A brief overview of Simply-Typed λ-Calculus

Renato Neves

Universidade do Minho

HASLab
HIGH-ASSURANCE SOFTWARE LABORATORY

Table of Contents

The Four Chapters of this course

1. CCS and its semantics: focus on distributed systems comprised of processing units that communicate with each other

2. Adaption of previous notions to real-time systems: semantics via timed labelled transition systems

3. Going cyber-physical: simple imperative while-language and (as usual) program analysis via its semantics

4. Programming with algebraic effects: a uniform approach to the previous chapters

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 via assumptions and logical rules to obtain new knowledge

(e.g. 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 artificial computers

So what does it have to do with programming?

A Logical Rule-Based System for Deductive Reasoning pt. I

Let $\mathbb{A}, \mathbb{B}, \mathbb{C} \ldots$ denote propositions and 1 denote a special proposition that is always true. Next, 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 says that $\mathbb{A}$ implies $\mathbb{B}$

A Logical Rule-Based System for Deductive Reasoning pt. II

Let Γ denote a list of propositions. $\Gamma \vdash \mathbb{A}$ means "if the propositions in Γ hold then we deduce that $\mathbb{A}$ also holds"

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

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

Exercise

Show 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

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

Exercise

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

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

Programs are built according to the previous deduction rules

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

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

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

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

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

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

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

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

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

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.

$$\llbracket x : \mathbb{A}, y : \mathbb{B} \vdash \pi_1 \langle x, y \rangle : \mathbb{A} \rrbracket = \llbracket x : \mathbb{A}, y : \mathbb{B} \vdash x : \mathbb{A} \rrbracket$$
$$\llbracket \Gamma \vdash V : \mathbb{A} \rrbracket = \llbracket \Gamma \vdash \langle \pi_1 V, \pi_2 V \rangle : \mathbb{A} \rrbracket$$

Chocolate for anyone that shows that the following equation holds!

$$\llbracket (\lambda f.\ \lambda x.\ f(f\ x))\,(\lambda x.\ \langle \pi_2\ x, \pi_1\ x \rangle) \rrbracket = \llbracket \lambda x.\ x \rrbracket$$