# On the Cartwright-Felleisen-Wadler conjecture

Ohad Kammar
University of Oxford

**Dylan McDermott**
University of Cambridge

HOPE 2017

# Extensible semantics

- Traditional semantics: the meaning of a program changes when we add something to the language
- *Extensible* semantics [Reynolds '74, Cartwright and Felleisen '94]: meaning should be stable under language extension

# Extensible semantics

- Traditional semantics: the meaning of a program changes when we add something to the language
- *Extensible* semantics [Reynolds '74, Cartwright and Felleisen '94]: meaning should be stable under language extension

Can we do extensible monadic semantics?

# Extensible monadic semantics [Wadler '98]

- Given a signature $\Sigma$
  - e.g. $\{\text{read} : 1 \to V,\ \text{write} : V \to 1\}$
- A monad $T_\varepsilon$ for $\varepsilon \subseteq \Sigma$
  - And a monad morphism $T_\varepsilon \to T_{\varepsilon'}$ for $\varepsilon \subseteq \varepsilon'$

Interpret terms $\Gamma \vdash M : A$ that only use effects in $\varepsilon$ as

$$\varepsilon [\![ M ]\!] : [\![ \Gamma ]\!] \to T_\varepsilon [\![ A ]\!]$$

Adding to $\Sigma$ doesn't change the semantics of a given program!

# This talk: constructing extensible semantics

Given a non-extensible semantics (with monad $T$), can we give an extensible semantics?

# This talk: constructing extensible semantics

Given a non-extensible semantics (with monad $T$), can we give an extensible semantics?

Yes: just choose $T_\varepsilon = T$.

- Doesn't help us reason about smaller parts of the language

# This talk: constructing extensible semantics

Given a non-extensible semantics (with monad $T$), can we give an extensible semantics?

Yes: just choose $T_\varepsilon = T$.

- Doesn't help us reason about smaller parts of the language

If $\varepsilon$ is smaller than $\Sigma$ then $T_\varepsilon$ should be "simpler" than $T$.

Example: if $T$ is the state+continuations monad

$$(V \Rightarrow - \Rightarrow R) \Rightarrow V \Rightarrow R$$

$T_{\{\text{write}\}}$ should be

$$(1 + V) \times -$$

(if $|V|, |R| > 1$)

# This talk: constructing extensible semantics

## Goal

Give a construction that:

- Gives us the best possible monads $T_\varepsilon$
- Is general: works for as many effects as possible
- Constructs a model with the right behaviour:

$$\varepsilon [\![ M ]\!] = \varepsilon [\![ N ]\!] \Leftrightarrow [\![ M ]\!] = [\![ N ]\!]$$

# Related work

- Cartwright and Felleisen '94: non-monadic extensible semantics
- Katsumata '14: give a construction for free monads. Uses a more general notion of effect system
- Kammar '14: gives a construction for algebraic $T$
  - Based on factorizations of morphisms of Lawvere theories

# Factoring monad morphisms

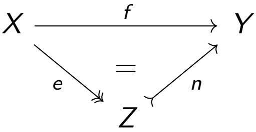
## Definition (Factorization system)

A *factorization system* for the category $\mathcal{C}$ consists of:

- A class $\mathcal{E}$ of morphisms $e : X \twoheadrightarrow Y$ ("epis")
- A class $\mathcal{M}$ of morphisms $m : X \rightarrowtail Y$ ("monos")

such that:

- Every morphism $f : X \to Y$ can be factored into an epi followed by a mono:

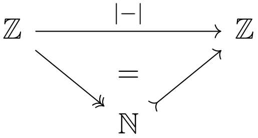



- Some other conditions hold

# Factoring monad morphisms

## Examples of factorization systems

- **Set**: surjections and injections




- $\omega$**Cpo**: dense epis (closure of image equals domain) and full monos ($n\,x \sqsubseteq n\,y \Rightarrow x \sqsubseteq y$)
- Presheaves: componentwise surjections and componentwise injections

# Factoring monad morphisms

## Theorem

*Let $m : S \to T$ be a strong monad morphism, and factorize $m$ componentwise:*

$$
\begin{array}{ccc}
SX & \xrightarrow{\quad m_X \quad} & TX \\
 & \underset{e_X}{\searrow\!\!\!\!\searrow} \quad = \quad \underset{n_X}{\nearrow} & \\
 & RX &
\end{array}
$$

*If $\mathcal{E}$ is closed under $S$ and products then:*

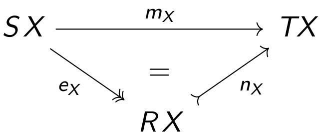
- *$R$ is a strong monad*
- *$e$ and $n$ are strong monad morphisms*

*For every $\mathrm{op}_S : [\![A]\!] \to S\,[\![B]\!]$ we can define $\mathrm{op}_R := e_{[\![B]\!]} \circ op_S$.*

# Factoring monad morphisms

**Theorem**

*Let $m : S \to T$ be a strong monad morphism, and factorize $m$ componentwise:*




*If $\mathcal{E}$ is closed under $S$ and products then:*

- *$R$ is a strong monad*
- *$e$ and $n$ are strong monad morphisms*

*For every* $\mathrm{op}_S : [\![A]\!] \to S\,[\![B]\!]$ *we can define* $\mathrm{op}_R := e_{[\![B]\!]} \circ op_S$.

We are given $T$, but what should $S$ and $m$ be?

# Using free monads

Want to choose $S$ and $m$ so that $R$:

- Supports exactly the operations in $\varepsilon$
- Behaves like $T$ (i.e. $\varepsilon [\![M]\!] = \varepsilon [\![N]\!] \Leftrightarrow [\![M]\!] = [\![N]\!]$)

# Using free monads

Want to choose $S$ and $m$ so that $R$:

- Supports exactly the operations in $\varepsilon$
- Behaves like $T$ (i.e. $\varepsilon\,[\![M]\!] = \varepsilon\,[\![N]\!] \Leftrightarrow [\![M]\!] = [\![N]\!]$)

Use the free monad for $\varepsilon$:

- Epi $\Rightarrow$ $R$ and $S$ have exactly the same operations
- Mono $\Rightarrow$ $R$ behaves like $T$?
  - This depends on the factorization system

We need the free monad to preserve epis.

# Using free monads

### Theorem

*Suppose that $\mathcal{C}$ has directed colimits and $F : C \to C$ preserves them. If $F$ also preserves $\mathcal{E}$-morphisms then the free monad preserves epis.*

Use

$$F = \sum_{(\mathrm{op}:A \to B) \in \varepsilon} A \times (B \Rightarrow (-))$$

to get the free monad we want

Get $m$ from initiality of the free monad

# Examples

## In **Set**:

- If $T$ is state+continuations:

$$T_\emptyset = \mathrm{Id} \qquad T_{\{\mathrm{read},\mathrm{write}\}} = V \Rightarrow V \times -$$
$$T_{\{\mathrm{read}\}} = V \Rightarrow - \qquad T_{\{\mathrm{write}\}} = (1 + V) \times -$$

- Non-example: can't get writer+nondeterminism from state+nondeterminism

## In $\omega$**Cpo**:

- If $T$ is exceptions+partiality then $T_{\{\mathrm{diverge}\}}$ is partiality

## Presheaves:

- If $T$ is local state [Plotkin and Power '02] then

$$T_{\{\mathrm{read},\mathrm{write}\}}\, n\, X \;=\; V^n \Rightarrow V^n \times X\, n$$

# Correctness

We want $\varepsilon \llbracket M \rrbracket = \varepsilon \llbracket N \rrbracket \Leftrightarrow \llbracket M \rrbracket = \llbracket N \rrbracket$

Need a notion of predicate

- Factorization systems of interest induce fibrations [Hughes and Jacobs '03]
- What does a suitable factorization system look like?

Anything else?

- Reynolds uses projection theorems
- Partial maps between non-extensible and extensible semantics

# How general is the construction?

- The category needs enough structure, including a suitable factorization system
  - All of our examples have this (but others might not!)
- We don't assume anything about $T$
  - This works for *arbitrary* effects
  - But $\Sigma$ contains only Kleisli arrows
- We only consider effects
  - Can't add/remove other language features (e.g. linear types)
- More interesting effect systems?

# Future work

- Correctness proofs using fibrations
- How easy is it to use the construction?
- More examples: full ground state, probability, ...

# Conclusions

- Can construct extensible semantics from non-extensible semantics
- Construction is general
  - No restrictions on $T$
- Still work in progress – don't know if the extensible semantics is correct!