Containers

Ramana Kumar

March 21, 2011

Definition 1. The category, **Cont**, of containers has

Objects An object in **Cont** is a (dependent) pair (S, P), written $S \triangleright P$, where S is a set and $P = \sum_{s \in S} P(s)$ is an S-indexed family of sets. $S \triangleright P$ is called a *container*; each element $s \in S$ is called a *shape* and $P(s)$ the *positions* for that shape.

Morphisms A morphism from $S \triangleright P$ to $T \triangleright Q$ is given by a pair (f, r), written $f \triangleright r$, where $f : S \to T$ is a function on shapes and $r : \prod_{s \in S} (Q(f(s)) \to P(s))$ is an S-indexed family of functions assigning source positions to target positions.

Identities The identity on $S \triangleright P$ is $\mathrm{id}_S \triangleright \prod_{s \in S} \mathrm{id}_{P(s)}$.

Composition The composite of $(f' \triangleright r') : (S' \triangleright P') \to (S'' \triangleright P'')$ after $(f \triangleright r) : (S \triangleright P) \to (S' \triangleright P')$ is $(f' \circ f) \triangleright \prod_{s \in S} r(s) \circ r'(f(s))$.

Proposition 1. **Cont** *is a category.*

Proof.

Identities The identities defined above exist, since $\mathrm{id}_S : S \to S$ and $\mathrm{id}_{P(s)} : P(\mathrm{id}_S(s)) \to P(s)$.

Let $(f \triangleright r) : (S \triangleright P) \to (T \triangleright Q)$. Then $(f \triangleright r) \circ (\mathrm{id}_S \triangleright \prod_{s \in S} \mathrm{id}_{P(s)}) = (f \circ \mathrm{id}_S) \triangleright \prod_{s \in S} \mathrm{id}_{P(\mathrm{id}_S(s))} \circ r(s) = f \triangleright \prod_{s \in S} r(s) = f \triangleright r$.

Now let $(f \triangleright r) : (T \triangleright Q) \to (S \triangleright P)$. Then $(\mathrm{id}_S \triangleright \prod_{s \in S} \mathrm{id}_{P(s)}) \circ (f \triangleright r) = (\mathrm{id}_S \circ f) \triangleright \prod_{s \in S} r(s) \circ \mathrm{id}_{P(f(s))} = f \triangleright \prod_{s \in S} r(s) = f \triangleright r$.

Composites The composites defined above exist: taking $f \triangleright r$ and $f' \triangleright r'$ as in the definition, we have $f : S \to S'$ and $f' : S' \to S''$, hence $f' \circ f : S \to S''$, and if $s \in S$ then $r(s) : P'(f(s)) \to P(s)$, and $f(s) \in S'$ so $r'(f(s)) : P''(f'(f(s))) \to P'(f(s))$, and hence $r(s) \circ r'(f(s)) : P''((f' \circ f)(s)) \to P(s)$.

Associativity Let $(f \triangleright r) : (S \triangleright P) \to (S' \triangleright P')$, $(f' \triangleright r') : (S' \triangleright P') \to (S'' \triangleright P'')$, and $(f'' \triangleright r'') :$

$(S'' \triangleright P'') \to (S''' \triangleright P''')$. Then

$$\begin{aligned}
&((f'' \triangleright r'') \circ (f' \triangleright r')) \circ (f \triangleright r) \\
&= \left((f'' \circ f') \triangleright \prod_{s' \in S'} r'(s') \circ r''(f'(s'))\right) \circ (f \triangleright r) \\
&= ((f'' \circ f') \circ f) \triangleright \prod_{s \in S} r(s) \circ (r'(f(s)) \circ r''(f'(f(s)))) \\
&= (f'' \circ (f' \circ f)) \triangleright \prod_{s \in S} (r(s) \circ r'(f(s))) \circ r''(f'(f(s))) \\
&= (f'' \triangleright r'') \circ \left((f' \circ f) \triangleright \prod_{s \in S} r(s) \circ r'(f(s))\right) \\
&= (f'' \triangleright r'') \circ ((f' \triangleright r') \circ (f \triangleright r))
\end{aligned}$$

□

Proposition 2. **Cont** *is cartesian.*

Proof. **Terminal object** The container $\mathbf{1} \triangleright \sum_{* \in \mathbf{1}} \mathbf{0}$ is terminal in **Cont**, where $\mathbf{1} = \{*\}$ is terminal and $\mathbf{0} = \emptyset$ is initial in **Set**. Given any container $S \triangleright P$, a morphism to $\mathbf{1} \triangleright \sum_{* \in \mathbf{1}} \mathbf{0}$ is given by ${}_S! \triangleright \prod_{s \in S} !_{P(s)}$. **TODO:** type check, uniqueness

Binary products Let $S \triangleright P$ and $T \triangleright Q$ be containers. Their product is given by $S \times T \triangleright \sum_{(s,t) \in S \times T} P(s) + Q(t)$ with projections $\pi_1 \triangleright \prod_{(s,t)} \iota_1$ and $\pi_2 \triangleright \prod_{(s,t)} \iota_2$. **TODO:** type check, universal property

□

Proposition 3. **Cont** *is cocartesian.*

Proof. **Initial object** $\mathbf{0} \triangleright \mathbf{0}$ is initial in **Cont**. **TODO:** check

Binary coproducts Let $S \triangleright P$ and $T \triangleright Q$ be containers. Their coproduct is given by $S + T \triangleright P + Q$ with injections $\iota_1 \triangleright \prod_{s \in S} \mathrm{id}_{P(s)}$ and $\iota_2 \triangleright \prod_{t \in T} \mathrm{id}_{Q(t)}$. **TODO:** type check, universal property

□

Definition 2. The *extension* of a container $S \triangleright P$ is an endofunctor $[S \triangleright P]$ on **Set** given by

On objects For $X \in \mathbf{Set}$, $[S \triangleright P](X) = \sum_{s \in S} (P(s) \Rightarrow X)$ is the S-indexed family of functions from positions to X.

On morphisms For $f : X \to Y$, $[S \triangleright P](f) = \lambda(s \in S, g \in (P(s) \Rightarrow X)).\,(s, f \circ g)$ is post-composition by f to get from positions to X then to Y.

Proposition 4. *The extension of a container is a functor.*

Proof.

$$[S \triangleright P](\mathrm{id}_X) = \lambda(s,g).\,(s, \mathrm{id}_X \circ g) = \lambda(s,g).\,(s,g) = \mathrm{id}_{\sum_{s \in S}(P(s) \Rightarrow X)}$$

and

$$[S \triangleright P](f \circ f') = \lambda(s,g).\,(s, f \circ f' \circ g) = \lambda(s,g').\,(s, f \circ g') \circ \lambda(s,g).\,(s, f' \circ g) = [S \triangleright P](f) \circ [S \triangleright P](f')$$

□

Definition 3. The *extension* of a morphism $f \triangleright r : (S \triangleright P \to T \triangleright Q)$ in **Cont** is a natural transformation $[f \triangleright r]$ between extensions given by

$$[f \triangleright r]_X = \lambda(s \in S, g \in (P(s) \Rightarrow X)).\, (f(s), g \circ r(s))$$

Proposition 5. *Extensions of morphisms are natural transformations.*

Proof. Let $f \triangleright r : (S \triangleright P \to T \triangleright Q)$ be a morphism in **Cont**. Let $h : X \to Y$ be a morphism in **Set**. We must show that the following naturality condition holds.

$$\begin{array}{ccc} [S \triangleright P]\,(X) & \xrightarrow{[S \triangleright P](h)} & [S \triangleright P]\,(Y) \\ \Big\downarrow{\scriptstyle [f \triangleright r]_X} & & \Big\downarrow{\scriptstyle [f \triangleright r]_Y} \\ [T \triangleright Q]\,(X) & \xrightarrow[{[T \triangleright Q](h)}]{} & [T \triangleright Q]\,(Y) \end{array}$$

Let $(s, g) \in [S \triangleright P]\,(X)$ (so $s \in S$ and $g : P(s) \to X$). Then

$$\begin{aligned} & [T \triangleright Q]\,(h)([f \triangleright r]_X\,(s, g)) \\ &= [T \triangleright Q]\,(h)(f(s), g \circ r(s)) \\ &= (f(s), h \circ g \circ r(s)) \\ &= [f \triangleright r]_Y\,(s, h \circ g) \\ &= [f \triangleright r]_Y\,([S \triangleright P]\,(h)(s, g)) \end{aligned}$$

□

Proposition 6. *The taking of extensions, $[-]$, is a functor (from* **Cont** *to* (**Set** $\Rightarrow$ **Set**)*).*

Proof. Let $S \triangleright P$ be a container and X a set. Then

$$[\mathrm{id}_{S \triangleright P}]_X = \left[\mathrm{id}_S \triangleright \prod_{s \in S} \mathrm{id}_{P(s)}\right]_X = \lambda(s, g).\, (\mathrm{id}_S(s), g \circ \mathrm{id}_{P(s)}) = \lambda(s, g).\, (s, g) = \mathrm{id}_X$$

Now let $f \triangleright r$ and $f' \triangleright r'$ be as in Definition 1. Then

$$\begin{aligned} & [(f' \triangleright r') \circ (f \triangleright r)]_X \\ &= \left[(f' \circ f) \triangleright \prod_{s \in S} r(s) \circ r'(f(s))\right]_X \\ &= \lambda(s, g).\, ((f' \circ f)(s), g \circ (r(s) \circ r'(f(s)))) \\ &= \lambda(s, g).\, (f'(f(s)), g \circ r(s) \circ r'(f(s))) \\ &= (\lambda(s', g').\, (f'(s'), g' \circ r'(s'))) \circ (\lambda(s, g).\, (f(s), g \circ r(s))) \\ &= [f' \triangleright r']_X \circ [f \triangleright r]_X \end{aligned}$$

□

Proposition 7. *Every natural transformation between extensions of containers is the extension of a unique morphism in* **Cont**.

Proof. **TODO:** the proof ☐

Proposition 8. *The functor* $[-]$ *is full and faithful.*

Proof. **TODO:** the proof ☐

Proposition 9. $\mathbf{Cont}$ *is a full subcategory of* $\mathbf{Set} \Rightarrow \mathbf{Set}$.

Proof. **TODO:** the proof ☐

Proposition 10. *The functor* $[-]$ *preserves products.*

Proof. **TODO:** the proof ☐

Proposition 11. *The functor* $[-]$ *preserves coproducts.*

Proof. **TODO:** the proof ☐

Proposition 12. *The functor* $[-]$ *preserves pullbacks.*

Proof. **TODO:** the proof ☐