1 Curry-Howard Correspondence III

1.1 Introduction

How can we formalise the notion of a proof? For example, we can say that $\Gamma \vdash M : \phi$, where Γ is the context, M is a proof and ϕ is some statement. Now we can denote an inference path as follows:

- $\Gamma \vdash M : \phi \to \psi, \Gamma \vdash N : \phi$
- $\Gamma \vdash (MN) : \psi$

We can also write our presuppositions before the proof and the statement, thus identifying explicitly elements of the context.

We can also parametrise our proofs with the language of abstractions:

- $\Gamma, x : \phi \vdash M : \psi$
- $\Gamma \vdash (\lambda x : \phi.M) : \phi \to \psi$

1.2 Types

Let $\Phi(\to)$ denote the implicative fragment of intuitionistic λ-calculus. Call them *simple types*.

Let $\Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\}$, where x_i are variables and τ_i are simple types, be our context. Let $\text{range}(\Gamma)$ be the set of all types.

Now, let M be a λ-term. If M is of type τ, we write $\Gamma \vdash M : \tau$.

If we know from the context that x has a type τ, we write $\Gamma, x : \tau \vdash x : \tau$. Denote this rule as *Var*.

We also introduce the *Abs* rule:

- $\Gamma, x : \tau \vdash M : \sigma$
- $\Gamma \vdash (\lambda x.M) : \tau \to \sigma$

Similarly, we give a rule for application:

- $\Gamma \vdash M : \tau \to \sigma, \Gamma \vdash N : \tau$
- $\Gamma \vdash (MN) : \sigma$

Example 1.1

Given the context $t : \tau_1, s : \tau_2$, we write $t : \tau_1, s : \tau_2 \vdash (\lambda x.t)s : \tau_1$.

Example 1.2

We can also write the following deduction:

- $t : \tau_1, s : \tau_2 \vdash t : \tau_1$
- $t : \tau_1, s : \tau_2 \vdash (\lambda x.t) : \tau_2 \to \tau_1$, $t : \tau_1, s : \tau_2 \vdash s : \tau_2$
- $t : \tau_1, s : \tau_2 \vdash (\lambda x.t)s : \tau_1$

Simple typed λ-calculus is denoted as $\lambda_{\to}$. In our case, we have constructed $\lambda_{\to}$ in the interpretation of Curry.

A similar construction $\lambda_{\to}$ was developed by Church.

Now, consider a spreading combinator $S = \lambda x.\lambda y.\lambda z.xz(yz)$. Having made it typed, we can write it as

$$\lambda x : p \to q \to r.\lambda y : p \to q.\lambda z : p.xz(yz) : (p \to q \to r) \to (p \to q) \to p \to r.$$

If some λ-term M is typed and β-convertible, then the type is conserved.

1.3 Curry-Howard Correspondence

Theorem 1.3

- If $\Gamma \vdash M : \phi : \phi$ in $\lambda_{\to}$, then $\Gamma \vdash \phi$ in the implicative fragment of the intuitionistic λ-calculus.
- If $\Delta \vdash \phi$, then there exists a term and a context Γ such that $\text{range}(\Gamma) = \Delta : F \vdash M : \phi$.

1.4 Extensions

For a pair of λ-terms, we can define another λ-term with a combined type:

$$\langle M, N \rangle : \sigma \times \tau$$

Thus, $\pi_1(\langle M, N \rangle) \to_\beta M$ and $\pi_2(\langle M, N \rangle) \to_\beta N$. We can write:

- $\Gamma \vdash M : \sigma, \Gamma \vdash N : \tau$
- $\Gamma \vdash \langle M, N \rangle : \sigma \times \tau$

Moreover,

- $\Gamma \vdash M : \sigma \times \tau$
- $\Gamma \vdash \pi_1(M) : \sigma.$

How can we assign a type to a disjunction?

We can define constructors:

$$\iota_1^{\sigma \vee \tau}(M) : \sigma \vee \tau \tag{1}$$
$$\iota_2^{\sigma \vee \tau}(N) : \sigma \vee \tau \tag{2}$$

To decide what to do with a λ-term, we can define a decision rule in the form

$$\text{case } M \text{ of } [x]P \text{ or } [y]Q$$

Thus, for example, for the disjunctive λ-term we can write

$$\text{case } \iota_1^{\sigma \vee \tau}(M) \text{ of } [x]P \text{ or } [y]Q \to_\beta [M/x]P$$

For further information, see *Lectures on Curry-Howard Isomorphism.*