + 1$.
We are going to show that we can always explain $\to$, i.e. that Cont is also closed under exponentiation.

Exponentials of functors

Given functors $F, G : \mathrm{Set} \to \mathrm{Set}$ what is their exponential $F \to G : \mathrm{Set} \to \mathrm{Set}$ (if it exists)?
It has to satisfy

$$\Pi X : \mathrm{Set}.(H \times F)\,X \to G\,X \simeq \Pi X : \mathrm{Set}.H\,X \to (F \to G)\,X$$

where $(H \times F)\,X = H\,X \times F\,X$.

Ends

We write $\Pi X : \mathrm{Set}.\Phi\,X\,X$ where $F : \mathrm{Set}^{\mathrm{op}} \to \mathrm{Set} \to \mathrm{Set}$ for the coend $\int_{X:\mathrm{Set}} .\Phi\,X\,X$.
Hence $\Pi X : \mathrm{Set}.F\,X \to G\,X$ is the (large) set of natural transformations.

If the exponential $F \to G$ exists, we can calculate it using the Yoneda lemma.

$$H\,X \simeq \Pi Y : \mathrm{Set}.(X \to Y) \to H\,Y \qquad \text{Yoneda}$$

$$\begin{aligned}
&(F \to G)\,X \\
&\simeq \Pi Y : \mathrm{Set}.(X \to Y) \to (F \to G)\,Y && \text{Yoneda} \\
&\simeq \Pi Y : \mathrm{Set}.(X \to Y) \times F\,Y \to G\,Y && \to\text{-adjunction} \\
&\simeq \Pi Y : \mathrm{Set}.(X \to Y) \to F\,Y \to G\,Y && \times\text{-adjunction}
\end{aligned}$$

In Set the exponential of functors doesn't always exist (see paper).

What is $I \to F$?

$$
\begin{aligned}
& (I \to F)\, X \\
& \simeq \Pi\, Y : \mathrm{Set}.(X \to Y) \times Y \to F\, Y \\
& \simeq \Pi\, Y : \mathrm{Set}.(X \to Y) \times (1 \to Y) \to F\, Y \\
& \simeq \Pi\, Y : \mathrm{Set}.(X + 1 \to Y) \to F\, Y \\
& \simeq F\,(X + 1) \\
& \simeq (F \circ (+1))\, X
\end{aligned}
$$

Lemma

In general we have for any $P : \mathrm{Set}$

$$(P \to) \to F \simeq F \circ (+P)$$

where $(P \to)\, X = P \to X$.

Exponentials of a functor by a container

A container is a coproduct of hom-functors:

$$[\![S \lhd P]\!] \simeq \Sigma s : S.((Ps) \to)$$

Given any functor $F : \mathrm{Set} \to \mathrm{Set}$

$$\begin{aligned} & [\![S \lhd P]\!] \to F \\ & \simeq (\Sigma s : S.((P\,s) \to)) \to F \\ & \simeq \Pi s : S.((P\,s) \to) \to F \\ & \simeq \Pi s : S.F \circ (+P\,s) \end{aligned}$$

Theorem

The exponential of a functor F by a container $S \lhd P$ always exists and is given by

$$[\![S \lhd P]\!] \to F = \Pi s : S.F \circ (+P\,s)$$

Exponentials of containers

Corollary

The exponential of a functor $T \lhd Q$ by a container $S \lhd P$ always exists and is given by

$$(S \lhd P) \to (T \lhd Q) = \Pi s : S.(T \lhd Q) \circ (+P\,s)$$

We can expand this:

$$\begin{aligned}
& S \lhd P \to T \lhd Q \\
& \simeq \Pi s \in S.(T \lhd Q) \circ (+P\,s) \\
& \simeq \Pi s \in S.(T \lhd Q) \circ (x : 1 + P\,s \lhd x = \mathrm{inl}\,()) \\
& \simeq \Pi s \in S.(t, f) : \Sigma t \in T.Q\,t \to 1 + P\,s \lhd \Sigma q \in Q\,s.f\,q = \mathrm{inl}\,() \\
& \simeq f \in \Pi s \in S.\Sigma t \in T.Q\,t \to 1 + P\,s \\
& \quad \lhd \Sigma s \in S.\Sigma q \in Q\,s.(f\,s).2\,q = \mathrm{inl}\,()
\end{aligned}$$

Local cartesian closure ?

- We can interpret the simply typed λ-calculus in Cont
- What about dependent types (local cartesian closure)?
- This would give us a notion of a containers for higher order functors
  (e.g. $F\,H = I + H \times H + I \to H$).
- While Cont has pullbacks, the local exponentials do not exist in general (i.e. we have Σ-types but not Π).

A short history of containers

- Abbott,A.,Ghani *Categories of Containers* (FOSSACS 03)
  n-ry containers: $\mathrm{Set}^n \to \mathrm{Set}$
- Hyland, Gambino *Wellfounded trees and dependent polynomial functors* (TYPES 03)
  Dependent polynomial functors: $\mathrm{Set}^I \to \mathrm{Set}^I$
- Abbott *Categories of Containers* (PhD 03)
- Abott,A,Ghani *Containers - Constructing Strictly Positive Types*(TCS 05)
- Abbott,A.,Ghani, McBride ∂ *for Data* (FI 05)
  Datatypes with a hole = derivatives
- A., Morris *Indexed containers* (LICS 09)
  $\mathrm{Set}^I \to \mathrm{Set}$, model inductive families

Beyond containers . . .

- How to model higher order functors as containers? (e.g. $F\,H = I + H \times H + I \to H$).
- How to interpret inductive recursive definitions?

```
data U : Set where
  nat : U
  pi : (a : U) → (T a → U) → U

T : U → Set
T nat = ℕ
T (pi a b) = (x : T a) → T (b x)
```

- Relation to Dialectica interpretation?