What Monads Can and Cannot Do with a bit of Extra Time

Rasmus Møgelberg & Maaike Zwart

IT University of Copenhagen

21 February 2024

Overview

Does the Powerset monad distribute over the Delay monad?

Overview

Does the Powerset **monad** distribute over the Delay monad?

Overview

Does the **Powerset monad** distribute over the **Delay monad**?

Overview

Does the Powerset monad **distribute** over the Delay monad?

Overview

Does the Powerset monad distribute over the Delay monad?

We will:

Overview

Does the Powerset monad distribute over the Delay monad?

We will:

- Try the obvious parallel computation

Overview

Does the Powerset monad distribute over the Delay monad?

We will:

- Try the obvious parallel computation
- See that it fails - Rasmus Møgelberg and Andrea Vezzosi

Overview

Does the Powerset monad distribute over the Delay monad?

We will:

- Try the obvious parallel computation
- See that it fails - Rasmus Møgelberg and Andrea Vezzosi
- Discover why it fails not because of idempotence!

Overview

Does the Powerset monad distribute over the Delay monad?

We will:

- Try the obvious parallel computation
- See that it fails - Rasmus Møgelberg and Andrea Vezzosi
- Discover why it fails not because of idempotence!
- Prove that it is impossible ...because of idempotence.

Monads

Why use monads?

Why combine monads?

What are monads?

Monads

Why use monads?

Models of Computation: non-determinism, probability, ...

Why combine monads?

What are monads?

Monads

Why use monads?

Models of Computation: non-determinism, probability, ...

Why combine monads?

Modelling multiple computational effects.

What are monads?

Monads

Why use monads?

Models of Computation: non-determinism, probability, ...

Why combine monads?

Modelling multiple computational effects.

What are monads?

Functors with some structure:

$$\langle \mathcal{M},\ \eta : 1 \to \mathcal{M},\ \mu : \mathcal{M}\mathcal{M} \to \mathcal{M} \rangle$$

Monads

Why use monads?

Models of Computation: non-determinism, probability, ...

Why combine monads?

Modelling multiple computational effects.

What are monads?

Functors with some structure:

$$\langle \mathcal{M},\ \eta : 1 \to \mathcal{M},\ \mu : \mathcal{M}\mathcal{M} \to \mathcal{M} \rangle$$

Powerset monad for non-determinism:

$$\langle \mathcal{P}, \eta_{\mathcal{P}}, \mu_{\mathcal{P}} \rangle$$

$$\mathcal{P}(X) = \{Y | Y \subseteq X \text{ finite}\}$$
$$\eta_{\mathcal{P}}(x) = \{x\}$$
$$\mu_{\mathcal{P}}(Y) = \bigcup Y$$

Monads

Why use monads?

Models of Computation: non-determinism, probability, ...

Why combine monads?

Modelling multiple computational effects.

What are monads?

Functors with some structure:

$$\langle \mathcal{M},\ \eta : 1 \to \mathcal{M},\ \mu : \mathcal{M}\mathcal{M} \to \mathcal{M} \rangle$$

Powerset monad for non-determinism:

$$\langle \mathcal{P}, \eta_{\mathcal{P}}, \mu_{\mathcal{P}} \rangle$$

$$\{1^{(0)}, *^{(2)}\}$$

$$\mathcal{P}(X) = \{Y | Y \subseteq X \text{ finite}\}$$
$$\eta_{\mathcal{P}}(x) = \{x\}$$
$$\mu_{\mathcal{P}}(Y) = \bigcup Y$$

$$1 * x = x$$
$$x * (y * z) = (x * y) * z$$
$$x * y = y * x$$
$$x * x = x$$

Delay Monad

For recursion / computation steps

Coinductive version:

$$\langle \mathcal{D}, \eta_{\mathcal{D}}, \mu_{\mathcal{D}} \rangle$$

$$\begin{aligned} \mathcal{D}(X) &\simeq X + \mathcal{D}(X) \\ \eta_{\mathcal{D}}(x) &= \mathsf{now}\,x = \mathsf{inl}\,x \\ & \quad\ \mathsf{step}\,x = \mathsf{inr}\,x \\ \mu_{\mathcal{D}}(d) &= \text{`adding steps'} \end{aligned}$$

Delay Monad

For recursion / computation steps

Coinductive version:

$$\langle \mathcal{D}, \eta_{\mathcal{D}}, \mu_{\mathcal{D}} \rangle$$

$$\begin{aligned} \mathcal{D}(X) &\simeq X + \mathcal{D}(X) \\ \eta_{\mathcal{D}}(x) &= \mathsf{now}\, x = \mathsf{inl}\, x \\ & \ \mathsf{step}\, x = \mathsf{inr}\, x \\ \mu_{\mathcal{D}}(d) &= \text{`adding steps'} \end{aligned}$$

$$\mu_{\mathcal{D}}(\mathsf{step}\,\mathsf{now}(\mathsf{step}\,\mathsf{step}\,\mathsf{now}\, x)) = \mathsf{step}\,\mathsf{step}\,\mathsf{step}\,\mathsf{now}\, x$$

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

Computation of a set,

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need
distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?,?,?,?,?\}$$

Computation of a set,

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?, ⌛ ,?,?,?\}$$

Computation of a set,

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?,5,?,?,?\}$$

Computation of a set,

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?, 5, ?, ?, ?\}$$

Computation of a set, via parallel computation:

 $\{?, ?, ?, ?, ?\}$

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?,5,?,?,?\}$$

Computation of a set, via parallel computation:

⌛ {⌛,⌛,⌛,⌛,⌛}

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?,5,?,?,?\}$$

Computation of a set, via parallel computation:

⌛ {2,5, ⌛ ,8, ⌛ }

Monad Composition - Distributive Laws

To make a monad $\mathcal{DP}$, need
distributive law $\lambda : \mathcal{PD} \to \mathcal{DP}$.

Set of computations:

$$\{?,5,?,?,?\}$$

Computation of a set, via parallel computation:

$$\{2,5,3,8,6\}$$

Total time: max of computation times

Parallel Computation

More precise:

$$\begin{aligned} \lambda\{\mathsf{now}\, x, \mathsf{now}\, y\} &= \mathsf{now}\{x, y\} \\ \lambda\{\mathsf{step}\, d, \mathsf{now}\, y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\, y\}) \\ \lambda\{\mathsf{step}\, d, \mathsf{step}\, d'\} &= \mathsf{step}(\lambda\{d, d'\}) \end{aligned}$$

so:

$$\{\mathsf{step}\,\mathsf{step}\,\mathsf{now}\, x, \mathsf{now}\, y, \mathsf{step}\,\mathsf{now}\, z\} \mapsto \mathsf{step}\,\mathsf{step}\,\mathsf{now}\{x, y, z\}$$

Parallel Computation

More precise:

$$\begin{aligned} \lambda\{\mathsf{now}\,x, \mathsf{now}\,y\} &= \mathsf{now}\{x, y\} \\ \lambda\{\mathsf{step}\,d, \mathsf{now}\,y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\,y\}) \\ \lambda\{\mathsf{step}\,d, \mathsf{step}\,d'\} &= \mathsf{step}(\lambda\{d, d'\}) \end{aligned}$$

so:

$$\{\mathsf{step}\,\mathsf{step}\,\mathsf{now}\,x, \mathsf{now}\,y, \mathsf{step}\,\mathsf{now}\,z\} \mapsto \mathsf{step}\,\mathsf{step}\,\mathsf{now}\{x, y, z\}$$

Not a distributive law!

Parallel Computation

More precise:

$$\begin{aligned} \lambda\{\mathsf{now}\, x, \mathsf{now}\, y\} &= \mathsf{now}\{x, y\} \\ \lambda\{\mathsf{step}\, d, \mathsf{now}\, y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\, y\}) \\ \lambda\{\mathsf{step}\, d, \mathsf{step}\, d'\} &= \mathsf{step}(\lambda\{d, d'\}) \end{aligned}$$

so:

$$\{\mathsf{step}\,\mathsf{step}\,\mathsf{now}\, x, \mathsf{now}\, y, \mathsf{step}\,\mathsf{now}\, z\} \mapsto \mathsf{step}\,\mathsf{step}\,\mathsf{now}\{x, y, z\}$$

Not a distributive law!
Why?

Parallel Computation

More precise:

$$\begin{aligned} \lambda\{\mathsf{now}\,x, \mathsf{now}\,y\} &= \mathsf{now}\{x, y\} \\ \lambda\{\mathsf{step}\,d, \mathsf{now}\,y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\,y\}) \\ \lambda\{\mathsf{step}\,d, \mathsf{step}\,d'\} &= \mathsf{step}(\lambda\{d, d'\}) \end{aligned}$$

so:

$$\{\mathsf{step\,step\,now}\,x, \mathsf{now}\,y, \mathsf{step\,now}\,z\} \mapsto \mathsf{step\,step\,now}\{x, y, z\}$$

Not a distributive law!

Why? NOT idempotence!

Parallel Computation

$$\begin{array}{ccccc} \mathcal{PDD} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{DPD} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{DDP} \\ \downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & \downarrow{\scriptstyle \mu^{\mathcal{D}}\mathcal{P}} \\ \mathcal{PD} & & \xrightarrow{\qquad\lambda\qquad} & & \mathcal{DP} \end{array}$$

$\{\mathsf{step}\ \mathsf{now}(\mathsf{now}\ x), \mathsf{now}(\mathsf{step}\ \mathsf{now}\ y)\}$

Parallel Computation

$$\begin{array}{ccccc}
\mathcal{PDD} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{DPD} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{DDP} \\
\Big\downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & {\scriptstyle \mu^{\mathcal{D}}\mathcal{P}}\Big\downarrow \\
\mathcal{PD} & & \xrightarrow{\qquad\quad\lambda\quad\qquad} & & \mathcal{DP}
\end{array}$$

$$\{\mathsf{step\ now}(\mathsf{now}\ x), \mathsf{now}(\mathsf{step\ now}\ y)\} \downarrow_{\mathcal{P}\mu^{\mathcal{D}}} \{\mathsf{step\ now}\ x, \mathsf{step\ now}\ y\}$$

Parallel Computation

$$\begin{array}{ccccc} \mathcal{PDD} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{DPD} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{DDP} \\ \downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & {\scriptstyle \mu^{\mathcal{D}}\mathcal{P}}\downarrow \\ \mathcal{PD} & & \xrightarrow{\qquad\quad\lambda\qquad\quad} & & \mathcal{DP} \end{array}$$

$$\{\mathsf{step\ now}(\mathsf{now}\ x), \mathsf{now}(\mathsf{step\ now}\ y)\} \downarrow_{\mathcal{P}\mu^{\mathcal{D}}} \{\mathsf{step\ now}\ x, \mathsf{step\ now}\ y\}$$

$$\xrightarrow{\lambda} \mathsf{step\ now}\{x, y\}$$

Parallel Computation

$$
\begin{array}{ccccc}
\mathcal{P}\mathcal{D}\mathcal{D} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{D}\mathcal{P}\mathcal{D} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{D}\mathcal{D}\mathcal{P} \\
\downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & \downarrow{\scriptstyle \mu^{\mathcal{D}}\mathcal{P}} \\
\mathcal{P}\mathcal{D} & & \xrightarrow{\quad\quad\lambda\quad\quad} & & \mathcal{D}\mathcal{P}
\end{array}
$$

$$
\begin{aligned}
\{\mathsf{step}\,\mathsf{now}(\mathsf{now}\,x), \mathsf{now}(\mathsf{step}\,\mathsf{now}\,y)\} &\downarrow_{\mathcal{P}\mu^{\mathcal{D}}} \{\mathsf{step}\,\mathsf{now}\,x, \mathsf{step}\,\mathsf{now}\,y\} \\
&\xrightarrow{\lambda} \mathsf{step}\,\mathsf{now}\{x, y\}
\end{aligned}
$$

$$\{\mathsf{step}\,\mathsf{now}(\mathsf{now}\,x), \mathsf{now}(\mathsf{step}\,\mathsf{now}\,y)\}$$

Parallel Computation

$$\begin{array}{ccccc}
\mathcal{PDD} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{DPD} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{DDP} \\
\big\downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & \big\downarrow{\scriptstyle \mu^{\mathcal{D}}\mathcal{P}} \\
\mathcal{PD} & & \xrightarrow{\qquad\quad\lambda\qquad\quad} & & \mathcal{DP}
\end{array}$$

$$\begin{aligned}
\{\mathsf{step}\,\mathsf{now}(\mathsf{now}\,x), \mathsf{now}(\mathsf{step}\,\mathsf{now}\,y)\} &\downarrow_{\mathcal{P}\mu^{\mathcal{D}}} \{\mathsf{step}\,\mathsf{now}\,x, \mathsf{step}\,\mathsf{now}\,y\} \\
&\xrightarrow{\lambda} \mathsf{step}\,\mathsf{now}\{x, y\}
\end{aligned}$$

$$\{\mathsf{step}\,\mathsf{now}(\mathsf{now}\,x), \mathsf{now}(\mathsf{step}\,\mathsf{now}\,y)\} \xrightarrow{\lambda\mathcal{D}} \mathsf{step}\,\mathsf{now}(\{\mathsf{now}\,x, \mathsf{step}\,\mathsf{now}\,y\})$$

Parallel Computation

$$\begin{array}{ccccc} \mathcal{PDD} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{DPD} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{DDP} \\ \big\downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & \big\downarrow{\scriptstyle \mu^{\mathcal{D}}\mathcal{P}} \\ \mathcal{PD} & & \xrightarrow{\qquad\lambda\qquad} & & \mathcal{DP} \end{array}$$

$$\begin{aligned} \{\text{step now}(\text{now}\,x), \text{now}(\text{step now}\,y)\} &\downarrow_{\mathcal{P}\mu^{\mathcal{D}}} \{\text{step now}\,x, \text{step now}\,y\} \\ &\xrightarrow{\lambda} \text{step now}\{x, y\} \end{aligned}$$

$$\begin{aligned} \{\text{step now}(\text{now}\,x), \text{now}(\text{step now}\,y)\} &\xrightarrow{\lambda\mathcal{D}} \text{step now}(\{\text{now}\,x, \text{step now}\,y\}) \\ &\xrightarrow{\mathcal{D}\lambda} \text{step now}(\text{step now}\{x, y\}) \end{aligned}$$

Parallel Computation

$$
\begin{array}{ccccc}
\mathcal{PDD} & \xrightarrow{\lambda\mathcal{D}} & \mathcal{DPD} & \xrightarrow{\mathcal{D}\lambda} & \mathcal{DDP} \\
\downarrow{\scriptstyle \mathcal{P}\mu^{\mathcal{D}}} & & & & \downarrow{\scriptstyle \mu^{\mathcal{D}}\mathcal{P}} \\
\mathcal{PD} & & \xrightarrow{\quad\quad\lambda\quad\quad} & & \mathcal{DP}
\end{array}
$$

$$
\begin{aligned}
\{\text{step now}(\text{now}\, x), \text{now}(\text{step now}\, y)\} &\downarrow_{\mathcal{P}\mu^{\mathcal{D}}} \{\text{step now}\, x, \text{step now}\, y\} \\
&\xrightarrow{\lambda} \text{step now}\{x, y\}
\end{aligned}
$$

$$
\begin{aligned}
\{\text{step now}(\text{now}\, x), \text{now}(\text{step now}\, y)\} &\xrightarrow{\lambda\mathcal{D}} \text{step now}(\{\text{now}\, x, \text{step now}\, y\}) \\
&\xrightarrow{\mathcal{D}\lambda} \text{step now}(\text{step now}\{x, y\}) \\
&\downarrow_{\mu^{\mathcal{D}}\mathcal{P}} \text{step step now}\{x, y\}
\end{aligned}
$$

Parallel Computation

So what went wrong?
Nothing specific to Powerset!
Only ingredient: "structure with two elements".

Theorem

Parallel computation is **never** *a distributive law for* $\mathcal{M}\mathcal{D} \to \mathcal{D}\mathcal{M}$ *if* $\mathcal{M}$ *is presented by a theory with a binary term.*

No Hope for Powerset

What can we do?

$$\begin{aligned}
\lambda\{\mathsf{now}\,x, \mathsf{now}\,y\} &= \mathsf{now}\{x, y\} \\
\lambda\{\mathsf{step}\,d, \mathsf{now}\,y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\,y\}) \\
\lambda\{\mathsf{step}\,d, \mathsf{step}\,d'\} &= \mathsf{step}(\lambda\{d, d'\})
\end{aligned}$$

No Hope for Powerset

What can we do?

$$\lambda\{\mathrm{now}\,x, \mathrm{now}\,y\} = \mathrm{now}\{x, y\}$$
$$\lambda\{\mathrm{step}\,d, \mathrm{now}\,y\} = \mathrm{step}(\lambda\{d, \mathrm{now}\,y\})$$
$$\underline{\lambda\{\mathrm{step}\,d, \mathrm{step}\,d'\}} = \mathrm{step}(\lambda\{d, d'\})$$

No Hope for Powerset

What can we do?

$$\begin{aligned} \lambda\{\mathsf{now}\,x, \mathsf{now}\,y\} &= \mathsf{now}\{x, y\} \\ \lambda\{\mathsf{step}\,d, \mathsf{now}\,y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\,y\}) \\ \underline{\lambda\{\mathsf{step}\,d, \mathsf{step}\,d'\}} &= \mathsf{step}(\lambda\{d, d'\}) \end{aligned}$$

$$\begin{array}{ccc} \mathcal{PD}(2) & \xrightarrow{\lambda} & \mathcal{DP}(2) \\ \downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\ \mathcal{PD}(1) & \xrightarrow{\lambda} & \mathcal{DP}(1) \end{array} \qquad \begin{array}{ccc} \{\mathsf{step\,now}\,x, \mathsf{step\,now}\,y\} & \xmapsto{\lambda} & ? \\ \downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\ \{\mathsf{step\,now}\,x\} & \xmapsto{\lambda} & \mathsf{step\,now}\{x\} \end{array}$$

No Hope for Powerset

What can we do?

$$\lambda\{\text{now}\,x, \text{now}\,y\} = \text{now}\{x, y\}$$
$$\lambda\{\text{step}\,d, \text{now}\,y\} = \text{step}(\lambda\{d, \text{now}\,y\})$$
$$\underline{\lambda\{\text{step}\,d, \text{step}\,d'\}} = \text{step}(\lambda\{d, d'\})$$

$$\begin{array}{ccc} \mathcal{PD}(2) & \xrightarrow{\lambda} & \mathcal{DP}(2) \\ \downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\ \mathcal{PD}(1) & \xrightarrow{\lambda} & \mathcal{DP}(1) \end{array} \qquad \begin{array}{ccc} \{\text{step now}\,x, \text{step now}\,y\} & \xmapsto{\lambda} & \text{step now}\{?\} \\ \downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\ \{\text{step now}\,x\} & \xmapsto{\lambda} & \text{step now}\{x\} \end{array}$$

No Hope for Powerset

What can we do?

$$
\begin{aligned}
\underline{\lambda\{\text{now } x, \text{now } y\}} &= \text{now}\{x, y\} \\
\lambda\{\text{step } d, \text{now } y\} &= \text{step}(\lambda\{d, \text{now } y\}) \\
\underline{\lambda\{\text{step } d, \text{step } d'\}} &= \text{step}(\lambda\{d, d'\})
\end{aligned}
$$

$$
\begin{array}{ccc}
\mathcal{PD}(2) & \xrightarrow{\lambda} & \mathcal{DP}(2) \\
\downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\
\mathcal{PD}(1) & \xrightarrow{\lambda} & \mathcal{DP}(1)
\end{array}
\qquad
\begin{array}{ccc}
\{\text{step now } x, \text{step now } y\} & \overset{\lambda}{\longmapsto} & \text{step now}\{?\} \\
\downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\
\{\text{step now } x\} & \overset{\lambda}{\longmapsto} & \text{step now}\{x\}
\end{array}
$$

$$\{?\} = \varnothing \quad \{?\} = \{x\} \quad \{?\} = \{y\} \quad \{?\} = \{x, y\}$$

No Hope for Powerset

What can we do?

$$\begin{aligned} \lambda\{\mathsf{now}\, x, \mathsf{now}\, y\} &= \mathsf{now}\{x, y\} \\ \lambda\{\mathsf{step}\, d, \mathsf{now}\, y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\, y\}) \\ \underline{\lambda\{\mathsf{step}\, d, \mathsf{step}\, d'\}} &= \mathsf{step}(\lambda\{d, d'\}) \end{aligned}$$

$$\begin{array}{ccc} \mathcal{PD}(2) & \xrightarrow{\lambda} & \mathcal{DP}(2) \\ \downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\ \mathcal{PD}(1) & \xrightarrow{\lambda} & \mathcal{DP}(1) \end{array} \qquad \begin{array}{ccc} \{\mathsf{step}\,\mathsf{now}\, x, \mathsf{step}\,\mathsf{now}\, y\} & \overset{\lambda}{\longmapsto} & \mathsf{step}\,\mathsf{now}\{?\} \\ \Big\downarrow{\scriptstyle \mathcal{PD}f} & & \Big\downarrow{\scriptstyle \mathcal{DP}f} \\ \{\mathsf{step}\,\mathsf{now}\, x\} & \overset{\lambda}{\longmapsto} & \mathsf{step}\,\mathsf{now}\{x\} \end{array}$$

$$? = \varnothing \qquad ? = \{x\} \qquad ? = \{y\} \qquad ? = \{x, y\}$$

No Hope for Powerset

What can we do?

$$\begin{aligned} \lambda\{\text{now}\,x, \text{now}\,y\} &= \text{now}\{x, y\} \\ \lambda\{\text{step}\,d, \text{now}\,y\} &= \text{step}(\lambda\{d, \text{now}\,y\}) \\ \underline{\lambda\{\text{step}\,d, \text{step}\,d'\}} &= \text{step}(\lambda\{d, d'\}) \end{aligned}$$

$$\begin{array}{ccc} \mathcal{PD}(2) & \xrightarrow{\lambda} & \mathcal{DP}(2) \\ \big\downarrow{\scriptstyle \mathcal{PD}f} & & \big\downarrow{\scriptstyle \mathcal{DP}f} \\ \mathcal{PD}(1) & \xrightarrow{\lambda} & \mathcal{DP}(1) \end{array} \qquad \begin{array}{ccc} \{\text{step now}\,x, \text{step now}\,y\} & \stackrel{\lambda}{\longmapsto} & \text{step now}\{?\} \\ \big\downarrow{\scriptstyle \mathcal{PD}f} & & \big\downarrow{\scriptstyle \mathcal{DP}f} \\ \{\text{step now}\,x\} & \stackrel{\lambda}{\longmapsto} & \text{step now}\{x\} \end{array}$$

$$? = \varnothing \qquad ? = \{x\} \qquad ? = \{y\} \qquad ? = \{x, y\}$$

No Hope for Powerset

What can we do?

$$\begin{aligned}
\lambda\{\mathsf{now}\,x, \mathsf{now}\,y\} &= \mathsf{now}\{x, y\} \\
\lambda\{\mathsf{step}\,d, \mathsf{now}\,y\} &= \mathsf{step}(\lambda\{d, \mathsf{now}\,y\}) \\
\lambda\{\mathsf{step}\,d, \mathsf{step}\,d'\} &= \mathsf{step}(\lambda\{d, d'\})
\end{aligned}$$

$$\begin{array}{ccc}
\mathcal{PD}(2) & \xrightarrow{\lambda} & \mathcal{DP}(2) \\
\downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\
\mathcal{PD}(1) & \xrightarrow{\lambda} & \mathcal{DP}(1)
\end{array}
\qquad
\begin{array}{ccc}
\{\mathsf{step\,now}\,x, \mathsf{step\,now}\,y\} & \overset{\lambda}{\longmapsto} & \mathsf{step\,now}\{?\} \\
\downarrow{\scriptstyle \mathcal{PD}f} & & \downarrow{\scriptstyle \mathcal{DP}f} \\
\{\mathsf{step\,now}\,x\} & \overset{\lambda}{\longmapsto} & \mathsf{step\,now}\{x\}
\end{array}$$

$? = \emptyset \quad ? = \{x\} \quad ? = \{y\} \quad ? = \{x, y\}$

No Hope in General?

Theorem

There is no **causal** *distributive law* $\mathcal{M}\mathcal{D} \to \mathcal{D}\mathcal{M}$*, if* $\mathcal{M}$ *is presented by a theory with an idempotent and commutative term.*

No Hope in General?

Theorem

There is no **causal** *distributive law $\mathcal{MD} \to \mathcal{DM}$, if $\mathcal{M}$ is presented by a theory with an idempotent and commutative term.*

$$\mathcal{MD}^{\kappa} \to \mathcal{D}^{\kappa}\mathcal{M} \Rightarrow \mathcal{MD} \to \mathcal{DM}$$

No Hope in General?

Theorem

There is no **causal** *distributive law* $\mathcal{MD} \to \mathcal{DM}$, *if* $\mathcal{M}$ *is presented by a theory with an idempotent and commutative term.*

$$\mathcal{MD}^{\kappa} \to \mathcal{D}^{\kappa}\mathcal{M} \Rightarrow \mathcal{MD} \to \mathcal{DM}$$

$$\mathcal{MD} \to \mathcal{DM} \not\Rightarrow \mathcal{MD}^{\kappa} \to \mathcal{D}^{\kappa}\mathcal{M}$$

Example in the paper.

Conclusion

We saw:

- Parallel computation fails to give a distributive law.
- Distributive law $\mathcal{PD} \to \mathcal{DP}$ impossible.
- Idempotence destroys causality.

Conclusion

We saw:

- Parallel computation fails to give a distributive law.
- Distributive law $\mathcal{PD} \to \mathcal{DP}$ impossible.
- Idempotence destroys causality.

More in the paper!

- Sequential computation:
    - Fails because of idempotence.
    - Works for balanced equations.
- Dist laws for Exceptions, Reader, State, Selection.

Conclusion

We saw:

- Parallel computation fails to give a distributive law.
- Distributive law $\mathcal{PD} \to \mathcal{DP}$ impossible.
- Idempotence destroys causality.

More in the paper!

- Sequential computation:
  - Fails because of idempotence.
  - Works for balanced equations.
- Dist laws for Exceptions, Reader, State, Selection.

A bit of hope:
Parallel computation does give a distributive law up to weak bisimilarity.