Distributive Laws in the Boom Hierarchy

Maaike Zwart

University of Oxford

2 December 2020
Overview

• Introduction
  • Motivation: monads and monad compositions
  • Reminder: algebraic theories and composites
  • My strategy proving for no-go theorems
• Boom hierarchy: examples and intuition
• Spotlight theorem: too many constants theorem
• 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 (!)
- ...

Intro

The Boom Hierarchy

Too Many Constants Theorem

Conclusion
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: What are they?

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:

\[
\begin{align*}
T & \xrightarrow{\eta_T} TT \\
TT & \xrightarrow{\mu} T
\end{align*}
\]

\[
\begin{align*}
TTT & \xrightarrow{T\mu} TT \\
TT & \xrightarrow{\mu} T
\end{align*}
\]

Examples:
- List
- Multiset/Bag
- Powerset
- Distribution
- Exception
- Writer
- Reader
Monads: What are they?

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:

\[ T \xrightarrow{\eta_T} TT \xrightarrow{\mu} T \]
\[ TTT \xrightarrow{T\mu} TT \]

Examples:
- List
- Multiset/Bag
- Powerset
- Distribution
- Exception
- Writer
- Reader
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.

  - Beck 1969.
Motivation: monads and distributive laws

Problem:

- Distributive laws are hard to find.

  (time consuming)
Motivation: monads and distributive laws

Problem:

- Distributive laws are hard to find.
  (time consuming)
- Axioms are hard to check.
  (resulting in mistakes in the literature!)
Motivation: monads and distributive laws

Problem:

- Distributive laws are hard to find. (time consuming)
- Axioms are hard to check. (resulting in mistakes in the literature!)
- Distributive laws might not even exist.
Motivation: monads and distributive laws

Problem:

- Distributive laws are hard to find. (time consuming)
- Axioms are hard to check. (resulting in mistakes in the literature!)
- Distributive laws might not even exist.
Motivation: monads and distributive laws

Problem:

- Distributive laws are hard to find. (time consuming)
- Axioms are hard to check. (resulting in mistakes in the literature!)
- Distributive laws might not even exist.

What I do:

- Find no-go theorems for distributive laws.
Motivation: monads and distributive laws

Problem:
- Distributive laws are hard to find. (time consuming)
- Axioms are hard to check. (resulting in mistakes in the literature!)
- Distributive laws might not even exist.

What I do:
- Find 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.

Reflexivity: $t = t$

Axiom: $\frac{(s, t) \in E}{s = t}$

Symmetry: $t = t' \Rightarrow t' = t$

Substitution: $t = t' \Rightarrow t[f] = t'[f]$

Transitivity: $t = t', t' = t'' \Rightarrow t = t''$

For any $\sigma$: $t_1 = t'_1, \ldots, t_n = t'_n \Rightarrow \sigma(t_1, \ldots, t_n) = \sigma(t'_1, \ldots, t'_n)$
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:

$\Sigma = \{0^{(0)}, -^{(1)}, +^{(2)}\}$

$E = \{0 + x = x = x + 0,$

$(x + y) + z = x + (y + z),$

$x + y = y + x,$

$x + (-x) = 0 = (-x) + x\}$
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:

\[
\Sigma = \{1^{(0)}, *^{(2)}\}
\]
\[
E = \{1 \ast x = x = x \ast 1, \quad (x \ast y) \ast z = x \ast (y \ast z)\}
\]

Abelian groups:

\[
\Sigma = \{0^{(0)}, -(1), +^{(2)}\}
\]
\[
E = \{0 + x = x = x + 0, \quad (x + y) + z = x + (y + z), \quad x + y = y + x, \quad x + (-x) = 0 = (-x) + x\}
\]

Monads arise from free/forgetful adjunction between Set and category of \((\Sigma, E)\)-algebras.
Composite theories: the equivalent of distributive laws

Example: Rings are a *composite theory*\(^1\) of Abelian groups after Monoids.

Rings:

\[
\Sigma = \Sigma^A \uplus \Sigma^M = \{0^{(0)}, 1^{(0)}, -(1), +^{(2)}, \ast^{(2)}\} \\
E = E^A \cup E^M \cup \\
\{a \ast (b + c) = (a \ast b) + (a \ast c) \\
(a + b) \ast c = (a \ast c) + (b \ast c)\}
\]

\(^1\)Piróg and Staton 2017.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose two theories to compose.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose two theories to compose.
- Assume composite theory exists.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose two theories to compose.
- Assume composite theory exists.
- Manipulate terms.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose two theories to compose.
- Assume composite theory exists.
- Manipulate terms.
- Derive contradiction of form $x = y$. 
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose two theories to compose.
- Assume composite theory exists.
- Manipulate terms.
- Derive contradiction of form $x = y$.
- Conclusion: no such theory possible.
My strategy: no-go theorems for distributive laws

Using composite theories:

- Choose two theories to compose.
- Assume composite theory exists.
- 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 two theories to compose.
- Assume composite theory exists.
- Manipulate terms.
- Derive contradiction of form $x = y$.
- Conclusion: no such theory possible.
- List equations in the proof.
- $\implies$ No-go theorem.
The Boom Hierarchy

The Boom Hierarchy is a set of data structures:

Trees
↓
Lists
↓
Bags
↓
Sets
The Boom Hierarchy

The Boom Hierarchy is a set of data structures:

- Trees
- Lists
- Bags
- Sets

Magmas:

\[ \Sigma = \{ 0^{(0)}, +^{(2)} \}, \text{ (unital equations)} \]
The Boom Hierarchy

The Boom Hierarchy is a set of data structures:

- **Trees**
- **Lists**
- **Bags**
- **Sets**

**Magmas:**

\[ \Sigma = \{0^{(0)}, +^{(2)}\}, \text{ (unital equations)} \]

plus associativity
The Boom Hierarchy is a set of data structures:

- Trees
- Magmas: $\Sigma = \{0^{(0)}, +^{(2)}\}$, (unital equations)
  - plus associativity
- Monoids
  - plus commutativity
- Bags
- Commutative Monoids
- Sets
The Boom Hierarchy

The Boom Hierarchy is a set of data structures:

- **Trees**
- **Lists**
- **Bags**
- **Sets**

**Magmas:**
- \( \Sigma = \{0^{(0)}, +^{(2)}\} \), (unital equations)

- **Monoids**
  - plus associativity

- **Commutative Monoids**
  - plus commutativity

- **Join-semilattices**
  - plus idempotence
The Boom Hierarchy

The Boom Hierarchy is a set of data structures:

- **Trees**
- **Lists**
- **Bags**
- **Sets**

**Magmas:**
- **Monoids**
- **Commutative Monoids**
- **Join-semilattices**

**Why this hierarchy?**

- Practical Monads
- Simple Theories
- Interesting Properties
A first look

<table>
<thead>
<tr>
<th>The Boom Hierarchy</th>
<th>Possibility of compositions</th>
<th>Column: ( \circ )</th>
<th>Row: ( \circ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>Lists</td>
<td>Bags</td>
<td>Sets</td>
</tr>
<tr>
<td>Trees</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Lists</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Bags</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Sets</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

- Manes and Mulry 2007, 2008
- Klin and Salamanca 2018
- Zwart and Marsden 2019, 2020 (under review)
A first look

The Boom Hierarchy

Too Many Constants Theorem

Conclusion

- Manes and Mulry 2007, 2008
- Klin and Salamanca 2018
- Zwart and Marsden 2019, 2020 (under review)
Extending the Boom hierarchy

- Boom Hierarchy: 4 structures (8 if non-empty are considered)
Extending the Boom hierarchy

- Boom Hierachy: 4 structures (8 if non-empty are considered)
- Extension: all combinations of axioms gives 16 structures.
Extending the Boom hierarchy

- Boom Hierarchy: 4 structures (8 if non-empty are considered)
- Extension: all combinations of axioms gives 16 structures.
  - Unit (U): Y/N
Extending the Boom hierarchy

- Boom Hierarchy: 4 structures (8 if non-empty are considered)
- Extension: all combinations of axioms gives 16 structures.
  - Unit (U): Y/N
  - Associativity (A): Y/N
Extending the Boom hierarchy

- Boom Hierarchy: 4 structures (8 if non-empty are considered)
- Extension: all combinations of axioms gives 16 structures.
  - Unit (U): Y/N
  - Associativity (A): Y/N
  - Commutativity (C): Y/N
Extending the Boom hierarchy

- Boom Hierarchy: 4 structures (8 if non-empty are considered)
- Extension: all combinations of axioms gives 16 structures.
  - Unit (U): Y/N
  - Associativity (A): Y/N
  - Commutativity (C): Y/N
  - Idempotence (I): Y/N
Extending the Boom hierarchy

- Boom Hierarchy: 4 structures (8 if non-empty are considered)
- Extension: all combinations of axioms gives 16 structures.
  - Unit (U): Y/N
  - Associativity (A): Y/N
  - Commutativity (C): Y/N
  - Idempotence (I): Y/N
- UAC stands for a structure with signature $\Sigma = \{0^{(0)}, +^{(2)}\}$ and the equations Unit, Associativity, Commutativity: bags.
A dramatic result

Combinations where both structures have units:

<table>
<thead>
<tr>
<th></th>
<th>U</th>
<th>UI</th>
<th>UC</th>
<th>UCI</th>
<th>UA</th>
<th>UAI</th>
<th>UAC</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UC</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UCI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UA</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UAI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UAC</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UACI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Venturing into unknown territory

The non-empty equivalents are more promising:

### The Extended Boom Hierarchy (2/4)

#### Possibility of compositions Column • Row.

<table>
<thead>
<tr>
<th></th>
<th>Ø</th>
<th>I</th>
<th>C</th>
<th>CI</th>
<th>A</th>
<th>AI</th>
<th>AC</th>
<th>ACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ø</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CI</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>AI</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>AC</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ACI</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
### Extended Boom Hierarchy, showing which compositions of form Column \( \circ \) Row are possible.

Theories all consist of one binary operator, that is possibly idempotent (I), commutative (C), and/or associative (A), with possibly a constant that satisfies the unit equations. (U)

\( Y \) indicates a successful composition, \( N \) indicates that the composition is impossible, empty cells represent unknowns. My own contributions have been highlighted in green.

<table>
<thead>
<tr>
<th></th>
<th>( \emptyset )</th>
<th>I</th>
<th>C</th>
<th>CI</th>
<th>A</th>
<th>AI</th>
<th>AC</th>
<th>ACI</th>
<th>U</th>
<th>UI</th>
<th>UC</th>
<th>UCI</th>
<th>UA</th>
<th>UAI</th>
<th>UAC</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \emptyset )</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>AC</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>U</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UC</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UA</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UAI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UAC</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UACI</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Lessons from the Boom hierarchy

• Idempotence/Units are bad.
Lessons from the Boom hierarchy

- Idempotence/Units are bad.
  (but not always)
Lessons from the Boom hierarchy

- Idempotence/Units are bad.
  (but not always)

\[ x \times x = x \quad \text{and} \quad x \times 1 = x \]
Lessons from the Boom hierarchy

- Idempotence/Units are bad.
  (but not always)

\[ x \ast x = x \quad x \ast 1 = x \quad x \lor (x \land y) = x \]
Lessons from the Boom hierarchy

- Idempotence/Units are bad. (but not always)
  \[ x \ast x = x \quad x \ast 1 = x \quad x \lor (x \land y) = x \]

- Key property: reducing a term to a variable.
Lessons from the Boom hierarchy

• Idempotence/Units are bad.
  (but not always)
  \[ x \times x = x \quad x \times 1 = x \quad x \lor (x \land y) = x \]

• Key property: reducing a term to a variable.

• Conjecture:
  Equations that reduce a term to a variable are necessary for distributive laws to fail.
  (but not sufficient)
Predictions

Extended Boom Hierarchy, showing which compositions of form Column \( \circ \) Row are possible.

Theories all consist of one binary operator, that is possibly idempotent (I), commutative (C), and/or associative (A), with possibly a constant that satisfies the unit equations. (U) Y indicates a successful composition, N indicates that the composition is impossible, empty cells represent unknowns. My own contributions have been highlighted in green.

<table>
<thead>
<tr>
<th></th>
<th>( \emptyset )</th>
<th>I</th>
<th>C</th>
<th>CI</th>
<th>A</th>
<th>AI</th>
<th>AC</th>
<th>ACI</th>
<th>U</th>
<th>UI</th>
<th>UC</th>
<th>UCI</th>
<th>UA</th>
<th>UAI</th>
<th>UAC</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \emptyset )</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>I</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>AI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>AC</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ACI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>U</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UC</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UC1</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UAI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>UAC</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>UACI</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
When it’s just too much

Too Many Constants Theorem
When it's just too much

Too Many Constants Theorem

But first, a proposition.
An important proposition

We need an interaction law:

**Proposition (Multiplicative Zeroes)**

Let $S$ be an algebraic theory with a term $s$ such that:

- $s$ can be reduced to a variable via a substitution. e.g. $x \cdot y$ with idempotent, $x \cdot p \cdot y \cdot q$ with absorption, etc.
- $T$ is an algebraic theory with a constant 0 such that:

Any composite theory $U$ of $T$ after $S$ has the following interaction:

For any $x p \var s q$: $s r 0 \{ x s U 0 \}$. 
An important proposition

We need an interaction law:

**Proposition (Multiplicative Zeroes)**

Let $S$ be an algebraic theory with a term $s$ such that:
- $s$ can be reduced to a variable via a substitution.
An important proposition

We need an interaction law:

**Proposition (Multiplicative Zeroes)**

Let $S$ be an algebraic theory with a term $s$ such that:
- $s$ can be reduced to a variable via a substitution.
  
  e.g. $x \ast y$ with $\ast$ idempotent, $x \lor (y \land z)$ with absorption, etc.
An important proposition

We need an interaction law:

**Proposition (Multiplicative Zeroes)**

Let $S$ be an algebraic theory with a term $s$ such that:

- $s$ can be reduced to a variable via a substitution.
  
  e.g. $x \ast y$ with $\ast$ idempotent, $x \lor (y \land z)$ with absorption, etc.

And let $T$ be an algebraic theory with a constant $0$ such that:
An important proposition

We need an interaction law:

**Proposition (Multiplicative Zeroes)**

Let $S$ be an algebraic theory with a term $s$ such that:
- $s$ can be reduced to a variable via a substitution.
  - e.g. $x * y$ with $*$ idempotent, $x \lor (y \land z)$ with absorption, etc.

And let $T$ be an algebraic theory with a constant 0 such that:

$$t[f] =_T 0 \Rightarrow t =_T 0$$
An important proposition

We need an interaction law:

**Proposition (Multiplicative Zeroes)**

Let $S$ be an algebraic theory with a term $s$ such that:
- $s$ can be reduced to a variable via a substitution.
  - e.g. $x * y$ with * idempotent, $x \lor (y \land z)$ with absorption, etc.

And let $T$ be an algebraic theory with a constant 0 such that:

$$t[f] =_T 0 \Rightarrow t =_T 0$$

Any composite theory $U$ of $T$ after $S$ has the following interaction:
For any $x \in \text{var}(s)$:

$$s[0/x] =_U 0.$$
Theorem (No-Go Theorem: Too Many Constants)

Let $\mathcal{S}$ be an algebraic theory with a term $s$ such that:

- $s$ can be reduced to a variable via a substitution.
- $s$ has two or more free variables.

And let $\mathcal{T}$ be an algebraic theory with at least two constants $0, 1$ such that for both constants:

$$t[f] =_{\mathcal{T}} 0 \Rightarrow t =_{\mathcal{T}} 0 \quad t[f] =_{\mathcal{T}} 1 \Rightarrow t =_{\mathcal{T}} 1$$

Then there exists no composite theory of $\mathcal{T}$ after $\mathcal{S}$. 
Proof of the Too Many Constants Theorem

Proof.

Suppose that $U$ is a composite theory of $T$ after $S$
Proof of the Too Many Constants Theorem

Proof.

Suppose that \( \mathcal{U} \) is a composite theory of \( \mathcal{T} \) after \( \mathcal{S} \)
Then by Proposition 1 we have:
Proof of the Too Many Constants Theorem

Proof.

Suppose that $U$ is a composite theory of $T$ after $S$. Then by Proposition 1 we have:

$$s(x, y)$$
Proof of the Too Many Constants Theorem

Proof.

Suppose that \( U \) is a composite theory of \( T \) after \( S \). Then by Proposition 1 we have:

\[ s(x, y)[0/x, 1/y] \]
Proof of the Too Many Constants Theorem

Proof.

Suppose that $U$ is a composite theory of $T$ after $S$. Then by Proposition 1 we have:

$$0 = U s(x, y)[0/x, 1/y] = U 1.$$
Proof of the Too Many Constants Theorem

Proof.
Suppose that $U$ is a composite theory of $T$ after $S$.
Then by Proposition 1 we have:

$$0 =_U s(x, y)[0/x, 1/y] =_U 1.$$  

Contradiction. So $U$ cannot be a composite of $T$ after $S$. 

$\square$
Another extension of the Boom hierarchy

<table>
<thead>
<tr>
<th></th>
<th>Trees</th>
<th>Lists</th>
<th>Bags</th>
<th>Sets</th>
<th>BT</th>
<th>BL</th>
<th>BB</th>
<th>ST</th>
<th>SL</th>
<th>SB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lists</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bags</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sets</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Another extension of the Boom hierarchy

<table>
<thead>
<tr>
<th></th>
<th>Trees</th>
<th>Lists</th>
<th>Bags</th>
<th>Sets</th>
<th>BT</th>
<th>BL</th>
<th>BB</th>
<th>ST</th>
<th>SL</th>
<th>SB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Lists</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Bags</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Sets</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Another extension of the Boom hierarchy

### Iterating Compositions in the Boom Hierarchy

**Possibility of compositions Column × Row.**

<table>
<thead>
<tr>
<th></th>
<th>Trees</th>
<th>Lists</th>
<th>Bags</th>
<th>Sets</th>
<th>BT</th>
<th>BL</th>
<th>BB</th>
<th>ST</th>
<th>SL</th>
<th>SB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Lists</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Bags</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Sets</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BT</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BL</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BB</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>ST</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SL</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SB</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Conclusion

- Not all monads compose via a distributive law.

- https://www.cs.ox.ac.uk/people/maaike.zwart/
- maaike.annebeth@gmail.com
Conclusion

- Not all monads compose via a distributive law.
- Boom hierarchy provides some intuition.

- https://www.cs.ox.ac.uk/people/maaike.zwart/
- maaike.annebeth@gmail.com
Conclusion

- Not all monads compose via a distributive law.
- Boom hierarchy provides some intuition.
- Reducing a term to a variable key property for no-go theorems.

- https://www.cs.ox.ac.uk/people/maaike.zwart/
- maaike.annebeth@gmail.com
Conclusion

- Not all monads compose via a distributive law.
- Boom hierarchy provides some intuition.
- Reducing a term to a variable key property for no-go theorems.
- Too many constants / multiplicative zeroes prevent iterated distributive laws within the Boom hierarchy.

- https://www.cs.ox.ac.uk/people/maaike.zwart/
- maaike.annebeth@gmail.com