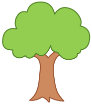
Lessons from failing distributive laws

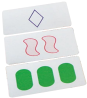
Maaike Zwart

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

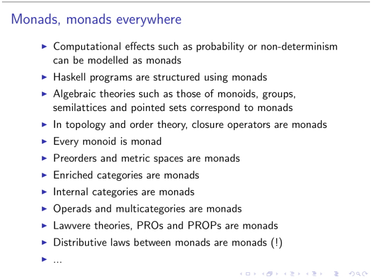
A monad is a categorical structure used for:

- 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:

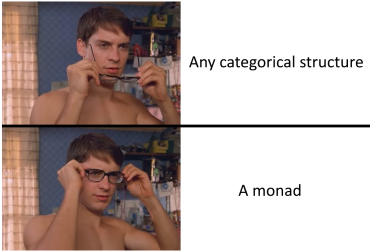
- Modelling of data structures (lists, trees, etc)
- Modelling of computation (exception, reader, writer, etc)



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:

$$
\begin{array}{ccc}
T & \xrightarrow{\eta T} & TT \\
{\scriptstyle T\eta}\downarrow & \searrow^{Id} & \downarrow{\scriptstyle \mu} \\
TT & \xrightarrow{\mu} & T
\end{array}
\qquad\qquad
\begin{array}{ccc}
TTT & \xrightarrow{T\mu} & TT \\
{\scriptstyle \mu T}\downarrow & & \downarrow{\scriptstyle \mu} \\
TT & \xrightarrow{\mu} & T
\end{array}
$$

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:

$$\begin{array}{ccc} T & \xrightarrow{\eta T} & TT \\ {\scriptstyle T\eta}\downarrow & \searrow^{Id} & \downarrow{\scriptstyle \mu} \\ TT & \xrightarrow{\mu} & T \end{array} \qquad \begin{array}{ccc} TTT & \xrightarrow{T\mu} & TT \\ {\scriptstyle \mu T}\downarrow & & \downarrow{\scriptstyle \mu} \\ TT & \xrightarrow{\mu} & T \end{array}$$

Examples:

- List
- Multiset/Bag
- Powerset
- Distribution
- Exception
- Writer
- Reader
- State

Composing Monads

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

Composing Monads

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

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

Composing Monads

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

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

- Same for μ^{TS}?

Composing Monads

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

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

- Same for μ^{TS}?
  - Need:

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

Composing Monads

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

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

- Same for μ^{TS}?
  - Need:

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

  - Have:

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

Composing Monads

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

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

- Same for μ^{TS}?
  - Need:

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

  - Have:

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

  - Solution:

$$\lambda : ST \Rightarrow TS$$

Composing Monads

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

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

- Same for μ^{TS}?
  - Need:

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

  - Have:

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

  - Solution:

$$\lambda : ST \Rightarrow TS$$

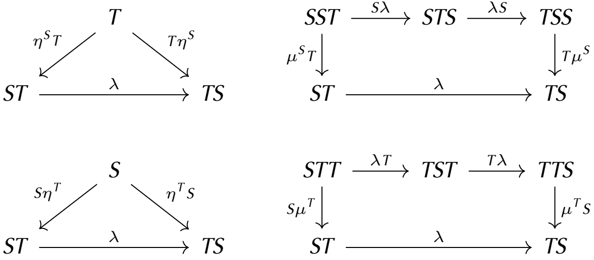
- If λ is a *distributive law*, then the above choices form a monad.

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 $\lambda : ST \to TS$ is a natural transformation satisfying the following axioms.



Examples

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

$$(a + b) * c = a * c + b * c$$
$$[\{a, b\}, \{c\}] \mapsto \{[a, c], [b, c]\}$$

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.
  $\rightarrow$ time consuming.

That sounds easy, but...

Problem:

- Distributive laws are hard to find.
  $\rightarrow$ time consuming.
- Axioms are hard to check (even though they look easy).
  $\rightarrow$ 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 Σ and a set of variables give *terms*.
- Axioms E and equational logic give equivalence of terms.

Monoids:

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

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	$\Longleftrightarrow$	Algebraic theories
Distributive laws	$\Longleftrightarrow$	

The algebraic equivalent of distributive laws

Monads	$\Longleftrightarrow$	Algebraic theories
Distributive laws	$\Longleftrightarrow$	Composite theories *- Piróg and Staton 2017*.

The algebraic equivalent of distributive laws

Monads	$\Longleftrightarrow$	Algebraic theories
Distributive laws	$\Longleftrightarrow$	Composite theories *- Piróg and Staton 2017.*

Example: Composing Abelian groups and Monoids: Rings!

$$\begin{aligned}
\Sigma^R &= \Sigma^A \uplus \Sigma^M \\
&= \{0^{(0)}, 1^{(0)}, -^{(1)}, +^{(2)}, *^{(2)}\} \\
E^R &= E^A \cup E^M \cup \\
&\quad \{a * (b + c) = (a * b) + (a * c) \\
&\quad (a + b) * c = (a * c) + (b * c)\}
\end{aligned}$$

The algebraic equivalent of distributive laws

Monads $\Longleftrightarrow$ Algebraic theories
Distributive laws $\Longleftrightarrow$ Composite theories *- Piróg and Staton 2017.*

- Terms can be separated

$$a * (b + c) = (a * b) + (a * c)$$

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

$$(a * b) + c =_R c + (a * b)$$
$$\Leftrightarrow \; x + c =_A c + x \text{ and } a * b =_M a * b$$

My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $\mathbb{T} \circ \mathbb{S}$.

My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $\mathbb{T} \circ \mathbb{S}$.
- Assume composite theory exists.

My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose theories to compose: $\mathbb{T} \circ \mathbb{S}$.
- 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: $\mathbb{T} \circ \mathbb{S}$.
- 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: $\mathbb{T} \circ \mathbb{S}$.
- 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 $x = y$.

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: $\mathbb{T} \circ \mathbb{S}$.
- 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: $\mathbb{T} \circ \mathbb{S}$.
- 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.

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: $\mathbb{T} \circ \mathbb{S}$.
- 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.

Image based on an image from Valter Bispo
Used under the creative commons licence

How to: Term Manipulation

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

- Start:

$$x * 0 = ?$$

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), $x * 0 = 0$

- Start:

$$x * 0 = ?$$

- Substitute 1 for x:

$$1 * 0 = ?[1/x]$$

- Simplify lhs (unit):

$$0 = ?[1/x]$$

How to: Term Manipulation

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

- Start:

$$x * 0 = ?$$

- Substitute 1 for x:

$$1 * 0 = ?[1/x]$$

- 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:
equalty 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: $x = 0$

$$
\begin{aligned}
& x \\
& \{unit\} \\
= \;& x + 0 \\
& \{inverse\} \\
= \;& x + (1 + (-1)) \\
& \{associativity\} \\
= \;& (x + 1) + (-1) \\
& \{x + 1 = 1\} \\
= \;& 1 + (-1) \\
& \{inverse\} \\
= \;& 0
\end{aligned}
$$

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 108

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

The canonical diagram of adjoint functors is:

[...]

The scheme is: the distributive law $\ell\colon TS \longrightarrow ST$ produces the adjoint square, which, being distributive (Section 3), induces a distributive law $\lambda\colon \mathbf{G_{Ab}}\mathbf{G_{Mon}} \longrightarrow \mathbf{G_{Mon}}\mathbf{G_{Ab}}$, where $G_{\mathbf{Mon}} = \widetilde{U^T}\widetilde{F^T}$, $G_{\mathbf{Ab}} = \mathrm{Hom}_{\mathbf{T}}(F^{ST},\ )\otimes_{\mathbf{T}} F^{ST}$. This λ is that employed by Barr in his *Composite cotriples*, this volume (Theorem 4.6).

A distributive law $ST \longrightarrow TS$ would have the air of a universal solution to the problem of factoring polynomials into linear factors. This suggests that the composite TS 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 $\circ$ Powerset

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:

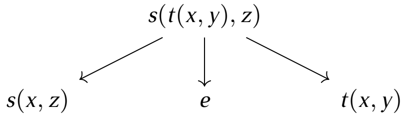
$$x * 0 \to 1 * 0 \to 0$$

Using, for all x:

$$1 * x = x$$

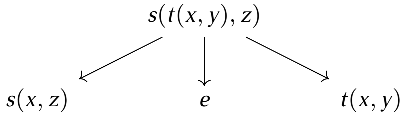
The crucial step

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



The crucial step

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



Units: $x + 0 = x$
Idempotence: $x * x = x$
absorption: $x \wedge (x \vee y) = x$
inverse: $x + (-x) = 0$

$\vdots$

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* $\mathbb{S}$ *for which the following axiom holds:*

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

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

The delay monad

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

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

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.*

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.

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 *- Møgelberg 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 *- Møgelberg 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:

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

- Powerset *- Møgelberg 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
- ...

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
- ...
- ⇒ General Theory

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.

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

J. Beck: Distributive Laws. In: Seminar on Triples and Categorical Homotopy Theory, Springer Berlin Heidelberg, pp. 119-140, 1969.

M. Piróg and S. Staton: Backtracking with cut via a distributive law and left-zero monoids. In: Journal of Functional Programming, vol. 27, 2017.

B. Klin and J. Salamanca: Iterated covariant powerset is not a monad. In: Proceedings of the 34th Conference on the Mathematical Foundations of Programming Semantics (MFPS2018), 2018

R.E. Møgelberg and A. Vezzosi: Two Guarded Recursive Powerdomains for Applicative Simulation. MFPS, 2021.
http://www.cs.ox.ac.uk/files/12453/MaaikeZwartDPhilThesis.pdf