A brief overview of simply-typed λ-calculus

Renato Neves

Universidade do Minho

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Architecture and Calculi Course Unit

Table of Contents

Architecture & Calculi

Software architecture pertains to the way software is structured

i.e. it abstracts from how building blocks are implemented and focusses instead on how the blocks connect to each other

Software calculi refers to algebraic mechanisms for proving that one program is equivalent to each other,

analogously to what we already do for arithmetic expressions (e.g. $3 + 2 = 5$)

The three topics of this course

1. Labelled transition systems and process algebra: focus on distributed systems comprised of processing units that communicate with each other

2. Adaption and application of the previous notions to real-time and cyber-physical systems

3. Programming with algebraic effects: emphasis on compositionality (via typing mechanisms), equational calculus, and uniform approach to different computational effects

Table of Contents

Overview

Modern programming typically involves different effects

- memory cell manipulation
- communication
- exception raising operations
- probabilistic operations
- real-time behaviour
- cyber-physical behaviour

In the following lectures we will study the mathematical foundations of

Programming with effects

in a uniform way

Table of Contents

Deductive reasoning

The process of reasoning from a set of assumptions and logical rules to obtain new knowledge

If every crow is black and x is a crow then x is black

Deductive reasoning has been studied in the last millenina, long before the age of computers

So what does it got to do with programming?

A rule-based system for deductive reasoning


Propositions

Let $\mathbb{A}, \mathbb{B}, \mathbb{C} \ldots$ denote propositions and 1 denote a special proposition that is always true. If $\mathbb{A}$ and $\mathbb{B}$ are propositions:

- then $\mathbb{A} \times \mathbb{B}$ is a proposition; it denotes the conjunction of $\mathbb{A}$ and $\mathbb{B}$
- $\mathbb{A} \to \mathbb{B}$ is a proposition; it denotes that $\mathbb{A}$ implies $\mathbb{B}$

...

A rule-based system for deductive reasoning

Let Γ denote a list of propositions. Then,

$$\frac{\mathbb{A} \in \Gamma}{\Gamma \vdash \mathbb{A}} \qquad \frac{}{\Gamma \vdash 1} \qquad \frac{\Gamma \vdash \mathbb{A} \times \mathbb{B}}{\Gamma \vdash \mathbb{A}} \qquad \frac{\Gamma \vdash \mathbb{A} \times \mathbb{B}}{\Gamma \vdash \mathbb{B}}$$

$$\frac{\Gamma \vdash \mathbb{A} \qquad \Gamma \vdash \mathbb{B}}{\Gamma \vdash \mathbb{A} \times \mathbb{B}} \qquad \frac{\Gamma, \mathbb{A} \vdash \mathbb{B}}{\Gamma \vdash \mathbb{A} \to \mathbb{B}} \qquad \frac{\Gamma \vdash \mathbb{A} \to \mathbb{B} \quad \Gamma \vdash \mathbb{A}}{\Gamma \vdash \mathbb{B}}$$

$\Gamma \vdash \mathbb{A}$ reads as "if the propositions in Γ hold then we deduce that $\mathbb{A}$ holds".

Exercise

Show that if both $\Gamma \vdash \mathbb{A}$ and $\Gamma, \mathbb{A} \vdash \mathbb{B}$ then $\Gamma \vdash \mathbb{B}$. Show also that $\mathbb{A} \times \mathbb{B} \vdash \mathbb{B} \times \mathbb{A}$.

Building new rules from the original ones

The following rules are derivable from the previous system:

$$\frac{\Gamma \vdash \mathbb{A}}{\Gamma, \mathbb{B} \vdash \mathbb{A}} \qquad \frac{\Gamma, \mathbb{A}, \mathbb{B}, \Delta \vdash \mathbb{C}}{\Gamma, \mathbb{B}, \mathbb{A}, \Delta \vdash \mathbb{C}}$$

Homework

Show that if $\Gamma \vdash \mathbb{A} \to \mathbb{B}$ and $\Gamma \vdash \mathbb{B} \to \mathbb{C}$ then $\Gamma \vdash \mathbb{A} \to \mathbb{C}$.

Going back to programming . . .

The essentials of programming

In order to study effectful programming, we should think of what are the basic features of (higher-order) programming ...

- variables
- function application
- function abstraction
- pairing ...

and base our study on the simplest programming language containing these features ...

Simply-typed λ-calculus

It is the basis of Haskell, ML, Eff, F#, Agda, Elm and many other programming languages.

Simply-typed λ-calculus

Types: $\mathbb{A} \ni 1 \mid \mathbb{A} \times \mathbb{A} \mid \mathbb{A} \to \mathbb{A}$

Programs are built according to the deduction rules that were previously presented:

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

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

$$\frac{\Gamma \vdash V : \mathbb{A} \to \mathbb{B} \quad \Gamma \vdash U : \mathbb{A}}{\Gamma \vdash V\, U : \mathbb{B}}$$

Γ is now a non-repetitive list of typed variables $x_1 : \mathbb{A}_1 \ldots x_n : \mathbb{A}_n$

Examples of λ-terms

$x : \mathbb{A} \vdash x : \mathbb{A}$ (identity)

$x : \mathbb{A} \vdash \langle x, x \rangle : \mathbb{A} \times \mathbb{A}$ (duplication)

$x : \mathbb{A} \times \mathbb{B} \vdash \langle \pi_2\ x, \pi_1\ x \rangle : \mathbb{B} \times \mathbb{A}$ (swap)

$f : \mathbb{A} \to \mathbb{B}, g : \mathbb{B} \to \mathbb{C} \vdash \lambda x : \mathbb{A}.\ g(f\ x) : \mathbb{A} \to \mathbb{C}$ (composition)

Exercise

Build a λ-term $f : \mathbb{A} \to \mathbb{A}, x : \mathbb{A} \vdash ? : \mathbb{A}$ that takes a variable $f : \mathbb{A} \to \mathbb{A}$, a variable $x : \mathbb{A}$, and applies f to x twice.

Semantics for simply-typed λ-calculus

We wish to assign a mathematical meaning to λ-terms

$$[\![-]\!] : \lambda\text{-Terms} \longrightarrow \ldots$$

so that we can reason about them in a rigorous way, and take advantage of known mathematical theories

Semantics for simply-typed λ-calculus

We wish to assign a mathematical meaning to λ-terms

$$[\![-]\!] : \lambda\text{-Terms} \longrightarrow \ldots$$

so that we can reason about them in a rigorous way, and take advantage of known mathematical theories

This is the goal of the next slides: we will study how to interpret λ-terms as functions. But first ...

Basic facts about functions

For every set X, there is a 'trivial' function

$$! : X \longrightarrow \{\star\} = 1, \qquad !(x) = \star$$

We can always pair two functions $f : X \to A$, $g : X \to B$ into

$$\langle f, g \rangle : X \to A \times B, \qquad \langle f, g \rangle(x) = (f\ x, g\ x)$$

Consider two sets X, Y. There exist 'projection' functions

$$\pi_1 : X \times Y \to X, \qquad \pi_1(x, y) = x$$
$$\pi_2 : X \times Y \to Y, \qquad \pi_2(x, y) = y$$

Basic facts about functions

We can always 'curry' a function $f : X \times Y \to Z$ into

$$\lambda f : X \to Z^Y, \qquad \lambda f(a) = (b \mapsto f(a, b))$$

Consider sets X, Y, Z. There exists an 'application' function

$$\mathrm{app} : Z^Y \times Y \to Z, \qquad \mathrm{app}(f, y) = f\ y$$

Functional semantics for the simply-typed λ-calculus

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

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

A typing context Γ is interpreted as

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

A λ-term $\Gamma \vdash V : \mathbb{A}$ is interpreted as a function

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

Functional semantics for the simply-typed λ-calculus

A λ-term $\Gamma \vdash V : \mathbb{A}$ is interpreted as a function

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

in the following way:

$$\frac{x_i : \mathbb{A} \in \Gamma}{[\![\Gamma \vdash x_i : \mathbb{A}]\!] = \pi_i} \qquad \frac{}{[\![\Gamma \vdash * : 1]\!] = !} \qquad \frac{[\![\Gamma \vdash V : \mathbb{A} \times \mathbb{B}]\!] = f}{[\![\Gamma \vdash \pi_1 V : \mathbb{A}]\!] = \pi_1 \cdot f}$$

$$\frac{[\![\Gamma \vdash V : \mathbb{A}]\!] = f \qquad [\![\Gamma \vdash U : \mathbb{B}]\!] = g}{[\![\Gamma \vdash \langle V, U \rangle : \mathbb{A} \times \mathbb{B}]\!] = \langle f, g \rangle} \qquad \frac{[\![\Gamma, x : \mathbb{A} \vdash V : \mathbb{B}]\!] = f}{[\![\Gamma \vdash \lambda x : \mathbb{A}.\, V : \mathbb{A} \to \mathbb{B}]\!] = \lambda f}$$

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

Exercises

Show that the following two equations hold.

$$\begin{aligned} [\![x : \mathbb{A}, y : \mathbb{B} \vdash \pi_1 \langle x, y \rangle : \mathbb{A}]\!] &= [\![x : \mathbb{A}, y : \mathbb{B} \vdash x : \mathbb{A}]\!] \\ [\![\Gamma \vdash V : \mathbb{A}]\!] &= [\![\Gamma \vdash \langle \pi_1 V, \pi_2 V \rangle : \mathbb{A}]\!] \end{aligned}$$

Extra points for anyone that shows that the following equation holds :-)

$$[\![(\lambda f.\ \lambda x.\ f(f\ x))\ (\lambda y.\ \langle \pi_2\ x, \pi_1\ x \rangle)]\!] = [\![\lambda x.\ x]\!]$$