Algebraic Lambda-calculus

Renato Neves

Universidade do Minho

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Table of Contents

Recalling Lambda-calculus

$$\frac{x : \mathbb{A} \in \Gamma}{\Gamma \vdash x : \mathbb{A}} \qquad \frac{}{\Gamma \vdash * : 1} \qquad \frac{\Gamma \vdash v : \mathbb{A} \times \mathbb{B}}{\Gamma \vdash \pi_1\, v : \mathbb{A}}$$

$$\frac{\Gamma \vdash v : \mathbb{A} \qquad \Gamma \vdash u : \mathbb{B}}{\Gamma \vdash \langle v, u \rangle : \mathbb{A} \times \mathbb{B}} \qquad \frac{\Gamma, x : \mathbb{A} \vdash v : \mathbb{B}}{\Gamma \vdash \lambda x : \mathbb{A}.\, v : \mathbb{A} \to \mathbb{B}}$$

$$\frac{\Gamma \vdash v : \mathbb{A} \to \mathbb{B} \quad \Gamma \vdash u : \mathbb{A}}{\Gamma \vdash v\, u : \mathbb{B}}$$

Sequential Composition Revisited

A "native" deductive rule

$$\frac{\Gamma \vdash v : \mathbb{A} \qquad x : \mathbb{A} \vdash u : \mathbb{B}}{\Gamma \vdash x \leftarrow v; u : \mathbb{B}}$$

It reads *"bind the computation v to x and then run u"*

Interpretation is defined as

$$\frac{[\![\Gamma \vdash v : \mathbb{A}]\!] = f \qquad [\![x : \mathbb{A} \vdash u : \mathbb{B}]\!] = g}{[\![\Gamma \vdash x \leftarrow v; u : \mathbb{B}]\!] = g \cdot f}$$

Table of Contents

Signatures

Signature

A set $\Sigma = \{(\sigma_1, n_1), (\sigma_2, n_2), \ldots\}$ of operations σ_i paired with the number of inputs n_i they are supposed to receive

These constitute the aforementioned algebraic operations

Examples

- Exceptions : $\{(\mathrm{e}, 0)\}$
- Read a bit from the environment : $\{(\mathrm{read}, 2)\}$
- Wait calls : $\{(\mathrm{wait}_n, 1) \mid n \in \mathbb{N}\}$
- Non-deterministic choice : $\{(+, 2)\}$
- Continuous trajectories : ...

Algebraic Operations in Lambda-calculus

We take a signature Σ of operations and introduce a new rule

$$\frac{(\sigma, n) \in \Sigma \qquad \forall 1 \leq i \leq n.\ \Gamma \vdash m_i : \mathbb{A}}{\Gamma \vdash \sigma(m_1, \ldots, m_n) : \mathbb{A}}$$

Examples

- $x : \mathbb{A} \vdash \mathrm{wait}_1(x) : \mathbb{A}$ – adds delay of one second to returning x
- $\Gamma \vdash \mathrm{e}() : \mathbb{A}$ – raises an exception e
- $\Gamma \vdash \mathrm{write}_v(m) : \mathbb{A}$ – writes v in memory and then runs m
- $x : \mathbb{A} \times \mathbb{A} \vdash \mathrm{read}(\pi_1\, x, \pi_2\, x) : \mathbb{A}$ – receives a bit: if 0 returns $\pi_1\, x$ and $\pi_2\, x$ otherwise

Exercise

Define a λ-term with variable x that requests a bit from the user and depending on the value read it returns x with either one or two seconds of delay.

Table of Contents

The Million-Dollar Question

How to provide semantics to such programming languages ?

The short answer: via monads !!

The long answer: see the next slides :-) . . .

The Core Idea

Programs $\Gamma \vdash v : \mathbb{A}$ interpreted as functions

$$[\![\Gamma \vdash v : \mathbb{A}]\!] : [\![\Gamma]\!] \longrightarrow [\![\mathbb{A}]\!]$$

... and there exists only one function of type

$$[\![\Gamma]\!] \longrightarrow [\![1]\!]$$

Problem: it is then necessarily the case that

$$[\![\Gamma \vdash x : 1]\!] = [\![\Gamma \vdash \mathrm{wait}_1(x) : 1]\!]$$

despite these programs having different execution times

The Core Idea

Interpreted $\Gamma \vdash v : \mathbb{A}$ as a function

$$[\![\Gamma \vdash v : \mathbb{A}]\!] : [\![\Gamma]\!] \longrightarrow [\![\mathbb{A}]\!]$$

But values now come with effects . . .

Instead of having $[\![\mathbb{A}]\!]$ as set of outputs, we will have a set $T[\![\mathbb{A}]\!]$ of effectful values

$$[\![\Gamma \vdash m : \mathbb{A}]\!] : [\![\Gamma]\!] \longrightarrow T[\![\mathbb{A}]\!]$$

T should thus be a set-constructor: given a set of outputs X it returns a set of effectful values TX over X

Examples

For wait calls, the corresponding set-constructor T is defined as

$$X \mapsto \mathbb{N} \times X$$

i.e. values in X paired with an execution time

For exceptions, the corresponding set-constructor T is defined as

$$X \mapsto X + \{e\}$$

i.e. values in X plus an element e representing the exception

Another Problem

This idea of a set-constructor T looks good, but . . .

. . . it breaks sequential composition

$$\begin{aligned} [\![\Gamma \vdash m : \mathbb{A}]\!] &: [\![\Gamma]\!] \longrightarrow T[\![\mathbb{A}]\!] \\ [\![x : \mathbb{A} \vdash n : \mathbb{B}]\!] &: [\![\mathbb{A}]\!] \longrightarrow T[\![\mathbb{B}]\!] \end{aligned}$$

Another Problem

This idea of a set-constructor T looks good, but ...

... it breaks sequential composition

$$\begin{aligned} [\![\Gamma \vdash m : \mathbb{A}]\!] &: [\![\Gamma]\!] \longrightarrow T[\![\mathbb{A}]\!] \\ [\![x : \mathbb{A} \vdash n : \mathbb{B}]\!] &: [\![\mathbb{A}]\!] \longrightarrow T[\![\mathbb{B}]\!] \end{aligned}$$

We'll need convert a function $h : X \to TY$ into one of type

$$h^\star : TX \to TY$$

Examples

There are set-constructors T for which this is possible

In the case of wait-calls

$$\frac{f : X \to TY = \mathbb{N} \times Y}{f^\star(n, x) = (n + m, y) \text{ where } f(x) = (m, y)}$$

In the case of exceptions

$$\frac{f : X \to TY = Y + \{e\}}{f^\star(x) = f(y) \qquad f^\star(e) = e}$$

Testing the Idea . . .

$$
\begin{aligned}
&[\![x : 1 \vdash y \leftarrow \mathrm{wait}_1(x); \mathrm{wait}_2(y) : 1]\!] \\
&= \; [\![y : 1 \vdash \mathrm{wait}_2(y) : 1]\!]^* \cdot [\![x : 1 \vdash \mathrm{wait}_1(x) : 1]\!] \\
&= \; (v \mapsto (2, v))^* \; \cdot \; (v \mapsto (1, v)) \\
&= \; v \mapsto (3, v)
\end{aligned}
$$

Yet Another Problem

Idea of interpreting λ-terms $\Gamma \vdash m : \mathbb{A}$ as functions

$$[\![\Gamma \vdash m : \mathbb{A}]\!] : [\![\Gamma]\!] \longrightarrow T[\![\mathbb{A}]\!]$$

looks good but it presupposes that all terms invoke effects

Some terms do not do this, *e.g.*

$$[\![x : \mathbb{A} \vdash x : \mathbb{A}]\!] : [\![\mathbb{A}]\!] \longrightarrow [\![\mathbb{A}]\!]$$

Solution

$T[\![\mathbb{A}]\!]$ should include effect-free values, and we should have

$$\eta_{[\![\mathbb{A}]\!]} : [\![\mathbb{A}]\!] \longrightarrow T[\![\mathbb{A}]\!]$$

which maps a value to its effect-free representation

Examples

In the case of wait-calls

$$\frac{TX = \mathbb{N} \times X}{\eta_X(x) = (0, x)}$$

(*i.e.* no wait call invoked)

In the case of exceptions

$$\frac{TX = X + \{e\}}{\eta_X(x) = x}$$

(*i.e.* no exception e raised)

Monads Unlocked !!

Previous analysis leads to the notion of a monad

Monad

Triple $(T, \eta, (-)^\star)$ where T is a set-constructor, η a function $\eta_X : X \to TX$ for each set X, and $(-)^\star$ an operation

$$\frac{f : X \to TY}{f^\star : TX \to TY}$$

s.t. the following laws hold: $\eta^\star = \mathrm{id}$, $f^\star \cdot \eta = f$, $(f^\star \cdot g)^\star = f^\star \cdot g^\star$

These laws are required to forbid "weird" computational behaviour

Exercises

Show that the set-constructor

$$X \mapsto \mathbb{N} \times X$$

can be equipped with a monadic structure

Show that the set-constructor

$$X \mapsto X + 1$$

can be equipped with a monadic structure

Exercises

Show that the set-constructor

$$X \mapsto \{U \mid U \subseteq X\}$$

can be equipped with a monadic structure

Show that the set-constructor

$$X \mapsto \{\mu \mid \mu : X \to [0,1] \text{ a distribution}\}$$

can be equipped with a monadic structure

To Keep In Mind

Recall that,

- we fixed a signature Σ of algebraic operations
- we now have monads at our disposal
- Programs $\Gamma \vdash v : \mathbb{A}$ can be seen either as functions of type $[\![\Gamma]\!] \to [\![\mathbb{A}]\!]$ or of type $[\![\Gamma]\!] \to T[\![\mathbb{A}]\!]$

Semantics

Types $\mathbb{A}$ interpreted as sets $[\![\mathbb{A}]\!]$

$$[\![1]\!] = \{\star\} \qquad [\![\mathbb{A} \times \mathbb{B}]\!] = [\![\mathbb{A}]\!] \times [\![\mathbb{B}]\!] \qquad [\![\mathbb{A} \to \mathbb{B}]\!] = (T[\![\mathbb{B}]\!])^{[\![\mathbb{A}]\!]}$$

Typing contexts Γ interpreted as

$$[\![\Gamma]\!] = [\![x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n]\!] = [\![\mathbb{A}_1]\!] \times \cdots \times [\![\mathbb{A}_n]\!]$$

For each $(\sigma, n) \in \Sigma$ and set X we postulate the existence of a map

$$[\![\sigma]\!]_X : (TX)^n \longrightarrow TX$$

Semantics

$$\frac{x_i : \mathbb{A} \in \Gamma}{[\![\Gamma \vdash x_i]\!] = \pi_i} \qquad \frac{}{[\![\Gamma \vdash *]\!] = !} \qquad \frac{[\![\Gamma \vdash v : \mathbb{A}]\!] = f \qquad [\![\Gamma \vdash u : \mathbb{B}]\!] = g}{[\![\Gamma \vdash \langle v, u \rangle : \mathbb{A} \times \mathbb{B}]\!] = \langle f, g \rangle}$$

$$\frac{[\![\Gamma, x : \mathbb{A} \vdash_{\mathsf{c}} m : \mathbb{B}]\!] = f}{[\![\Gamma \vdash \lambda x : \mathbb{A}.\, m : \mathbb{A} \to \mathbb{B}]\!] = \lambda f} \qquad \frac{[\![\Gamma \vdash v : \mathbb{A} \times \mathbb{B}]\!] = f}{[\![\Gamma \vdash \pi_1\, v : \mathbb{A}]\!] = \pi_1 \cdot f}$$

. . . .

$$\frac{[\![\Gamma \vdash v : \mathbb{A}]\!] = f}{[\![\Gamma \vdash_{\mathsf{c}} \mathtt{return}\; v : \mathbb{A}]\!] = \eta \cdot f} \qquad \frac{[\![\Gamma \vdash_{\mathsf{c}} m : \mathbb{A}]\!] = f \qquad [\![x : \mathbb{A} \vdash_{\mathsf{c}} n : \mathbb{B}]\!] = g}{[\![\Gamma \vdash_{\mathsf{c}} x \leftarrow m\, ;\, n : \mathbb{B}]\!] = g^\star \cdot f}$$

$$\frac{[\![\Gamma \vdash v : \mathbb{A} \to \mathbb{B}]\!] = f \quad [\![\Gamma \vdash u : \mathbb{A}]\!] = g}{[\![\Gamma \vdash_{\mathsf{c}} v\, u : \mathbb{B}]\!] = \mathrm{app} \cdot \langle f, g \rangle}$$

$$\frac{(\sigma, n) \in \Sigma \qquad \forall 1 \leq i \leq n.\, [\![\Gamma \vdash_{\mathsf{c}} m_i : \mathbb{A}]\!] = f_i}{[\![\Gamma \vdash_{\mathsf{c}} \sigma(m_1, \ldots m_n) : \mathbb{A}]\!] = [\![\sigma]\!]_{[\![\mathbb{A}]\!]} \cdot \langle f_1, \ldots, f_n \rangle}$$

Table of Contents

Exercise

Build a λ-term that receives a function $f : \mathbb{A} \to \mathbb{A}$, a value $x : \mathbb{A}$, and applies f to x twice

In classical λ-calculus such would be defined as

$$f : \mathbb{A} \to \mathbb{A}, x : \mathbb{A} \vdash f(f\ x) : \mathbb{A}$$

Sharing Contexts

It will be useful to have two programs in sequential composition that are able share contexts

I.e. it will be useful to have the following rule for seq. composition

$$\frac{\Gamma \vdash_{\mathsf{c}} m : \mathbb{A} \qquad \Gamma, x : \mathbb{A} \vdash_{\mathsf{c}} n : \mathbb{B}}{\Gamma \vdash_{\mathsf{c}} x \leftarrow m \,;\, n : \mathbb{B}}$$

It will be useful to have two programs in sequential composition that are able share contexts

I.e. it will be useful to have the following rule for seq. composition

$$\frac{\Gamma \vdash_{c} m : \mathbb{A} \qquad \Gamma, x : \mathbb{A} \vdash_{c} n : \mathbb{B}}{\Gamma \vdash_{c} x \leftarrow m \,;\, n : \mathbb{B}}$$

This would allow us to solve the previous exercise quite easily

$$f : \mathbb{A} \to \mathbb{A}, x : \mathbb{A} \vdash_{c} y \leftarrow f(x) \,;\, f(y) : \mathbb{A}$$

Sharing Contexts

Natural way of interpreting the rule

$$\frac{[\![\Gamma \vdash_{\mathsf{c}} m : \mathbb{A}]\!] = f \qquad [\![\Gamma, x : \mathbb{A} \vdash_{\mathsf{c}} n : \mathbb{B}]\!] = g}{[\![\Gamma \vdash_{\mathsf{c}} x \leftarrow m \,;\, n : \mathbb{B}]\!] = g^{\star} \cdot \langle \mathrm{id}, f \rangle}$$

but $\langle \mathrm{id}, f \rangle : [\![\Gamma]\!] \to [\![\Gamma]\!] \times T[\![\mathbb{A}]\!]$ and $g^{\star} : T([\![\Gamma]\!] \times [\![\mathbb{A}]\!]) \to T[\![\mathbb{B}]\!]$

Need to find a suitable function

$$\mathrm{str} : [\![\Gamma]\!] \times T[\![\mathbb{A}]\!] \longrightarrow T([\![\Gamma]\!] \times [\![\mathbb{A}]\!])$$

Sharing Contexts

Natural way of interpreting the rule

$$\frac{[\![\Gamma \vdash_{\mathsf{c}} m : \mathbb{A}]\!] = f \qquad [\![\Gamma, x : \mathbb{A} \vdash_{\mathsf{c}} n : \mathbb{B}]\!] = g}{[\![\Gamma \vdash_{\mathsf{c}} x \leftarrow m \,;\, n : \mathbb{B}]\!] = g^\star \cdot \langle \mathrm{id}, f \rangle}$$

but $\langle \mathrm{id}, f \rangle : [\![\Gamma]\!] \to [\![\Gamma]\!] \times T[\![\mathbb{A}]\!]$ and $g^\star : T([\![\Gamma]\!] \times [\![\mathbb{A}]\!]) \to T[\![\mathbb{B}]\!]$

Need to find a suitable function

$$\mathrm{str} : [\![\Gamma]\!] \times T[\![\mathbb{A}]\!] \longrightarrow T([\![\Gamma]\!] \times [\![\mathbb{A}]\!])$$

. . . and there is actually a natural way of doing such !

Tensorial Strength

For every monad T and function $f : X \to Y$ we can build

$$Tf = (\eta \cdot f)^\star : TX \to TY$$

Moreover for every $x \in X$ we can define

$$\mathsf{id}_x : Y \to X \times Y, \quad y \mapsto (x, y)$$

From these we define the strength of T

$$\mathrm{str} : X \times TY \to T(X \times Y), \quad (x, t) \mapsto (T\mathsf{id}_x)(t)$$

$$\frac{[\![\Gamma \vdash_{\mathsf{c}} m : \mathbb{A}]\!] = f \qquad [\![\Gamma, x : \mathbb{A} \vdash_{\mathsf{c}} n : \mathbb{B}]\!] = g}{[\![\Gamma \vdash_{\mathsf{c}} x \leftarrow m \,;\, n : \mathbb{B}]\!] = g^\star \cdot \mathrm{str} \cdot \langle \mathsf{id}, f \rangle}$$

Exercises

Given an explicit definition for the tensorial strength of

- the monad of exceptions,
- the monad of durations,
- the powerset monad,
- the distributions monad

Exercises

Consider the λ-term

$$f : \mathbb{A} \to \mathbb{A}, x : \mathbb{A} \vdash_{\mathsf{c}} y \leftarrow f(x) \,;\, f(y) : \mathbb{A}$$

What is its execution time when

$$f \text{ is given by } (v \mapsto (1, v))$$

Semantics

$$\frac{x_i : \mathbb{A} \in \Gamma}{[\![\Gamma \vdash x_i]\!] = \pi_i} \qquad \frac{}{[\![\Gamma \vdash *]\!] = !} \qquad \frac{[\![\Gamma \vdash v : \mathbb{A}]\!] = f \qquad [\![\Gamma \vdash u : \mathbb{B}]\!] = g}{[\![\Gamma \vdash \langle v, u\rangle : \mathbb{A} \times \mathbb{B}]\!] = \langle f, g\rangle}$$

$$\frac{[\![\Gamma, x : \mathbb{A} \vdash_c m : \mathbb{B}]\!] = f}{[\![\Gamma \vdash \lambda x : \mathbb{A}.\, m : \mathbb{A} \to \mathbb{B}]\!] = \lambda f} \qquad \frac{[\![\Gamma \vdash v : \mathbb{A} \times \mathbb{B}]\!] = f}{[\![\Gamma \vdash \pi_1\, v : \mathbb{A}]\!] = \pi_1 \cdot f}$$

. . . .

$$\frac{[\![\Gamma \vdash v : \mathbb{A}]\!] = f}{[\![\Gamma \vdash_c \texttt{return}\ v : \mathbb{A}]\!] = \eta \cdot f} \qquad \frac{[\![\Gamma \vdash_c m : \mathbb{A}]\!] = f \qquad [\![\Gamma, x : \mathbb{A} \vdash_c n : \mathbb{B}]\!] = g}{[\![\Gamma \vdash_c x \leftarrow m\, ;\, n : \mathbb{B}]\!] = g^\star \cdot \mathrm{str} \cdot \langle \mathrm{id}, f\rangle}$$

$$\frac{[\![\Gamma \vdash v : \mathbb{A} \to \mathbb{B}]\!] = f \quad [\![\Gamma \vdash u : \mathbb{A}]\!] = g}{[\![\Gamma \vdash_c v\, u : \mathbb{B}]\!] = \mathrm{app} \cdot \langle f, g\rangle}$$

$$\frac{(\sigma, n) \in \Sigma \qquad \forall 1 \le i \le n.\, [\![\Gamma \vdash_c m_i : \mathbb{A}]\!] = f_i}{[\![\Gamma \vdash_c \sigma(m_1, \dots m_n) : \mathbb{A}]\!] = [\![\sigma]\!]_{[\![\mathbb{A}]\!]} \cdot \langle f_1, \dots, f_n\rangle}$$