Lessons from failing distributive laws

Maaike Zwart

IT University of Copenhagen
maaike.annebeth@gmail.com / www.maaikezwart.com

2 March 2022
Overview

- Introduction
  - Motivation: monads and monad compositions
- How to: No-go Theorems
  - including proof of 50 year old problem!
- A crucial step
- What I am doing now
- Conclusion
Motivation: monads and monad compositions

A monad is a categorical structure used for:

- Modelling of data structures (lists, trees, etc)
Motivation: monads and monad compositions

A monad is a categorical structure used for:

- Modelling of data structures (lists, trees, etc)
- Modelling of computation (exception, reader, writer, etc)
Motivation: monads and monad compositions

A monad is a categorical structure used for:

- Modelling of data structures (lists, trees, etc)
- Modelling of computation (exception, reader, writer, etc)

Monads, monads everywhere

- Computational effects such as probability or non-determinism can be modelled as monads
- Haskell programs are structured using monads
- Algebraic theories such as those of monoids, groups, semilattices and pointed sets correspond to monads
- In topology and order theory, closure operators are monads
- Every monoid is monad
- Preorders and metric spaces are monads
- Enriched categories are monads
- Internal categories are monads
- Operads and multicategories are monads
- Lawvere theories, PROs and PROPs are monads
- Distributive laws between monads are monads (!)
- ...
Motivation: monads and monad compositions

A monad is a categorical structure used for:

- Modelling of data structures (lists, trees, etc)
- Modelling of computation (exception, reader, writer, etc)
Motivation: monads and monad compositions

A monad is a categorical structure used for:

- Modelling of data structures (lists, trees, etc)
- Modelling of computation (exception, reader, writer, etc)

Monads, monads everywhere

- Computational effects such as probability or non-determinism can be modelled as monads
- Haskell programs are structured using monads
- Algebraic theories such as those of monoids, groups, semilattices and pointed sets correspond to monads
- In topology and order theory, closure operators are monads
- Every monoid is monad
- Preorders and metric spaces are monads
- Enriched categories are monads
- Internal categories are monads
- Operads and multicategories are monads
- Lawvere theories, PROs and PROPs are monads
- Distributive laws between monads are monads (!)
- ...

Compositions of monads allow simultaneous modelling of multiple computational aspects.
Monads: Monoids in the category of endofunctors

A monad is a triple \( \langle T, \eta, \mu \rangle \), with \( T \) an endofunctor and \( \eta : 1 \Rightarrow T \), \( \mu : TT \Rightarrow T \) natural transformations, such that:

\[
\begin{array}{ccc}
T & \xrightarrow{\eta T} & TT \\
\downarrow{T\eta} & & \downarrow{T\mu} \\
TT & \xrightarrow{\mu} & T
\end{array}
\quad
\begin{array}{ccc}
TTT & \xrightarrow{T\mu} & TT \\
\downarrow{T\eta} & & \downarrow{\mu} \\
TT & \xrightarrow{\mu} & T
\end{array}
\]
Monads: Monoids in the category of endofunctors

A monad is a triple $\langle T, \eta, \mu \rangle$, with $T$ an endofunctor and $\eta : 1 \Rightarrow T$, $\mu : TT \Rightarrow T$ natural transformations, such that:

$$
\begin{array}{ccc}
T & \xrightarrow{\eta_T} & TT \\
\downarrow T\eta & & \downarrow \mu \\
TT & \xrightarrow{\mu} & T \\
\end{array}

\quad
\quad
\begin{array}{ccc}
TTT & \xrightarrow{T\mu} & TT \\
\downarrow \mu & & \downarrow \mu \\
TT & \xrightarrow{\mu} & T \\
\end{array}
$$

Examples:

- List
- Multiset/Bag
- Powerset
- Distribution

- Exception
- Writer
- Reader
- State
Composing Monads

- Find $\eta^{TS}, \mu^{TS}$ such that $\langle TS, \eta^{TS}, \mu^{TS} \rangle$ is a monad.
Composing Monads

- Find $\eta^{TS}, \mu^{TS}$ such that $\langle TS, \eta^{TS}, \mu^{TS} \rangle$ is a monad.
- Good candidate for $\eta^{TS}$:

  \[ \eta^T \eta^S : 1 \Rightarrow TS \]
Composing Monads

- Find \( \eta^{TS}, \mu^{TS} \) such that \( \langle TS, \eta^{TS}, \mu^{TS} \rangle \) is a monad.
- Good candidate for \( \eta^{TS} \):
  \[
  \eta^T \eta^S : 1 \Rightarrow TS
  \]
- Same for \( \mu^{TS} \)?
Composing Monads

- Find $\eta^{TS}$, $\mu^{TS}$ such that $\langle TS, \eta^{TS}, \mu^{TS} \rangle$ is a monad.
- Good candidate for $\eta^{TS}$:
  
  $$\eta^T \eta^S : 1 \Rightarrow TS$$

- Same for $\mu^{TS}$?
  - Need:

  $$\mu^{TS} : TSTS \Rightarrow TS$$
Composing Monads

- Find $\eta^{TS}, \mu^{TS}$ such that $\langle TS, \eta^{TS}, \mu^{TS} \rangle$ is a monad.
- Good candidate for $\eta^{TS}$:

  $$\eta^T \eta^S : 1 \Rightarrow TS$$

- Same for $\mu^{TS}$?
  - Need:

    $$\mu^{TS} : TSTS \Rightarrow TS$$

  - Have:

    $$\mu^T \mu^S : TTSS \Rightarrow TS$$
Composing Monads

- Find $\eta^{TS}, \mu^{TS}$ such that $\langle TS, \eta^{TS}, \mu^{TS} \rangle$ is a monad.
- Good candidate for $\eta^{TS}$:
  \[ \eta^T \eta^S : 1 \Rightarrow TS \]

- Same for $\mu^{TS}$?
  - Need:
    \[ \mu^{TS} : TSTS \Rightarrow TS \]
  - Have:
    \[ \mu^T \mu^S : TTSS \Rightarrow TS \]
  - Solution:
    \[ \lambda : ST \Rightarrow TS \]
Composing Monads

• Find $\eta^{TS}, \mu^{TS}$ such that $\langle TS, \eta^{TS}, \mu^{TS} \rangle$ is a monad.
• Good candidate for $\eta^{TS}$:

  $\eta^T \eta^S : 1 \Rightarrow TS$

• Same for $\mu^{TS}$?
  • Need:

    $\mu^{TS} : TSTS \Rightarrow TS$

  • Have:

    $\mu^T \mu^S : TTSS \Rightarrow TS$

  • Solution:

    $\lambda : ST \Rightarrow TS$

• If $\lambda$ is a *distributive law*, then the above choices form a monad.
Composing Monads with Distributive Laws

The following composite is a monad - Beck 1969.

\[ \langle TS, \eta^T \eta^S, \mu^T \mu^S \cdot T\lambda S \rangle, \]

where \( \lambda : ST \rightarrow TS \) is a natural transformation satisfying the following axioms.

\[ \begin{align*}
ST & \xrightarrow{T} TS \\
SST & \xrightarrow{S\lambda} STS \xrightarrow{\lambda S} TSS \\
STT & \xrightarrow{\lambda T} TST \xrightarrow{T\lambda} TTTS
\end{align*} \]
Examples

There is a distributive law for Powerset over List. It works like the famous ‘times over plus’ distributivity:

\[(a + b) \times c = a \times c + b \times c\]

\[[\{a, b\}, \{c\}] \mapsto \{[a, c], [b, c]\}\]

Many more work like this:

- Multiset over itself
- List over Multiset
- Multiset over Powerset
- ...
That sounds easy, but...

Problem:

- Distributive laws are hard to find.
  → time consuming.
That sounds easy, but...

Problem:

- Distributive laws are hard to find.
  → time consuming.
- Axioms are hard to check (even though they look easy).
  → mistakes in the literature!
That sounds easy, but...

Problem:

- Distributive laws are hard to find.  
  → time consuming.

- Axioms are hard to check (even though they look easy).  
  → mistakes in the literature!

- Distributive laws might not even exist.
That sounds easy, but...

Problem:

- Distributive laws are hard to find.
  → time consuming.
- Axioms are hard to check (even though they look easy).
  → mistakes in the literature!
- Distributive laws might not even exist.
That sounds easy, but...

Problem:

- Distributive laws are hard to find.
  → time consuming.
- Axioms are hard to check (even though they look easy).
  → mistakes in the literature!
- Distributive laws might not even exist.

My Thesis:

- No-go theorems for distributive laws.
That sounds easy, but...

Problem:

- Distributive laws are hard to find.
  → time consuming.
- Axioms are hard to check (even though they look easy).
  → mistakes in the literature!
- Distributive laws might not even exist.

My Thesis:

- No-go theorems for distributive laws.

My weapon of choice:

- Algebra.
A quick reminder: algebraic theories

Algebraic theory:

- Signature $\Sigma$ and a set of variables give *terms*.
- Axioms $E$ and equational logic give equivalence of terms.

Monoids:

$$\Sigma = \{1^{(0)}, \ast^{(2)}\}$$
$$E = \{1 \ast x = x = x \ast 1,$$
$$(x \ast y) \ast z = x \ast (y \ast z)\}$$

Abelian groups:

$$\Sigma = \{0^{(0)}, -^{(1)}, +^{(2)}\}$$
$$E = \{0 + x = x = x + 0,$$
$$(x + y) + z = x + (y + z),$$
$$x + y = y + x,$$
$$x + (-x) = 0 = (-x) + x\}$$
The algebraic equivalent of distributive laws

Monads $\leftrightarrow$ Algebraic theories
Distributive laws $\leftrightarrow$
The algebraic equivalent of distributive laws

Monads $\iff$ Algebraic theories
Distributive laws $\iff$ Composite theories - Piróg and Staton 2017.
The algebraic equivalent of distributive laws

Monads \iff \text{Algebraic theories}
Distributive laws \iff \text{Composite theories} \quad \text{- Piróg and Staton 2017.}

Example: Composing Abelian groups and Monoids: Rings!

\[
\begin{align*}
\Sigma^R &= \Sigma^A \uplus \Sigma^M \\
&= \{0^{(0)}, 1^{(0)}, -^{(1)}, +^{(2)}, \ast^{(2)}\} \\
E^R &= \Sigma^A \cup \Sigma^M \cup \\
&\{a \ast (b + c) = (a \ast b) + (a \ast c) \\
(a + b) \ast c &= (a \ast c) + (b \ast c)\}
\end{align*}
\]
The algebraic equivalent of distributive laws

Monads \iff Algebraic theories
Distributive laws \iff Composite theories - Piróg and Staton 2017.

- Terms can be separated

\[ a \ast (b + c) = (a \ast b) + (a \ast c) \]

- Equality preservation of component theories (*essential uniqueness*) - only for two separated terms!

\[ (a \ast b) + c =_R c + (a \ast b) \]
\[ \iff x + c =_A c + x \text{ and } a \ast b =_M a \ast b \]
My strategy: no-go theorems for distributive laws

Using composite theories:

• Choose theories to compose: $T \circ S$. 
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
  - $\Rightarrow$ terms can be separated:
    $s(t(x, y), t(z, w)) = t'[s'_x/x]$
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
  - $\Rightarrow$ terms can be separated:
    $s(t(x, y), t(z, w)) = t'[s'_x/x]$
- Manipulate terms.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
  - $\Rightarrow$ terms can be separated:
    $$s(t(x, y), t(z, w)) = t'[s'_x/x]$$
- Manipulate terms.
- Derive contradiction of form $x = y$. 

Conclusion: no such theory possible.

Image based on an image from Walter Bispo
Used under the creative commons licence
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
  - $\Rightarrow$ terms can be separated:
    \[
    s(t(x, y), t(z, w)) = t'[s'_x/x]
    \]
- Manipulate terms.
- Derive contradiction of form $x = y$.
- Conclusion: no such theory possible.

Image based on an image from Valter Bispo
Used under the creative commons licence
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
  - $\Rightarrow$ terms can be separated:
    $$s(t(x, y), t(z, w)) = t'[s'_x/x]$$
- Manipulate terms.
- Derive contradiction of form $x = y$.
- Conclusion: no such theory possible.
- List equations in the proof.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $T \circ S$.
- Assume composite theory exists.
  - $\Rightarrow$ terms can be separated:
    $s(t(x, y), t(z, w)) = t'[s'_x/x]$
- Manipulate terms.
- Derive contradiction of form $x = y$.
- Conclusion: no such theory possible.
- List equations in the proof.
- $\Rightarrow$ No-go theorem.
How to: Term Manipulation

Proof that in Rings (Abelian groups after Monoids), \( x \ast 0 = 0 \)

- Start:

\[
x \ast 0 = ?
\]
How to: Term Manipulation

Proof that in Rings (Abelian groups after Monoids), $x * 0 = 0$

- Start:

  $$x * 0 = ?$$

- Substitute 1 for $x$:

  $$1 * 0 = ?[1/x]$$
How to: Term Manipulation

Proof that in Rings (Abelian groups after Monoids), \( x \ast 0 = 0 \)

• Start:

\[
x \ast 0 = ?
\]

• Substitute 1 for \( x \):

\[
1 \ast 0 = ?[1/x]
\]

• Simplify lhs (unit):

\[
0 = ?[1/x]
\]
How to: Term Manipulation

Proof that in Rings (Abelian groups after Monoids), \( x \ast 0 = 0 \)

- Start:
  \[ x \ast 0 = ? \]

- Substitute 1 for \( x \):
  \[ 1 \ast 0 = ?[1/x] \]

- Simplify lhs (unit):
  \[ 0 = ?[1/x] \]

- Two separated terms:
  equality holds in component theories.

\[ \Rightarrow ? = 0 \]
How to: Term Manipulation

Proof that in Gnirs (Monoids after Abelian groups), \( x + 1 = 1 \)

- Start:
  \[ x + 1 = ? \]

- Substitute 0 for \( x \):
  \[ 0 + 1 = ?[0/x] \]

- Simplify lhs (unit):
  \[ 1 = ?[0/x] \]

- Two separated terms:
  equality holds in component theories.

  \[ \Rightarrow ? = 1 \]
Gnirs as a first counterexample

**There is no composite theory of Monoids after Abelian groups.**

**Proof:**
We know: $x + 1 = 1$
We show: $x = 0$
Gnirs as a first counterexample

There is no composite theory of Monoids after Abelian groups.
Proof:
We know: $x + 1 = 1$
We show: $x = 0$

Hence for any two variables: $x = 0 = y$
which means any composite theory is inconsistent.
Gnirs as a first counterexample

There is no composite theory of Monoids after Abelian groups.

Proof:
We know: $x + 1 = 1$
We show: $x = 0$

\[
\begin{align*}
  x & \quad \text{\{} unit \text{\}} \\
  = & \quad x + 0 \\
  \text{\{} inverse \text{\}} & \quad = x + (1 + (-1)) \\
  = & \quad 0 \quad \text{\{} inverse \text{\}} \\
\end{align*}
\]

Hence for any two variables: $x = 0 = y$
which means any composite theory is inconsistent.
Gnirs as a first counterexample

There is no composite theory of Monoids after Abelian groups.

Jon Beck

The composite $\mathbf{ST}$ is the free ring triple. $XST$ is the polynomial ring $\mathbb{Z}[X]$ with the elements of $X$ as noncommuting indeterminates.

The canonical diagram of adjoint functors is:

\[
\begin{array}{ccc}
\end{array}
\]

The scheme is: the distributive law $\ell: TS \longrightarrow ST$ produces the adjoint square, which, being distributive (Section 3), induces a distributive law $\lambda: G_{\text{Ab}}G_{\text{Mon}} \longrightarrow G_{\text{Mon}}G_{\text{Ab}}$.

where $G_{\text{Mon}} = \bar{U} T F^T$, $G_{\text{Ab}} = \text{Hom}_T(F^ST, \ ) \otimes_T FST$. This $\lambda$ is that employed by Barr in his *Composite cotriples*, this volume (Theorem 4.6).

\[\text{A distributive law } ST \longrightarrow TS \text{ would have the air of a universal solution to the problem of factoring polynomials into linear factors. This suggests that the composite } TS \text{ has little chance of being a triple.}\]

(2) **CONSTANTS.** Any set $C$ can be interpreted as a triple in the category of sets, $\mathbf{A}$, via the coproduct injection and folding map $X \longrightarrow C + X, C + C + X \longrightarrow C + X$. $\mathbf{A}^{C+()}$ is the category of sets with $C$ as constants. For example, if $C = 1$, $\mathbf{A}^{1+()}$ is the category of pointed sets.
Some examples from various No-Go Theorems

Powerset $\circ$ Abelian groups

List$^2$

Multiset$^3$

Exception $\circ$ List

Multiset $\circ$ Rings

Powerset$^2$

Distribution$^2$
The crucial step

Composite theories give 2 properties:

- **Separation**
  Start with term \( x \) that is **not** separated:
  \( x = ?, \) where \( ? \) is separated.

- **Equality preservation**
  Needs equality between two separated terms in normal form.

TODO: obtain a separated term from \( x \).
Previously:

\[
x \ast 0 \rightarrow 1 \ast 0 \rightarrow 0
\]

Using, for all \( x \):

\[
1 \ast x = x
\]
The crucial step

Trick: shrinking terms to variables or constants creates separated terms.

\[ s(t(x, y), z) \]

\[ s(x, z) \quad e \quad t(x, y) \]
The crucial step

Trick: shrinking terms to variables or constants creates separated terms.

Units: $x + 0 = x$
Idempotence: $x \times x = x$
Absorption: $x \land (x \lor y) = x$
Inverse: $x + (-x) = 0$

\[ s(t(x, y), z) \]
\[ s(x, z) \quad e \quad t(x, y) \]
The crucial step

**Conjecture 1.** Any theorem that proves the non-existence of a distributive law will involve at least one monad that is presented by an algebraic theory $\mathcal{S}$ for which the following axiom holds:

- $\mathcal{S}$ has an $n$-ary term $s$ ($n \geq 2$), for which there is a substitution $f : \text{var}(s) \to \mathcal{S}$ such that for any $x \in \text{var}(s)$:

$$\Gamma \vdash s[f(y)/y \neq x] =_{\mathcal{S}} x.$$
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad $L$:

$$LX \simeq X + ▶LA$$
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad $L$:

$$LX \simeq X + \triangleright LA$$

- **Powerset** - Mogelberg and Vezzosi 2021.
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad \( L \):

\[
LX \simeq X + \triangle LA
\]

- **Powerset** - Mogelberg and Vezzosi 2021.
  - Difficult: idempotence vs time steps.
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad $L$:

$$LX \simeq X + \triangleright LA$$

- **Powerset** - Møgelberg and Vezzosi 2021.

  - Difficult: idempotence vs time steps.

- **List ✓**
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad $L$:

$$LX \simeq X + \triangleright LA$$

- **Powerset** - Mogelberg and Vezzosi 2021.
  - Difficult: idempotence vs time steps.
- **List ✓**
- **Multiset ✓**
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad $L$:

$$LX \simeq X + \triangleright LA$$

- **Powerset** - Mogelberg and Vezzosi 2021.
  - Difficult: idempotence vs time steps.
- List ✓
- Multiset ✓
- Reader
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad \( L \):

\[ LX \simeq X + \triangle LA \]

- **Powerset** - Mogelberg and Vezzosi 2021.
  - Difficult: idempotence vs time steps.
- **List ✓**
- **Multiset ✓**
- **Reader**
- **State**
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad $L$:

$$LX \simeq X + \triangleright LA$$

- **Powerset** - Møgelberg and Vezzosi 2021.
  - Difficult: idempotence vs time steps.
- List ✓
- Multiset ✓
- Reader
- State
- ...

Difficult: idempotence vs time steps.
The delay monad

Combining algebraic effect with guarded recursion, modeled by the Delay monad \( L \):

\[
LX \simeq X + \triangleright LA
\]

- **Powerset** - Møgelberg and Vezzosi 2021.
  - Difficult: idempotence vs time steps.
- **List ✓**
- **Multiset ✓**
- **Reader**
- **State**
- . . .
- \( \Rightarrow \) General Theory
Conclusion and peek into the future:

- Not all monads compose via a distributive law.
- Algebra provides method to prove counterexamples, which can be generalised to no-go theorems.
- Reducing a term to a variable is a key property for no-go theorems.
Conclusion & What is next

Conclusion and peek into the future:

- Not all monads compose via a distributive law.
- Algebra provides method to prove counterexamples, which can be generalised to no-go theorems.
- Reducing a term to a variable is a key property for no-go theorems.

What I am going to do:

- Combine algebraic effects with guarded recursion.
  - List and Multiset: done
  - Powerset might not be possible.
  - upcoming: reader, state, ...
References


http://www.cs.ox.ac.uk/files/12453/MaaikeZwartDPhilThesis.pdf