Intensionality, Intensional Recursion, and the Gödel-Löb axiom

Alex Kavvos
Department of Computer Science, University of Oxford

IMLA 2017, 18 July 2017

arXiv:1702.01288

What this talk is about

- It is about using **modal logic**, to present a **typing discipline** for **programs-as-data**.
- It is about investigating the central rule/ axiom of **provability logic** in this setting.

What this talk is about

Curry-Howard isomorphism
for intensional programming

- It is about using **modal logic**, to present a **typing discipline** for **programs-as-data**.

- It is about investigating the central rule/axiom of **provability logic** in this setting.

What this talk is about

Curry-Howard isomorphism
for intensional programming

- It is about using **modal logic**, to present a **typing discipline** for **programs-as-data**.
- It is about investigating the central rule/axiom of **provability logic** in this setting.

intensional
recursion

Programs-as-data

- More than just '**functions as first-class-citizens**.'
  - The **extensional paradigm**: a program can call a functional argument *at a finite set of points*.
- Instead, very close to the idea of **Gödel numbering**.
  - The **intensional paradigm**: a program can inspect the source code of its functional argument, and can do rather arbitrary things with it (inspect, simulate, deconstruct, count its symbols...).
- Non-functional operations.
- Homoiconicity: when one does not need coding at all; e.g. LISP.

Programs-as-data

- More than just '**functions as first-class-citizens**.'
  - The **extensional paradigm**: a program can call a functional argument *at a finite set of points*.
- Instead, very close to the idea of **Gödel numbering**.
  - The **intensional paradigm**: a program can inspect the source code of its functional argument, and can do rather arbitrary things with it (inspect, simulate, deconstruct, count its symbols...).

How can we do this in a **typed**, **well-structured**, **safe**, **coding-free** manner?

Intensional Recursion

- A very strong kind of recursion, discovered by Kleene in 1938. For CS, lost in the mists of time (Abramsky).

- In the **untyped λ-calculus**:

First Recursion Theorem	$\forall f \in \Lambda.\ \exists u \in \Lambda.\ u = f\ u$
Second Recursion Theorem	$\forall f \in \Lambda.\ \exists u \in \Lambda.\ u = f\ \ulcorner u \urcorner$
Enumeration Theorem	$\exists \mathbf{E} \in \Lambda.\ \forall u \in \Lambda^0.\ \mathbf{E}\ \ulcorner u \urcorner = u$

Given EN, the SRT implies the FRT, hence it is stronger.

But **what does it really do?**

Types for Intensionality

- Strangely, intensionality follows a typing discipline.
- Suppose $u : A$; let's say $\ulcorner u \urcorner : \Box A$
- Then well-known combinators of λ-calculus that perform operations on Gödel numbers acquire types; e.g.

$$\mathbf{gnum}\ \ulcorner M \urcorner = \ulcorner \ulcorner M \urcorner \urcorner \qquad \mathbf{E}\ \ulcorner M \urcorner = M$$

$$\mathbf{app}\ \ulcorner M \urcorner\ \ulcorner N \urcorner = \ulcorner M\ N \urcorner$$

Types for Intensionality

- Strangely, intensionality follows a typing discipline.
- Suppose $u : A$; let's say $\ulcorner u \urcorner : \Box A$
- Then well-known combinators of λ-calculus that perform operations on Gödel numbers acquire types; e.g.

$$\mathbf{gnum} : \Box A \to \Box\Box A$$

$$\mathbf{gnum} \ulcorner M \urcorner = \ulcorner\ulcorner M \urcorner\urcorner \qquad \mathbf{E} \ulcorner M \urcorner = M$$

$$\mathbf{app} \ulcorner M \urcorner \ulcorner N \urcorner = \ulcorner M\ N \urcorner$$

Types for Intensionality

- Strangely, intensionality follows a typing discipline.
- Suppose $u : A$; let's say $\ulcorner u \urcorner : \Box A$
- Then well-known combinators of λ-calculus that perform operations on Gödel numbers acquire types; e.g.

$$\mathbf{gnum} : \Box A \to \Box\Box A$$

$$\mathbf{gnum}\ \ulcorner M \urcorner = \ulcorner\ulcorner M \urcorner\urcorner \qquad\qquad \mathbf{E}\ \ulcorner M \urcorner = M$$

$$\mathbf{app}\ : \Box(A \to B) \to \Box A \to \Box B$$

$$\mathbf{app}\ \ulcorner M \urcorner\ \ulcorner N \urcorner = \ulcorner M\ N \urcorner$$

Types for Intensionality

- Strangely, intensionality follows a typing discipline.
- Suppose $u : A$; let's say $\ulcorner u \urcorner : \Box A$
- Then well-known combinators of λ-calculus that perform operations on Gödel numbers acquire types; e.g.

$$\mathbf{gnum} : \Box A \to \Box\Box A \qquad \mathbf{E} : \Box A \to A$$

$$\mathbf{gnum}\ \ulcorner M \urcorner = \ulcorner \ulcorner M \urcorner \urcorner \qquad \mathbf{E}\ \ulcorner M \urcorner = M$$

$$\mathbf{app} : \Box(A \to B) \to \Box A \to \Box B$$

$$\mathbf{app}\ \ulcorner M \urcorner\ \ulcorner N \urcorner = \ulcorner M\ N \urcorner$$

Types for Intensionality

- Strangely, intensionality follows a typing discipline.
- Suppose $u : A$; let's say $\ulcorner u \urcorner : \Box A$
- Then well-known combinators of λ-calculus that perform operations on Gödel numbers acquire types; e.g.

$$\mathbf{gnum} : \Box A \to \Box\Box A \qquad \mathbf{E} : \Box A \to A$$

$$\mathbf{gnum} \ulcorner M \urcorner = \ulcorner\ulcorner M \urcorner\urcorner \qquad \mathbf{E} \ulcorner M \urcorner = M$$

$$\mathbf{app} : \Box(A \to B) \to \Box A \to \Box B$$

$$\mathbf{app} \ulcorner M \urcorner \ulcorner N \urcorner = \ulcorner M\ N \urcorner$$

It's **S4**!

Types for Intensional Recursion

Types for Intensional Recursion

- Take $u : A$ so that $u = f \ulcorner u \urcorner$

Types for Intensional Recursion

- Take $u : A$ so that $u = f \ulcorner u \urcorner$
- Then it is forced that $f : \Box A \to A$

Types for Intensional Recursion

- Take $u : A$ so that $u = f \ulcorner u \urcorner$
- Then it is forced that $f : \Box A \to A$
- Yields the following

Types for Intensional Recursion

- Take $u : A$ so that $u = f \ulcorner u \urcorner$
- Then it is forced that $f : \Box A \to A$
- Yields the following

Logical interpretation of the Second Recursion Theorem

$$\frac{f : \Box A \to A}{u : A}$$

Types for Intensional Recursion

- Take $u : A$ so that $u = f \ulcorner u \urcorner$
- Then it is forced that $f : \Box A \to A$
- Yields the following

Logical interpretation of the Second Recursion Theorem

$$\frac{f : \Box A \to A}{u : A}$$

... such that $u = f \ulcorner u \urcorner$

Types for Intensional Recursion

- Take $u : A$ so that $u = f \ulcorner u \urcorner$
- Then it is forced that $f : \Box A \to A$
- Yields the following

Logical interpretation of the Second Recursion Theorem

$$\frac{f : \Box A \to A}{\ulcorner u \urcorner : \Box A}$$

... such that $u = f \ulcorner u \urcorner$

Prospectus

- We will first revisit Davies & Pfenning's **S4**.
- We will add **intensional operations** to it.
- Then we will add **intensional recursion**.
- The resulting system is called **Intensional PCF**.

Prospectus

- We will first revisit Davies & Pfenning's **S4**.
- We will add **intensional operations** to it.
- Then we will add **intensional recursion**.
- The resulting system is called **Intensional PCF**.

THEOREM.
'Full' reduction of Intensional PCF is **confluent**.
Hence, Intensional PCF is consistent.

1. Curry-Howard and **S4**

Curry-Howard

Annotate sequents with **proof terms** (= 'summary' of derivation).

$$\dfrac{\dfrac{\vdots}{\Gamma \vdash A} \qquad \dfrac{\vdots}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B} \quad \longrightarrow \quad \dfrac{\dfrac{\vdots}{\Gamma \vdash M : A} \qquad \dfrac{\vdots}{\Gamma \vdash N : B}}{\Gamma \vdash \langle M, N \rangle : A \times B}$$

$$\dfrac{\dfrac{\vdots}{\Gamma \vdash A \times B}}{\Gamma \vdash A} \quad \longrightarrow \quad \dfrac{\dfrac{\vdots}{\Gamma \vdash M : A \times B}}{\Gamma \vdash \pi_1(M) : A}$$

$$\pi_1(\langle M, N \rangle) \to M$$

Curry-Howard

Annotate sequents with **proof terms** (= 'summary' of derivation).

$$\dfrac{\dfrac{\vdots}{\Gamma, A \vdash B}}{\Gamma \vdash A \to B} \quad \longrightarrow \quad \dfrac{\dfrac{\vdots}{\Gamma, x : A \vdash M : B}}{\Gamma \vdash \lambda x.M : A \to B}$$

$$\dfrac{\dfrac{\vdots}{\Gamma \vdash A \to B} \qquad \dfrac{\vdots}{\Gamma \vdash A}}{\Gamma \vdash B} \quad \longrightarrow \quad \dfrac{\dfrac{\vdots}{\Gamma \vdash M : A \to B} \qquad \dfrac{\vdots}{\Gamma \vdash N : A}}{\Gamma \vdash M\,N : B}$$

$$(\lambda x.M)N \to M[N/x]$$

Dual-context systems

- A kind of natural deduction with **two contexts,** introduced by Girard, developed by many over the 1990s (Davies and Pfenning, Andreoli, Wadler, Barber and Plotkin, ...)

- Judgments:

intuitionistic assumptions

$$\Delta \,;\Gamma \vdash M : A$$

modal assumptions

In our setting Δ = code/intensional variables,
Γ = value/extension variables.

The Modal Rules

$$\frac{\Delta\,;\cdot \vdash M : A}{\Delta\,;\Gamma \vdash \text{box } M : \Box A}\,(\Box\mathcal{I}) \qquad \frac{}{\Delta, u{:}A, \Delta'\,;\Gamma \vdash u : A}\,(\Box\text{var})$$

$$\frac{\Delta\,;\Gamma \vdash M : \Box A \qquad \Delta, u{:}A\,;\Gamma \vdash N : C}{\Delta\,;\Gamma \vdash \text{let box } u \Leftarrow M \text{ in } N : C}\,(\Box\mathcal{E})$$

- Hiding Δ, it looks just like simply-typed λ-calculus.
- This is augmented with the reduction

$$\text{let box } u \Leftarrow \text{box } M \text{ in } N \longrightarrow N[M/u]$$

The Modal Rules

$$\frac{\Delta\,;\cdot \vdash M : A}{\Delta\,;\Gamma \vdash \mathsf{box}\ M : \Box A}\ (\Box\mathcal{I}) \qquad \frac{}{\Delta, u{:}A, \Delta'\,;\Gamma \vdash u : A}\ (\Box\mathsf{var})$$

$$\frac{\Delta\,;\Gamma \vdash M : \Box A \qquad \Delta, u{:}A\,;\Gamma \vdash N : C}{\Delta\,;\Gamma \vdash \mathsf{let\ box}\ u \Leftarrow M\ \mathsf{in}\ N : C}\ (\Box\mathcal{E})$$

THEOREM (Davies-Pfenning). This system captures **S4**; satisfies all the expected structural rules; and is confluent and strongly normalising.

$$\mathsf{let\ box}\ u \Leftarrow \mathsf{box}\ M\ \mathsf{in}\ N \longrightarrow N[M/u]$$

Example

$$\mathbf{app} \equiv \lambda f.\ \lambda x.\ \mathsf{let\ box}\ u \Leftarrow f\ \mathsf{in\ let\ box}\ v \Leftarrow x\ \mathsf{in\ box}\ (u\ v)$$

$$\vdash \mathbf{app}\ : \Box(A \to B) \to \Box A \to \Box B$$

$$\mathbf{app}\ (\mathsf{box}\ F)(\mathsf{box}\ M) \longrightarrow^{*} \mathsf{box}\ (F M)$$

S4

- Davies and Pfenning defined the above system for **homogeneous, staged metaprogramming** (POPL 1996), which was also implemented and tested.

- The purpose of that language was to separate the **static** and **dynamic** phases: some (modal) things would happen at compile-time, some (intuitionistic) things at run-time.

- But, even though mentioned in the paper (MSCS 2001), **intensionality** is completely absent! Everything is functional.

II. Intensional operations

Intensional Operations

The quintessential example: **is the term an application?**

$$\mathbf{is\text{-}app}\ (\mathsf{box}\ PQ) \to \mathsf{true}$$

$$\mathbf{is\text{-}app}\ (\mathsf{box}\ M) \to \mathsf{false} \qquad \text{if} \qquad M \not\equiv PQ$$

This function can almost be considered a **criterion of intensionality**.

Intensional Operations: first attempt

- Let's suppose **any** function on terms $f : \mathcal{T}(A) \to \mathcal{T}(B)$ can can be included as a constant $\tilde{f} : \Box A \to \Box B$

$$\tilde{f}(\text{box } M) \to \text{box } f(M)$$

This is not confluent.

$$\text{let box } u \Leftarrow \text{box } PQ \text{ in } \mathbf{is\text{-}app}\ (\text{box } u)$$

$$\mathbf{is\text{-}app}\ (\text{box } PQ) \qquad\qquad \text{let box } u \Leftarrow \text{box } PQ \text{ in false}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\text{true} \qquad\qquad \text{false}$$

Intensional Operations, second attempt

- The problem: $M \longrightarrow N$ yet $M[P/u] \not\longrightarrow N[P/u]$
- Violated because of constants like **is-app**.
- How to fix? Consider **substitutive** intensional operations $f : \mathcal{T}(A) \to \mathcal{T}(B)$ such that $f(N[P/u]) \equiv f(N)[P/u]$
- Indeed a fix. But a standard **naturality argument** yields
$$f(P) \equiv f(u[P/u]) \equiv f(u)[P/u]$$
- So already defined by $\tilde{f} = \lambda x.\ \text{let box}\ u \Leftarrow x\ \text{in box}\ f(u)$
$\tilde{f}(\text{box}\ M) \longrightarrow^* \text{box}\ f(u)[M/u] \equiv \text{box}\ f(u[M/u]) \equiv \text{box}\ f(M)$
- ... so we have achieved precisely **nothing**.

Intensional Operations, with success

- Solution: **restrict everything to closed terms**:

$$\mathcal{T}(A) = \{\, M \mid \cdot\,;\cdot \vdash M : A \,\}$$

- ... and for each $f : \mathcal{T}(A) \to \mathcal{T}(B)$

  add a constant $\tilde{f} : \Box A \to \Box B$

  with reduction $\tilde{f}(\text{box}\, M) \to \text{box}\, f(M)$

  which happens **only when M is closed**.

- It so happens that this is confluent! Will see in a moment.

III. Intensional Recursion

Löb's rule

Without further ado:

$$\frac{\Delta\,;\quad \Box A \vdash \quad A}{\Delta\,;\Gamma \vdash \quad \Box A}$$

Observation by Abramsky: if one erases the boxes, it's PCF!

We use this form, prompted by proof-theoretic considerations (see K, LICS 2017).

Löb's rule

Without further ado:

$$\frac{\Delta\,;\, z : \Box A \vdash M : A}{\Delta\,;\,\Gamma \vdash \mathsf{fix}\ z\ \mathsf{in}\ \mathsf{box}\ M : \Box A}$$

Observation by Abramsky: if one erases the boxes, it's PCF!

We use this form, prompted by proof-theoretic considerations (see K, LICS 2017).

Löb's rule

Without further ado:

$$\frac{\Delta\,;\, z : \Box A \vdash M : A}{\Delta\,;\,\Gamma \vdash \mathsf{fix}\ z\ \mathsf{in}\ \mathsf{box}\ M : \Box A}$$

$$\mathsf{fix}\ z\ \mathsf{in}\ \mathsf{box}\ M \longrightarrow \mathsf{box}\ M[\mathsf{fix}\ z\ \mathsf{in}\ \mathsf{box}\ M/z]$$

Observation by Abramsky: if one erases the boxes, it's PCF!

We use this form, prompted by proof-theoretic considerations (see K, LICS 2017).

Löb's rule

Without further ado:

$$\frac{\Delta\,;z:\quad A \vdash M : A}{\Delta\,;\Gamma \vdash \mathsf{fix}\ z\ \mathsf{in}\qquad M :\quad A}$$

$$\mathsf{fix}\ z\ \mathsf{in}\qquad M \longrightarrow\qquad M[\mathsf{fix}\ z\ \mathsf{in}\qquad M/z]$$

Observation by Abramsky: if one erases the boxes, it's PCF!

We use this form, prompted by proof-theoretic considerations (see K, LICS 2017).

An objection

"But Löb's rule, in conjunction with **S4**,
means that every type is inhabited!"

Indeed, if we let

$$\mathsf{eval}_A \equiv \lambda x.\ \mathsf{let\ box}\ u \Leftarrow x\ \mathsf{in}\ u$$
$$\Omega_A \equiv \mathsf{fix}\ z\ \mathsf{in\ box}\ \mathsf{eval}_A\ z$$

then $\vdash \Omega_A : \Box A$ and hence $\vdash \mathsf{eval}_A\ \Omega_A : A$

with $\Omega_A \to \mathsf{box}\,(\mathsf{eval}_A\ \Omega_A)$ and $\mathsf{eval}_A \Omega_A \to^* \mathsf{eval}_A\ \Omega_A$

Answer: **It's OK**. If we want general recursion, which the SRT gives, there will be non-normalising terms. Like PCF, not a logic but a programming language: **the terms still matter**.

Confluence

- As long as we do not admit $\dfrac{M \longrightarrow N}{\mathsf{box}\ M \longrightarrow \mathsf{box}\ N}$,

THEOREM. The resulting system is confluent, and hence consistent.

- The proof uses the standard parallel reduction method of Tait and Martin-Löf.

- The fact that intensional operations only reduce when the term is closed is crucial to the argument.

Conclusions

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**

- We proved that it is **confluent**, and hence **consistent.** Price to pay: intensional operations on **closed terms only** (a thorn known to metaprogrammers…).

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**

- We proved that it is **confluent**, and hence **consistent.** Price to pay: intensional operations on **closed terms only** (a thorn known to metaprogrammers...).

- The type of the **Gödel-Löb axiom** is inhabited by an an intensional fixed point combinator. The standard fixed point combinator (Y) is **definable in the system**.

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**
- We proved that it is **confluent**, and hence **consistent.** Price to pay: intensional operations on **closed terms only** (a thorn known to metaprogrammers...).
- The type of the **Gödel-Löb axiom** is inhabited by an an intensional fixed point combinator. The standard fixed point combinator (Y) is **definable in the system**.
- Open questions:

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**
- We proved that it is **confluent**, and hence **consistent.** Price to pay: intensional operations on **closed terms only** (a thorn known to metaprogrammers...).
- The type of the **Gödel-Löb axiom** is inhabited by an an intensional fixed point combinator. The standard fixed point combinator (Y) is **definable in the system**.
- Open questions:
  - which are the correct primitives?

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**
- We proved that it is **confluent**, and hence **consistent.** Price to pay: intensional operations on **closed terms only** (a thorn known to metaprogrammers...).
- The type of the **Gödel-Löb axiom** is inhabited by an an intensional fixed point combinator. The standard fixed point combinator (Y) is **definable in the system**.
- Open questions:
  - which are the correct primitives?
  - what is the expressivity of this system? what does it do?

Conclusions

- We have constructed a **modal λ-calculus**, inspired by **S4** and **GL**, with **intensional (non-functional) operations**, and **intensional recursion.**
- We proved that it is **confluent**, and hence **consistent.** Price to pay: intensional operations on **closed terms only** (a thorn known to metaprogrammers…).
- The type of the **Gödel-Löb axiom** is inhabited by an an intensional fixed point combinator. The standard fixed point combinator (Y) is **definable in the system**.
- Open questions:
  - which are the correct primitives?
  - what is the expressivity of this system? what does it do?

Thank you for your attention.