Procontainers

for

idioms, arrows and monads

E. Rivas (TTU)

Procontainers

for

idioms, arrows and monads

Procontainers

for

idioms, arrows and monads

containers

Containers

They are a class of functors

$$\mathsf{F} : \mathbb{C} \to \mathbb{C}$$

Containers

They are a class of functors

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

Containers

They are a class of functors

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

They can be characterised as functors of the form

$$\sum_{s \in S} X^{P(s)} \qquad \text{for data } S : \mathrm{Set},\ P : S \to \mathrm{Set}.$$

Containers

They are a class of functors

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

They can be characterised as functors of the form

$$\sum_{s \in S} P(s) \Rightarrow X \qquad \text{for data } S : \mathsf{Set},\ P : S \to \mathsf{Set}.$$

Containers

They are a class of functors

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

They can be characterised as functors of the form

$$\sum_{s \in S} P(s) \Rightarrow X \qquad \text{for data } S : \mathsf{Set}, P : S \to \mathsf{Set}.$$

We write $S \triangleright P$ for a container, and $[\![S \triangleright P]\!] = \sum_{s \in S} X^{P(s)}$ for its realization.

Containers

They are a class of functors

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

They can be characterised as functors of the form

$$\sum_{s \in S} P(s) \Rightarrow X \qquad \text{for data } S : \mathsf{Set}, P : S \to \mathsf{Set}.$$

We write $S \triangleright P$ for a container, and $[\![S \triangleright P]\!] = \sum_{s \in S} X^{P(s)}$ for its realization.

Equivalently, data can be captured as a bundle $p : E \to B$.

Containers

E.g., lists

$$L\,X = \sum_{n \in \mathbb{N}} X^{\mathrm{Fin}(n)}$$

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

Containers

E.g., lists

$$L\,X = \sum_{n \in \mathbb{N}} X^{\mathsf{Fin}(n)}$$

where $\mathsf{Fin}(n) = \{0, \ldots, n-1\}$

They are closed under many functor operations

- Products
- Coproducts
- Composition
- Day convolution w.r.t. products

Containers

A morphism $f \triangleright f^{\#} : S \triangleright P \to T \triangleright Q$ is given by:

- $f : S \to T$
- $f^{\#} : \forall s. Q(f(s)) \to P(s)$

Containers

A morphism $f \triangleright f^{\#} : S \triangleright P \to T \triangleright Q$ is given by:

- $f : S \to T$
- $f^{\#} : \forall s. Q(f(s)) \to P(s)$

A morphism between containers is realized as a natural transformation:

$$\left[\!\left[f \triangleright f^{\#} \right]\!\right]_X : [\![S \triangleright P]\!]\ X \to [\![T \triangleright Q]\!]\ X$$

Containers

A morphism $f \triangleright f^{\#} : S \triangleright P \to T \triangleright Q$ is given by:

- $f : S \to T$
- $f^{\#} : \forall s. Q(f(s)) \to P(s)$

A morphism between containers is realized as a natural transformation:

$$\left[\!\left[f \triangleright f^{\#} \right]\!\right]_X : [\![S \triangleright P]\!]\, X \to [\![T \triangleright Q]\!]\, X$$

Containers form a category Cont.

Containers

A morphism $f \triangleright f^{\#} : S \triangleright P \to T \triangleright Q$ is given by:

- $f : S \to T$
- $f^{\#} : \forall s. Q(f(s)) \to P(s)$

A morphism between containers is realized as a natural transformation:

$$\left[\!\left[f \triangleright f^{\#} \right]\!\right]_X : [\![S \triangleright P]\!]\ X \to [\![T \triangleright Q]\!]\ X$$

Containers form a category Cont.

It has multiple monoidal structures:

$$(\mathrm{Cont}, \times, K_1) \quad (\mathrm{Cont}, +, K_0) \quad (\mathrm{Cont}, \circ, \mathrm{Id}) \quad (\mathrm{Cont}, \star, \mathrm{Id})$$

Procontainers

for

idioms, arrows and monads

idioms, arrows and monads

Computational effects

```
let f (x : int) = if x % 2 = 0 then x / 2 else 3 * x + 1
```

Computational effects

```
let f (x : int) = if x % 2 = 0 then x / 2 else 3 * x + 1
```

$$\rightsquigarrow [\![\mathsf{f}]\!] : \mathbb{Z} \to \mathbb{Z}$$

Computational effects

```
let f (x : int) = if x % 2 = 0 then x / 2 else 3 * x + 1
```

$$\rightsquigarrow [\![\mathsf{f}]\!] : \mathbb{Z} \to \mathbb{Z}$$

“A value is, a computation does” – P. B. Levy

Computational effects

```
let f (x : int) = if x % 2 = 0 then x / 2 else 3 * x + 1
```

$$\rightsquigarrow [\![\mathsf{f}]\!] : \mathbb{Z} \to \mathbb{Z}$$

"A value is, a computation does" – P. B. Levy

An effect is something that a computation can do when interacting with its environment:

- Print a message in the screen.
- Read some bytes from a network socket.
- Use a memory cell to store a value.

Idioms, arrows and monads

We think of effects as given by primitive operations. E.g., read : $1 \rightsquigarrow \mathbb{B}$, write : $\mathbb{B} \rightsquigarrow 1$

But how do we compose such primitive operations to obtain a program?

- Monads
- Idioms, or applicative functors
- Arrows

Idioms, arrows and monads

We think of effects as given by primitive operations. E.g., read : $1 \rightsquigarrow \mathbb{B}$, write : $\mathbb{B} \rightsquigarrow 1$

But how do we compose such primitive operations to obtain a program?

- Monads

```
m :: * -> *
return :: a -> m a
bind :: m a -> (a -> m b) -> m b
```

- Idioms, or applicative functors
- Arrows

Idioms, arrows and monads

We think of effects as given by primitive operations. E.g., read $: 1 \rightsquigarrow \mathbb{B}$, write $: \mathbb{B} \rightsquigarrow 1$

But how do we compose such primitive operations to obtain a program?

- Monads
- Idioms, or applicative functors

```
f :: * -> *
pure :: a -> f a
app  :: f a -> f (a -> b) -> f b
```

- Arrows

Idioms, arrows and monads

We think of effects as given by primitive operations. E.g., read : $1 \rightsquigarrow \mathbb{B}$, write : $\mathbb{B} \rightsquigarrow 1$

But how do we compose such primitive operations to obtain a program?

- Monads
- Idioms, or applicative functors
- Arrows

```
p :: * -> * -> *
first :: p a b -> p (a, c) (b, c)
arr :: (a -> b) -> p a b
(>>>) :: p a b -> p b c -> p a c
```

Procontainers

for

idioms, arrows and monads

Containers for idioms and monads... and arrows?

Containers are a class of endofunctors.

Containers for idioms and monads... and arrows?

Containers are a class of endofunctors.

Interfaces of computational effects build on endofunctors:

- For monads `m : * -> *`
- For idioms `f : * -> *`.

Containers for idioms and monads... and arrows?

Containers are a class of endofunctors.

Interfaces of computational effects build on endofunctors:

- For monads `m : * -> *`
- For idioms `f : * -> *`.

Monad: monoid in $(\mathsf{Cont}, \circ, \mathsf{Id})$

Containers for idioms and monads... and arrows?

Containers are a class of endofunctors.

Interfaces of computational effects build on endofunctors:

- For monads `m : * -> *`
- For idioms `f : * -> *`.

Monad: monoid in $(\text{Cont}, \circ, \text{Id})$, idiom: monoid in $(\text{Cont}, \star, \text{Id})$.

Containers for idioms and monads... and arrows?

Containers are a class of endofunctors.

Interfaces of computational effects build on endofunctors:

- For monads `m : * -> *`
- For idioms `f : * -> *`.

Monad: monoid in $(\mathsf{Cont}, \circ, \mathsf{Id})$, idiom: monoid in $(\mathsf{Cont}, \star, \mathsf{Id})$.

What about arrows? In the definition, `p` is not an endofunctor... it is a profunctor:

```
p :: * -> * -> *
```

Containers for idioms and monads... and arrows?

Containers are a class of endofunctors.

Interfaces of computational effects build on endofunctors:

- For monads `m : * -> *`
- For idioms `f : * -> *`.

Monad: monoid in $(\text{Cont}, \circ, \text{Id})$, idiom: monoid in $(\text{Cont}, \star, \text{Id})$.

What about arrows? In the definition, `p` is not an endofunctor... it is a profunctor:

```
p :: * -> * -> *
```

Can we find procontainers such that arrows are monoids w.r.t. a monoidal structure?

Procontainers

for

idioms, arrows and monads

Procontainers

Procontainers are a class of profunctors

$$P : \mathsf{Set}^{op} \times \mathsf{Set} \to \mathsf{Set}$$

Procontainers

Procontainers are a class of profunctors

$$P : \mathsf{Set}^{op} \times \mathsf{Set} \to \mathsf{Set}$$

They are of the form

$$\sum_{s \in S} \left(X \Rightarrow \left(\sum_{p^+ \in P^+(s)} P^-(s, p^+) \Rightarrow Y \right) \right)$$

for data

$$S : \mathsf{Set}, P^+ : S \to \mathsf{Set}, P^- : \sum_{s \in S} P^+(s) \to \mathsf{Set}$$

Procontainers

Procontainers are a class of profunctors

$$P : \mathsf{Set}^{op} \times \mathsf{Set} \to \mathsf{Set}$$

They are of the form

$$\sum_{s \in S} \left(X \Rightarrow \left(\sum_{p^+ \in P^+(s)} P^-(s, p^+) \Rightarrow Y \right) \right)$$

for data

$$S : \mathsf{Set}, P^+ : S \to \mathsf{Set}, P^- : \sum_{s \in S} P^+(s) \to \mathsf{Set}$$

Equivalently,

$$F \longrightarrow E \longrightarrow B$$

Procontainers

A value of $[\![S \triangleright P]\!]\ X$ is a shape together with a function, giving an X value for each position.

Procontainers

A value of $[\![S \triangleright P]\!]\ X$ is a shape together with a function, giving an X value for each position.
E.g. for List $= \mathbb{N} \triangleright$ Fin,

$$\left[\text{true}, \text{false}, \text{true}\right] : [\![\mathbb{N} \triangleright \text{Fin}]\!]\ \mathbb{B} = \left(3 : \mathbb{N}, \{\bullet_0 \mapsto \text{true}, \bullet_1 \mapsto \text{false}, \bullet_2 \mapsto \text{true}\} : \text{Fin } 3 \to \mathbb{B}\right)$$

Procontainers

A value of $[\![S \triangleright P]\!]\ X$ is a shape together with a function, giving an X value for each position.
E.g. for List $= \mathbb{N} \triangleright$ Fin,

$$\left[\mathsf{true}, \mathsf{false}, \mathsf{true}\right] : [\![\mathbb{N} \triangleright \mathsf{Fin}]\!]\ \mathbb{B} = \left(3 : \mathbb{N}, \{\bullet_0 \mapsto \mathsf{true}, \bullet_1 \mapsto \mathsf{false}, \bullet_2 \mapsto \mathsf{true}\} : \mathsf{Fin}\ 3 \to \mathbb{B}\right)$$

A procontainer could be pictured as sort of a set of containers.

A value of $[\![S \triangleright P^{+} \triangleright P^{-}]\!]\ (X, Y)$ is a choice of a container ($s \in S$) together with a function

$$X \to [\![P^{+}(s) \triangleright P^{-}(s)]\!]\ Y$$

Procontainers

Given a signature $\{\mathsf{op}_\sigma : \mathsf{A}_\sigma \rightsquigarrow \mathsf{B}_\sigma\}_{\sigma \in \Sigma}$, encoded as a polynomial:

$$\left(\sum_{\sigma : \Sigma} \mathsf{A}_\sigma\right) \triangleright \lambda(\sigma, \_).\mathsf{B}_\sigma$$

where each operation $\mathsf{op} : \mathsf{A} \rightsquigarrow \mathsf{B}$ is represented as a functor $\mathsf{F}(\mathsf{X}) = \mathsf{A} \times (\mathsf{B} \Rightarrow \mathsf{X})$.

Procontainers

Given a signature $\{\mathsf{op}_\sigma : A_\sigma \rightsquigarrow B_\sigma\}_{\sigma \in \Sigma}$, encoded as a polynomial:

$$\left(\sum_{\sigma:\Sigma} A_\sigma\right) \triangleright \lambda(\sigma, \_).B_\sigma$$

where each operation $\mathsf{op} : A \rightsquigarrow B$ is represented as a functor $F(X) = A \times (B \Rightarrow X)$.

As a procontainer, we can encode it separating Σ from A_σ:

$$\Sigma \triangleright \lambda\sigma.A_\sigma \triangleright \lambda(\sigma, \_).B_\sigma$$

where $\mathsf{op} : A \rightsquigarrow B$ is represented as the profunctor $F(X, Y) = X \Rightarrow (A \times (B \Rightarrow Y))$.

Procontainers

As in the case of containers, procontainers form a category.

Procontainers

As in the case of containers, procontainers form a category.

Natural transformations between realized profunctors of $\mathsf{P} = \mathsf{S} \triangleright \mathsf{P}^+ \triangleright \mathsf{P}^-$ and $\mathsf{Q} = \mathsf{T} \triangleright \mathsf{Q}^+ \triangleright \mathsf{Q}^-$

$$\alpha_{\mathsf{X},\mathsf{Y}} : [\![\mathsf{S} \triangleright \mathsf{P}^+ \triangleright \mathsf{P}^-]\!]\ (\mathsf{X}, \mathsf{Y}) \longrightarrow [\![\mathsf{T} \triangleright \mathsf{Q}^+ \triangleright \mathsf{Q}^-]\!]\ (\mathsf{X}, \mathsf{Y})$$

Procontainers

As in the case of containers, procontainers form a category.

Natural transformations between realized profunctors of $P = S \triangleright P^+ \triangleright P^-$ and $Q = T \triangleright Q^+ \triangleright Q^-$

$$\alpha_{X,Y} : \sum_{s \in S} \left(X \Rightarrow \sum_{p^+ \in P^+(s)} P^-(s, p^+) \Rightarrow Y \right) \longrightarrow \sum_{t \in T} \left(X \Rightarrow \sum_{q^+ \in Q^+(t)} Q^-(t, q^+) \Rightarrow Y \right)$$

Procontainers

As in the case of containers, procontainers form a category.

Natural transformations between realized profunctors of $P = S \triangleright P^+ \triangleright P^-$ and $Q = T \triangleright Q^+ \triangleright Q^-$

$$\alpha_{X,Y} : \sum_{s \in S} \left(X \Rightarrow \sum_{p^+ \in P^+(s)} P^-(s, p^+) \Rightarrow Y \right) \longrightarrow \sum_{t \in T} \left(X \Rightarrow \sum_{q^+ \in Q^+(t)} Q^-(t, q^+) \Rightarrow Y \right)$$

A procontainer morphism is given by

- $f : S \to T$
- $f^+ : \forall s. P^+(s) \to Q^+(f(s))$
- $f^- : \forall s. \forall p^+. Q^-(f(s), f^+(p^+)) \to P^-(s, p^+)$

Procontainers

They are closed under useful constructors:

- Products
- Coproducts
- Bénabou's composition

Procontainers

They are closed under useful constructors:

- Products
- Coproducts
- Bénabou's composition

Recall: Bénabou's composition is the horizontal composition of two profunctors:

$$(\mathsf{P} \otimes \mathsf{Q} : \mathbb{C} \nrightarrow \mathbb{E})\,(\mathsf{X}, \mathsf{Y}) = \int^{\mathsf{I} \in \mathbb{D}} \mathsf{P}(\mathsf{X}, \mathsf{I}) \times \mathsf{Q}(\mathsf{I}, \mathsf{Y})$$

Procontainers

They are closed under useful constructors:

- Products
- Coproducts
- Bénabou's composition

Recall: Bénabou's composition is the horizontal composition of two profunctors:

$$(\mathsf{P} \otimes \mathsf{Q} : \mathbb{C} \nrightarrow \mathbb{E})\,(\mathsf{X}, \mathsf{Y}) = \int^{\mathsf{I} \in \mathbb{D}} \mathsf{P}(\mathsf{X}, \mathsf{I}) \times \mathsf{Q}(\mathsf{I}, \mathsf{Y})$$

Picking an object, endoprofunctors with Bénabou's composition form with it a monoidal structure, with Hom as unit.

Procontainers

For procontainers, $P = S \triangleright P^{+} \triangleright P^{-}$, $Q = T \triangleright Q^{+} \triangleright Q^{-}$:

$$P \otimes Q = (S \times T) \triangleright (\lambda(s,t).\sum_{s^{+} \in P^{+} s} (P^{-}(s,s^{+}) \Rightarrow Q^{+}(t)))$$

$$\triangleright\, \lambda(s,t)(s^{+},f).\sum_{s^{-} \in P^{-}(s,s^{+})} Q^{-}(t, f(s^{-}))$$

Procontainers

For procontainers, $P = S \triangleright P^+ \triangleright P^-$, $Q = T \triangleright Q^+ \triangleright Q^-$:

$$P \otimes Q = (S \times T) \triangleright (\lambda(s,t). \sum_{s^+ \in P^+ s} (P^-(s,s^+) \Rightarrow Q^+(t)))$$
$$\triangleright\, \lambda(s,t)(s^+,f). \sum_{s^- \in P^-(s,s^+)} Q^-(t, f(s^-))$$

A monoid w.r.t. this structure has:

$$m \quad : \quad P \otimes P \longrightarrow P \qquad\qquad e \quad : \quad \mathrm{Hom} \longrightarrow P$$

Procontainers

For procontainers, $P = S \triangleright P^{+} \triangleright P^{-}$, $Q = T \triangleright Q^{+} \triangleright Q^{-}$:

$$
\begin{aligned}
P \otimes Q = & (S \times T) \triangleright (\lambda(s,t). \sum_{s^{+} \in P^{+} s} (P^{-}(s, s^{+}) \Rightarrow Q^{+}(t))) \\
& \triangleright \lambda(s,t)(s^{+}, f). \sum_{s^{-} \in P^{-}(s, s^{+})} Q^{-}(t, f(s^{-}))
\end{aligned}
$$

A monoid w.r.t. this structure has:

$$
\begin{array}{llll}
m & : & P \otimes P & \longrightarrow & P \\
m_{X,Y} & : & \int^{I \in \mathbb{D}} P(X, I) \times P(I, Y) & \longrightarrow & P(X, Y)
\end{array}
\qquad
\begin{array}{llll}
e & : & \mathrm{Hom} & \longrightarrow & P \\
e_{X,Y} & : & \mathrm{Hom}(X, Y) & \longrightarrow & P(X, Y)
\end{array}
$$

Procontainers

For procontainers, $P = S \triangleright P^+ \triangleright P^-$, $Q = T \triangleright Q^+ \triangleright Q^-$:

$$P \otimes Q = (S \times T) \triangleright (\lambda(s,t). \sum_{s^+ \in P^+ s} (P^-(s,s^+) \Rightarrow Q^+(t)))$$
$$\triangleright\, \lambda(s,t)(s^+, f). \sum_{s^- \in P^-(s,s^+)} Q^-(t, f(s^-))$$

A monoid w.r.t. this structure has:

$$\begin{array}{rclcl} m & : & P \otimes P & \longrightarrow & P \\ m_{X,Y} & : & \int^{I \in \mathbb{D}} P(X,I) \times P(I,Y) & \longrightarrow & P(X,Y) \\ \texttt{(>>>)} & \texttt{::} & \texttt{(p x i, p i y)} & \texttt{->} & \texttt{p x y} \end{array} \qquad \begin{array}{rclcl} e & : & Hom & \longrightarrow & P \\ e_{X,Y} & : & Hom(X,Y) & \longrightarrow & P(X,Y) \\ \texttt{arr} & \texttt{::} & \texttt{(x -> y)} & \texttt{->} & \texttt{p x y} \end{array}$$

Procontainers

What about `first`?

Procontainers

What about `first`?

```
first :: p x y -> p (x, z) (y, z)
```

Procontainers

What about `first`?

```
first :: p x y -> p (x, z) (y, z)
```

We are interested in strong profunctors and strong natural transformations:

$$\mathrm{st}_{X,Y,Z} : P(X, Y) \longrightarrow P(X \times Z, Y \times Z)$$

Procontainers

What about `first`?

```
first :: p x y -> p (x, z) (y, z)
```

We are interested in strong profunctors and strong natural transformations:

$$\mathsf{st}_{X,Y,Z} : P(X, Y) \longrightarrow P(X \times Z, Y \times Z)$$

Procontainers come with a canonical strength. Moreover, this strength is unique.

$$\mathsf{st}_{X,Y,Z} : [\![S \rhd P^{+} \rhd P^{-}]\!] \, (X, Y) \longrightarrow [\![S \rhd P^{+} \rhd P^{-}]\!] \, (X \times Z, Y \times Z)$$

Procontainers

What about `first`?

```
first :: p x y -> p (x, z) (y, z)
```

We are interested in strong profunctors and strong natural transformations:

$$\mathrm{st}_{X,Y,Z} : P(X, Y) \longrightarrow P(X \times Z, Y \times Z)$$

Procontainers come with a canonical strength. Moreover, this strength is unique.

$$\mathrm{st}_{X,Y,Z} : \llbracket S \triangleright P^+ \triangleright P^- \rrbracket\,(X, Y) \longrightarrow \llbracket S \triangleright P^+ \triangleright P^- \rrbracket\,(X \times Z, Y \times Z)$$

And all morphisms between procontainers are strong as well.

Adjunctions between containers and procontainers

There are a number of connections between containers and procontainers:

$$\begin{tikzcd}
\mathsf{Cont} \arrow[r, shift right=2, "-_*"'] \arrow[r, phantom, "\perp"] & \mathsf{Procont} \arrow[l, shift right=2, "-^*"'] \arrow[r, shift right=2, "-^*"'] \arrow[r, phantom, "\perp"] & \mathsf{Cont} \arrow[l, shift right=2, "-_!"']
\end{tikzcd}$$

Adjunctions between containers and procontainers

There are a number of connections between containers and procontainers:

$$
\begin{tikzcd}
\mathsf{Cont} \arrow[r, bend right, "{-}_*"'] \arrow[r, phantom, "\perp"] & \mathsf{Procont} \arrow[l, bend right, "{-}^*"'] \arrow[r, bend right, "{-}^*"'] \arrow[r, phantom, "\perp"] & \mathsf{Cont} \arrow[l, bend right, "{-}_!"']
\end{tikzcd}
$$

Moreover, these adjunctions respect monoidal structures which encode how computational effects on different interfaces relate.

$$
\begin{tikzcd}
(\mathsf{Cont}, \circ) \arrow[r, bend right, "{-}_*"'] \arrow[r, phantom, "\perp"] & (\mathsf{Procont}, \otimes) \arrow[l, bend right, "{-}^*"'] \arrow[r, bend right, "{-}^*"'] \arrow[r, phantom, "\perp"] & (\mathsf{Cont}, \star) \arrow[l, bend right, "{-}_!"']
\end{tikzcd}
$$

Adjunctions between containers and procontainers

There are a number of connections between containers and procontainers:

$$
\begin{tikzcd}
\mathsf{Cont} \arrow[r, bend right, "-_*"'] \arrow[r, phantom, "\perp"] & \mathsf{Procont} \arrow[l, bend right, "-^*"'] \arrow[r, bend right, "-^*"'] \arrow[r, phantom, "\perp"] & \mathsf{Cont} \arrow[l, bend right, "-_!"']
\end{tikzcd}
$$

Moreover, these adjunctions respect monoidal structures which encode how computational effects on different interfaces relate.

$$
\begin{tikzcd}
(\mathsf{Cont}, \circ) \arrow[r, bend right, "-_*"'] \arrow[r, phantom, "\perp"] & (\mathsf{Procont}, \otimes) \arrow[l, bend right, "-^*"'] \arrow[r, bend right, "-^*"'] \arrow[r, phantom, "\perp"] & (\mathsf{Cont}, \star) \arrow[l, bend right, "-_!"']
\end{tikzcd}
$$

Also, we can characterise when a procontainer is an arrow (similar to T. Uustalu combinatorics of containers).

Questions? Thanks!