1 Intuitionistic Logic and Constructive Mathematics

We have previously observed that several familiar type judgments $\vdash e : \tau$ of the pure simply-typed λ-calculus correspond to a tautologies of propositional logic:

type judgment	*propositional tautology*
$\vdash I : \alpha \to \alpha$	$P \to P$
$\vdash K : \alpha \to \beta \to \alpha$	$P \to (Q \to P)$
$\vdash S : (\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to (\alpha \to \gamma)$	$(P \to Q \to R) \to (P \to Q) \to (P \to R)$

This is no accident. It turns out that all derivable type judgments $\vdash e : \tau$ (with the empty environment to the left of the turnstile) give propositional tautologies. This is because the typing rules of $\lambda^{\to}$ correspond exactly to the proof rules of propositional *intuitionistic logic*.

Intuitionistic logic is the basis of *constructive mathematics*. Constructive mathematics takes a much more conservative view of truth than classical mathematics. It is concerned less with *truth* than with *provability*. Two of its main proponents were Kronecker and Brouwer. These views generated great controversy in the mathematical world when they first appeared.

In constructive mathematics, not all deductions of classical logic are considered valid. For example, to prove in classical logic that there exists an object having a certain property, it is enough to assume that no such object exists and derive a contradiction. Intuitionists would not consider this argument valid. Intuitionistically, you must actually construct the object and prove that it has the desired property.

Intuitionists do not accept the law of double negation: $P \leftrightarrow \neg\neg P$. They do believe that $P \to \neg\neg P$, that is, if P is true then it is not false; but they do not believe $\neg\neg P \to P$, that is, even if P is not false, then that does not automatically make it true.

Similarly, intuitionists do not accept the law of the excluded middle $P \vee \neg P$. In order to prove $P \vee \neg P$, you must prove either P or $\neg P$. It may well be that neither is provable, in which case the intuitionist would not accept that $P \vee \neg P$.

For intuitionists, the implication $P \to Q$ has a much stronger meaning than merely $\neg P \vee Q$, as in classical logic. To prove $P \to Q$, one must show how to construct a proof of Q from any given proof of P. So a proof of $P \to Q$ is a (computable) function from proofs of P to proofs of Q. Similarly, to prove $P \wedge Q$, you must prove both P and Q; thus a proof of $P \wedge Q$ is a pair consisting of a proof of P and a proof of Q.

1.1 Example

Here is an example of a proof that would not be accepted by an intuitionist.

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

Proof. Either $\sqrt{2}^{\sqrt{2}}$ is rational or not. If it is, take $a = b = \sqrt{2}$ and we are done. If it is not, take $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$; then $a^b = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2$, and again we are done. $\square$

Now an intuitionist would not like this, because we have not constructed a definite a and b with the desired property. We have used the law of the excluded middle, which the intuitionist would regard as cheating.[1]

2 Syntax

Syntactically, formulas $\varphi, \psi, \ldots$ of intuitionistic logic look the same as their classical counterparts. At the propositional level, we have propositional variables $P, Q, R, \ldots$ and formulas

$$\varphi \quad ::= \quad \top \ \mid \ \bot \ \mid \ P \ \mid \ \varphi \to \psi \ \mid \ \varphi \vee \psi \ \mid \ \varphi \wedge \psi \ \mid \ \neg\varphi.$$

We might also add a second-order quantifier $\forall P$ ranging over propositions:

$$\varphi \quad ::= \quad \cdots \ \mid \ \forall P.\,\varphi.$$

3 Natural Deduction (Gentzen, 1943)

Intuitionistic logic uses a sequent calculus to derive the truth of formulas. Assertions are judgments of the form $\varphi_1, \ldots, \varphi_n \vdash \varphi$, which means that φ can be derived from the assumptions $\varphi_1, \ldots, \varphi_n$. If $\vdash \varphi$ without assumptions, then φ is a theorem of intuitionistic logic. The system is called *natural deduction*.

As we write down the proof rules, it will be clear that they correspond exactly to the typing rules of the pure simply-typed λ-calculus $\lambda^{\to}$ (and with quantifiers, System F). We will show them side by side. There are generally *introduction* and *elimination* rules for each operator.

	intuitionistic logic	*$\lambda^{\to}$ or System F type system*
(axiom)	$\Gamma, \varphi \vdash \varphi$	$\Gamma, x:\tau \vdash x:\tau$
($\to$-intro)	$\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$	$\dfrac{\Gamma, x:\sigma \vdash e:\tau}{\Gamma \vdash (\lambda x:\sigma.\,e):\sigma \to \tau}$
($\to$-elim)	$\dfrac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$	$\dfrac{\Gamma \vdash e_0:\sigma \to \tau \quad \Gamma \vdash e_1:\sigma}{\Gamma \vdash (e_0\ e_1):\tau}$
($\wedge$-intro)	$\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}$	$\dfrac{\Gamma \vdash e_1:\sigma \quad \Gamma \vdash e_2:\tau}{\Gamma \vdash (e_1, e_2):\sigma * \tau}$
($\wedge$-elim)	$\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \quad \dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}$	$\dfrac{\Gamma \vdash e:\sigma * \tau}{\Gamma \vdash \#1\,e:\sigma} \quad \dfrac{\Gamma \vdash e:\sigma * \tau}{\Gamma \vdash \#2\,e:\tau}$
($\vee$-intro)	$\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \quad \dfrac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}$	$\dfrac{\Gamma \vdash e:\sigma}{\Gamma \vdash \mathsf{inl}_{\sigma+\tau}:e\sigma + \tau} \quad \dfrac{\Gamma \vdash e:\tau}{\Gamma \vdash \mathsf{inr}_{\sigma+\tau}:e\sigma + \tau}$
($\vee$-elim)	$\dfrac{\Gamma \vdash \varphi \vee \psi \quad \Gamma \vdash \varphi \to \chi \quad \Gamma \vdash \psi \to \chi}{\Gamma \vdash \chi}$	$\dfrac{\Gamma \vdash e:\sigma + \tau \quad \Gamma \vdash e_1:\sigma \to \rho \quad \Gamma \vdash e_2:\tau \to \rho}{\Gamma \vdash \mathsf{case}\ e_0\ \mathsf{of}\ e_1 \mid e_2:\rho}$
($\forall$-intro)	$\dfrac{\Gamma, P \vdash \varphi}{\Gamma \vdash \forall P.\,\varphi}$	$\dfrac{\Delta, \alpha; \Gamma \vdash e:\tau \quad \alpha \notin FV(\Gamma)}{\Delta; \Gamma \vdash (\Lambda\alpha.\,e):\forall\alpha.\,\tau}$
($\forall$-elim)	$\dfrac{\Gamma \vdash \forall P.\,\varphi}{\Gamma \vdash \varphi\{\psi/P\}}$	$\dfrac{\Delta; \Gamma \vdash e:\forall\alpha.\,\tau \quad \Delta \vdash \sigma}{\Delta; \Gamma \vdash (e\ \sigma):\tau\{\sigma/\alpha\}}$

The $\to$-elimination rule is often called *modus ponens.*

[1]It is possible to give a constructive proof using only elementary facts. Hint: consider $\sqrt{2}^{\log_2 9}$.

4 The Brouwer–Heyting–Kolmogorov (BHK) Principle

The fact that propositions in intuitionistic logic correspond to types in our λ-calculus type systems is known as the *Brouwer–Heyting–Kolmogorov* (BKH) interpretation or *propositions as types* principle, and sometimes the *Curry–Howard isomorphism* (although neither is it due to Curry and Howard nor is it an isomorphism). The analogy is far reaching:

type theory		*logic*	
τ	type	φ	proposition
τ	inhabited type	φ	theorem
e	well-typed program	π	proof
$\rightarrow$	function space	$\rightarrow$	implication
$*$	product	$\wedge$	conjunction
$+$	sum	$\vee$	disjunction
$\forall$	type quantifier	$\forall$	second-order quantifier
1	unit	$\top$	truth
0	void	$\bot$	falsity

A proof in intuitionistic logic is a construction, which is essentially a program (λ-term). Saying that a proposition has an intuitionistic or constructive proof says essentially that the corresponding type is inhabited by a λ-term.

If we are given a well-typed term in System F or $\lambda^{\rightarrow}$, then its proof tree will look exactly like the proof tree for the corresponding formula in intuitionistic logic. This means that every well-typed program proves something, that is, it is a proof in constructive logic. Conversely, every theorem in constructive logic corresponds to an inhabited type. Several automated deduction systems (for example, Nuprl and Coq) are based on this idea.

5 Theorem Proving and Type Checking

We have seen that *type inference* is the process of inferring a type for a given λ-term. Under the propositions-as-types principle, this is the same as determining what theorem a given proof proves. Theorem proving, on the other hand, is going in the opposite direction: Given a formula, does it have a proof? Equivalently, given a type, is it inhabited?

For example, consider the formula expressing transitivity of implication:

$$\forall P, Q, R\,.\; ((P \rightarrow Q) \wedge (Q \rightarrow R)) \;\rightarrow\; (P \rightarrow R)$$

Under the propositions-as-types principle, this is related to the type

$$\forall \alpha, \beta, \gamma\,.\; (\alpha \rightarrow \beta) * (\beta \rightarrow \gamma) \;\rightarrow\; (\alpha \rightarrow \gamma).$$

If we can construct a term of this type, we will have proved the theorem in intuitionistic logic. The program

$$\Lambda \alpha, \beta, \gamma.\, \lambda p : (\alpha \rightarrow \beta) * (\beta \rightarrow \gamma).\, \lambda x : \alpha.\, (\#2\, p)\; ((\#1\, p)\; x)$$

does it. This is a function that takes a pair of functions as its argument and returns their composition. The proof tree that establishes the typing of this function is essentially an intuitionistic proof of the transitivity of implication.

Here is another example. Consider the formula

$$\forall P, Q, R\,.\; (P \wedge Q \rightarrow R) \;\leftrightarrow\; (P \rightarrow Q \rightarrow R)$$

The double implication $\leftrightarrow$ is an abbreviation for the conjunction of the implications in both directions. It says that the two formulas on either side are propositionally equivalent. The typed expressions corresponding to each side of the formula above are

$$\alpha * \beta \to \gamma \qquad\qquad \alpha \to \beta \to \gamma.$$

We know that any term of the first type can be converted to one of the second by *currying*, and we can go in the opposite direction by *uncurrying*. The two λ-terms that convert a function to its curried form and back constitute a proof of the logical statement.