Last time

- Dependent types $(x : \tau) \to \tau'$
- Some Agda
- Encoding the list ADT with only $\forall$ and $\to$

$$\alpha \textit{ list} = \forall\beta(\beta \to (\alpha \to \beta \to \beta) \to \beta)$$

This time

- Connection between logic and types

Curry-Howard correspondence

Logic	$\leftrightarrow$	Type system
propositions, ϕ	$\leftrightarrow$	types, τ
(constructive) proofs, p	$\leftrightarrow$	expressions, M
'p is a proof of ϕ'	$\leftrightarrow$	'M is an expression of type τ'
simplification of proofs	$\leftrightarrow$	reduction of expressions

2IPC (Girard) **vs.** **PLC (Reynolds)**

second-order intuitionistic propositional calculus

Intuitionistic (constructive) logic

- About *provability* rather than truth (in classical logic) – propositions are *inhabited* by proofs (justification)
- Weaker than classical logic (no LEM *or equivalently* no double-negation elimination *or equivalently* no Peirce's law)
- but extremely useful

Example of a non-constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Proof. Either $\sqrt{2}^{\sqrt{2}}$ is rational, or it is not (LEM!).

If it is, we can take $a = b = \sqrt{2}$, since $\sqrt{2}$ is irrational by a well-known theorem attributed to Euclid.

If it is not, we can take $a = \sqrt{2}$ and $b = \sqrt{2}^{\sqrt{2}}$, since then $b^a = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\times\sqrt{2}} = \sqrt{2}^2 = 2$.

QED

Intuitionistic (constructive) logic

- About *provability* rather than truth (in classical logic) – propositions are *inhabited* by proofs (justification)
- Weaker than classical logic (no LEM *or equivalently* no double-negation elimination *or equivalently* no Peirce's law)
- but extremely useful

If we did have LEM:

$$\forall p.terminates(p) \vee \neg terminates(p)$$

Propositions inhabited by proofs $\Rightarrow$ LEM solves halting problem!
Not allowed in a constructive logic

Second-order intuitionistic propositional calculus (2IPC)

2IPC propositions: $\boxed{\phi ::= p \mid \phi \to \phi \mid \forall\, p\,(\phi)}$, where p ranges over an infinite set of propositional variables.

2IPC sequents: $\boxed{\Phi \vdash \phi}$, where Φ is a finite (multi)set of 2IPC propositions and ϕ is a 2IPC proposition.

$\Phi \vdash \phi$ is provable if it is in the set of sequents inductively generated by:

$$(\mathrm{Id})\ \Phi \vdash \phi \quad \text{if } \phi \in \Phi$$

$$(\to\mathrm{I})\ \frac{\Phi, \phi \vdash \phi'}{\Phi \vdash \phi \to \phi'} \qquad (\to\mathrm{E})\ \frac{\Phi \vdash \phi \to \phi' \qquad \Phi \vdash \phi}{\Phi \vdash \phi'}$$

$$(\forall\mathrm{I})\ \frac{\Phi \vdash \phi}{\Phi \vdash \forall\, p\,(\phi)} \text{ if } p \notin fv(\Phi) \qquad (\forall\mathrm{E})\ \frac{\Phi \vdash \forall\, p\,(\phi)}{\Phi \vdash \phi[\phi'/p]}$$

In 2IPC

2IPC
proof tree

$$\dfrac{\Phi, \quad \varphi \ \vdash \ \varphi'}{\Phi \vdash \quad \varphi \to \varphi'} \ (\to I)$$

Label hypotheses with variables & recursively build up a "proof term" describing the 2IPC proof

PLC

proof tree

$$\frac{\overline{x} : \overline{\Phi},\ x : \varphi \ \vdash\ M : \varphi'}{\overline{x} : \overline{\Phi} \ \vdash \qquad\qquad \varphi \to \varphi'}\ (\to I)$$

Label hypotheses with variables & recursively build up a "proof term" describing the 2IPC proof

PLC

proof tree

$$\frac{\overline{x} : \Phi ,\ x : \varphi \ \vdash\ M : \varphi'}{\overline{x} : \Phi \ \vdash\ \lambda x{:}\varphi\,(M) : \varphi \to \varphi'}\ (\mathrm{fn})$$

$$\text{(Id) } \Phi, \phi \vdash \phi \qquad \mapsto \qquad \text{(id) } \overline{x} : \Phi, x : \phi \vdash x : \phi$$

$$\text{($\rightarrow$I) } \frac{\Phi, \phi \vdash \phi'}{\Phi \vdash \phi \rightarrow \phi'} \qquad \mapsto \qquad \text{(fn) } \frac{\overline{x} : \Phi, x : \phi \vdash M : \phi'}{\overline{x} : \Phi \vdash \lambda\, x : \phi\, (M) : \phi \rightarrow \phi'}$$

$$\text{($\rightarrow$E) } \frac{\Phi \vdash \phi \rightarrow \phi' \quad \Phi \vdash \phi}{\Phi \vdash \phi'} \qquad \mapsto \qquad \text{(app) } \frac{\overline{x} : \Phi \vdash M_1 : \phi \rightarrow \phi' \quad \overline{x} : \Phi \vdash M_2 : \phi}{\overline{x} : \Phi \vdash M_1\, M_2 : \phi'}$$

$$\text{($\forall$I) } \frac{\Phi \vdash \phi}{\Phi \vdash \forall\, p\, (\phi)} \qquad \mapsto \qquad \text{(gen) } \frac{\overline{x} : \Phi \vdash M : \phi}{\overline{x} : \Phi \vdash \Lambda\, p\, (M) : \forall\, p\, (\phi)}$$

$$\text{($\forall$E) } \frac{\Phi \vdash \forall\, p\, (\phi)}{\Phi \vdash \phi[\phi'/p]} \qquad \mapsto \qquad \text{(spec) } \frac{\overline{x} : \Phi \vdash M : \forall\, p\, (\phi)}{\overline{x} : \Phi \vdash M\, \phi' : \phi[\phi'/p]}$$

A 2IPC proof

$$\forall p, q\,((p \mathbin{\&} q) \to p).$$

$$
\cfrac{
(\to E)\;
\cfrac{
(\to I)\;\cfrac{(\to I)\;\cfrac{(Id)\;\cfrac{}{\{p \mathbin{\&} q, p, q\} \vdash p}}{\{p \mathbin{\&} q, p\} \vdash q \to p}}{\{p \mathbin{\&} q\} \vdash p \to q \to p}
\qquad
(\forall E)\;\cfrac{(Id)\;\cfrac{}{\{p \mathbin{\&} q\} \vdash \forall r\,((p \to q \to r) \to r)}}{\{p \mathbin{\&} q\} \vdash (p \to q \to p) \to p}
}{\{p \mathbin{\&} q\} \vdash p}
}{}
$$

$$
(\forall I)\;\cfrac{(\forall I)\;\cfrac{(\to I)\;\cfrac{\{p \mathbin{\&} q\} \vdash p}{\{\,\} \vdash p \mathbin{\&} q \to p}}{\{\,\} \vdash \forall q\,(p \mathbin{\&} q \to p)}}{\{\,\} \vdash \forall p, q\,(p \mathbin{\&} q \to p)}
$$

where $p \mathbin{\&} q$ is an abbreviation for $\forall r\,((p \to q \to r) \to r)$.

The PLC expression corresponding to this proof is:

$$\Lambda\, p, q\,(\lambda\, z : p \mathbin{\&} q\,(z\, p\,(\lambda\, x : p, y : q\,(x)))).$$

$$p\&q = \forall r.\,(p\to q\to r)\to r$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z:p\&q,\, x:p,\, y:q \vdash x : p}\,\text{id}}{z : p\&q \vdash \lambda x{:}p\,\lambda y{:}q\,(x) : p \to (q\to p)}\,\text{fn}^2 \qquad \dfrac{\dfrac{}{z : p\&q \vdash z : \forall r.(p\to q\to r)\to r}\,\text{id}}{z:p\&q \vdash z\,p : (p\to q\to p)\to p}\,\forall E}{z : p\&q \vdash z\,p\,(\lambda x\,\lambda y\;\, x) : p}\,\text{app}}{\emptyset \vdash \lambda z{:}p\&q\,(z\,p\,(\lambda x\,\lambda y(x))) : p\&q \to p}\,\text{abs (fn)}}{\emptyset \vdash \Lambda p,q\,\lambda z{:}p\&q\,(z\,p\,(\lambda x{:}p,\,\lambda y{:}q\,(x)) : \forall p,q.\, p\&q \to p}\,\forall I^2$$

Exercise (4 mins)

In 2IPC, prove:

$$\forall p, q, r, s(((p \to q \to r) \to r) \to s) \to (p \to q \to s)$$

hint: Is there a PLC function with type:

$$\forall p, q, r, s(((p \to q \to r) \to r) \to s) \to p \to q \to s$$

(i.e., function with three parameters and result type s)

Since $p \wedge q = \forall r.((p \to q \to r) \to r)$ then

$$\begin{aligned} & \forall p, q, r, s\; (((p \to q \to r) \to r) \to s) \to (p \to q \to s) \\ \cong\; & \forall p, q, s\; ((p \wedge q) \to s) \to (p \to q \to s) \end{aligned}$$

The proof of which is witnessed by the *curry* function, via the Curry-Howard correspondence.

Curry-Howard proof in Agda

```
exercise : forall {p q r s : Set} ->
          (((p -> q -> r) -> r) -> s) -> p -> q -> s
exercise k p q = k (\f -> f p q)
```

$$\frac{\Phi \vdash A \quad \Phi \vdash B}{\Phi \vdash A \wedge B}\ \wedge i \qquad\qquad *i\ \frac{\Gamma \vdash e : \tau \quad \Gamma \vdash e' : \tau'}{\Gamma \vdash (e, e') : \tau * \tau'}$$

$$\frac{\Phi \vdash A \wedge B}{\Phi \vdash A}\ \wedge e_1 \qquad\qquad *e_1\ \frac{\Gamma \vdash e : \tau * \tau'}{\Gamma \vdash \mathit{fst}\ e : \tau}$$

$$\frac{\Phi \vdash A \wedge B}{\Phi \vdash B}\ \wedge e_2 \qquad\qquad *e_2\ \frac{\Gamma \vdash e : \tau * \tau'}{\Gamma \vdash \mathit{snd}\ e' : \tau'}$$

$$\wedge \xleftrightarrow{\text{corresponds to}} *$$

Data Constructor $\quad (-,-) : \tau \to \tau' \to (\tau * \tau')$

PLC encoding:

$$\forall \tau \forall \tau' .\ \forall p.\ ((\tau \to \tau' \to p) \to p) \quad \underset{\alpha\text{-equiv}}{\cong}$$

$$\forall p \forall q .\ \forall r.\ ((p \to q \to r) \to r)$$

$$\cong (p * q)$$

Conjunction $\leftrightarrow$ Pairs correspondence

Logical operations definable in 2IPC

- Truth: $\mathit{true} \stackrel{\text{def}}{=} \forall p\,(p \to p)$.
- Falsity: $\mathit{false} \stackrel{\text{def}}{=} \forall p\,(p)$.
- Conjunction: $\phi \mathbin{\&} \phi' \stackrel{\text{def}}{=} \forall p\,((\phi \to \phi' \to p) \to p)$ (where $p \notin \mathit{fv}(\phi, \phi')$).
- Disjunction: $\phi \vee \phi' \stackrel{\text{def}}{=} \forall p\,((\phi \to p) \to (\phi' \to p) \to p)$ (where $p \notin \mathit{fv}(\phi, \phi')$).
- Negation: $\neg\phi \stackrel{\text{def}}{=} \phi \to \mathit{false}$.
- Existential quantification: $\exists\, p\,(\phi) \stackrel{\text{def}}{=} \forall p'\,(\forall p\,(\phi \to p') \to p')$ (where $p' \notin \mathit{fv}(\phi, p)$).

2IPC is a constructive logic

For example, there is no proof of the Law of Excluded Middle

$$\forall\, p\, (p \vee \neg p)$$

Using the definitions on Slide 67, this is an abbreviation for

$$\forall\, p, q\, ((p \to q) \to ((p \to \forall\, r\, (r)) \to q) \to q)$$

(The fact that there is no closed PLC term of type $\forall\, p\, (p \vee \neg p)$ can be proved using the technique developed in the Tripos question 13 on paper 9 in 2000.)

Curry-Howard correspondence

Logic	$\leftrightarrow$	Type system
propositions, ϕ	$\leftrightarrow$	types, τ
(constructive) proofs, p	$\leftrightarrow$	expressions, M
'p is a proof of ϕ'	$\leftrightarrow$	'M is an expression of type τ'
simplification of proofs	$\leftrightarrow$	reduction of expressions

Proof simplification $\leftrightarrow$ term reduction

$$
{\to}E\ \dfrac{{\to}I\ \dfrac{\dfrac{\vdots}{\Phi, \psi \vdash \psi'}}{\Phi \vdash \psi \to \psi'} \qquad \dfrac{\vdots}{\Phi \vdash \psi}}{\Phi \vdash \psi'}
\qquad\Bigg|\qquad
\dfrac{\dfrac{\dfrac{\vdots}{\Gamma : \Phi, x : \psi \vdash {\color{red}M} : \psi'}}{\Gamma : \Phi \vdash {\color{red}\lambda x : \psi(M)} : \psi \to \psi'} \qquad \dfrac{\vdots}{\Gamma : \Phi \vdash {\color{red}N} : \psi}}{\Gamma : \Phi \vdash {\color{red}(\lambda x : \psi(M))N} : \psi'}
$$

$$
\Bigg\downarrow \text{(simplify)} \qquad\qquad\qquad \Bigg\downarrow (\beta\text{-reduce})
$$

$$
\text{cut}\ \dfrac{\dfrac{\vdots}{\Phi, \psi \vdash \psi'} \qquad \dfrac{\vdots}{\Phi \vdash \psi}}{\Phi \vdash \psi'}
\qquad\Bigg|\qquad
\text{cut}\ \dfrac{\dfrac{\vdots}{\Gamma : \Phi, x : \psi \vdash {\color{red}M} : \psi'} \qquad \dfrac{\vdots}{\Gamma : \Phi \vdash {\color{red}N} : \psi}}{\Gamma : \Phi \vdash {\color{red}M[N/x]} : \psi'}
$$

Type-inference versus proof search

Type-inference: 'given Γ and M, is there a type τ such that $\Gamma \vdash M : \tau$?'
(For PLC/2IPC this is decidable.)

Proof-search: 'given Γ and ϕ, is there a proof term M such that $\Gamma \vdash M : \phi$?'

(For PLC/2IPC this is undecidable.)

Course outline

- **Introduction**. The role of type systems in programming languages. Formalizing type systems. [1 lecture]
- **ML polymorphism**. ML-style polymorphism. Principal type schemes and type inference. [2 lectures]
- **Polymorphic reference types**. The pitfalls of combining ML polymorphism with reference types. [1 lecture]
- **Polymorphic lambda calculus**. Syntax and reduction semantics. Examples of datatypes definable in the polymorphic lambda calculus. Applications. [2 lectures]
- **Further topics**. The Curry-Howard correspondence (types-as-formulae, terms-as-proofs) as a source of type systems. Dependent types. [2 lectures]