CS 4110

Programming Languages & Logics

Lecture 29
Propositions as Types

10 November 2014

Propositions as Types

There is a deep connection between type systems and formal logic

This connection was known to early 20th century mathematicians but was later developed substantially by Curry and Howard

Although it is usually formulated in terms of type systems like System F and proof systems like natural deduction, the connection is actually very robust and has been extended to many other systems including classical logic

Propositions as Types

The main intuitions for propositions-as-types comes from thinking of proofs constructively

For example, the proof rule for introducting a conjunction $\phi \wedge \psi$,

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge\text{-intro}$$

can be thought of as a function that takes a proof of ϕ and a proof of ψ and builds a proof of $\phi \wedge \psi$

Propositions as Types

The main intuitions for propositions-as-types comes from thinking of proofs constructively

For example, the proof rule for introducting a conjunction $\phi \wedge \psi$,

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge\text{-intro}$$

can be thought of as a function that takes a proof of ϕ and a proof of ψ and builds a proof of $\phi \wedge \psi$

This is a significant departure from classical logic, which has rules such as excluded middle or double-negation elimination,

$$\frac{}{\Gamma \vdash \psi \wedge \neg\psi}\ \text{excluded middle}$$

that do not have an obvious constructive interpretation

Propositions as Types

Propositions-as-types recongizes that each constructive proof can be turned into a program that witnesses the proof, as summarized by the following table:

Type Systems		Formal Logic	
τ	Type	ϕ	Formula
τ	Inhabited type	ϕ	Theorem
e	Well-typed expression	π	Proof

Hence, for every proof in first-order logic, we can obtain a well-typed expression in λ-calculus, and vice versa

Formulas

Let *P* range over first-order propositional variables

$$\begin{array}{rcl}
\phi & ::= & \top \\
 & | & \bot \\
 & | & P \\
 & | & \phi \land \psi \\
 & | & \phi \lor \psi \\
 & | & \phi \rightarrow \psi \\
 & | & \neg\phi \\
 & | & \forall x.\ \phi
\end{array}$$

Formulas

Let P range over first-order propositional variables

$$\begin{array}{rcl} \phi & ::= & \top \\ & | & \bot \\ & | & P \\ & | & \phi \wedge \psi \\ & | & \phi \vee \psi \\ & | & \phi \rightarrow \psi \\ & | & \neg\phi \\ & | & \forall x.\ \phi \end{array}$$

We will let negation $\neg P$ be an abbreviation for $P \rightarrow \bot$

Natural Deduction

$$\frac{}{\Gamma, \phi \vdash \mathit{phi}}\ \text{Axiom}$$

$$\frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \to \psi}\ \to\text{-intro} \qquad \frac{\Gamma \vdash \phi \to \psi \qquad \Gamma \vdash \phi}{\Gamma \vdash \psi}\ \to\text{-elim}$$

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge\text{-intro} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi}\ \wedge\text{-elim1} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi}\ \wedge\text{-elim2}$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi}\ \vee\text{-intro1} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi}\ \vee\text{-intro2}$$

$$\frac{\Gamma \vdash \phi \vee \psi \qquad \Gamma \vdash \phi \to \chi \qquad \Gamma \vdash \psi \to \chi}{\Gamma \vdash \chi}\ \vee\text{-elim}$$

$$\frac{\Gamma, P \vdash \phi}{\Gamma \vdash \forall P.\, \phi}\ \forall\text{-intro} \qquad \frac{\Gamma \vdash \forall P.\, \phi}{\Gamma \vdash \phi\{\psi/P\}}\ \forall\text{-elim}$$

System F Type System

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ \text{Axiom}$$

$$\frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \lambda x{:}\sigma.\, e{:}\tau}\ \to\text{-intro} \qquad \frac{\Gamma \vdash e_1 : \sigma \to \tau \qquad \Gamma \vdash e_2 : \sigma}{\Gamma \vdash e_1\ e_2 : \tau}\ \to\text{-elim}$$

$$\frac{\Gamma \vdash e_1 {:} \sigma \qquad \Gamma \vdash e_2 {:} \tau}{\Gamma \vdash (e_1, e_2) {:} \sigma \times \tau}\ \wedge\text{-intro} \qquad \frac{\Gamma \vdash e {:} \sigma \times \tau}{\Gamma \vdash \#1\, e : \sigma}\ \wedge\text{-elim1} \qquad \frac{\Gamma \vdash e {:} \sigma \times \tau}{\Gamma \vdash \#2\, e : \tau}\ \wedge\text{-elim2}$$

$$\frac{\Gamma \vdash e : \sigma}{\Gamma \vdash inl_{\sigma+\tau}\ e {:} \sigma + \tau}\ \vee\text{-intro1} \qquad \frac{\Gamma \vdash e : \tau}{\Gamma \vdash inr_{\sigma+\tau}\ e {:} \sigma + \tau}\ \vee\text{-intro2}$$

$$\frac{\Gamma \vdash e\sigma + \tau \qquad \Gamma \vdash e_1 {:} \sigma \to \rho \qquad \Gamma \vdash e_2 {:} \tau \to \rho}{\Gamma \vdash case\ e\ of\ e_1\ e_2 {:} \rho}\ \vee\text{-elim}$$

$$\frac{\Delta, \alpha; \Gamma \vdash e : \tau}{\Delta; \Gamma \vdash \Lambda\alpha.\, e : \forall\alpha.\tau}\ \forall\text{-intro} \qquad \frac{\Gamma \vdash e {:} \forall\alpha.\, \tau}{\Gamma \vdash e\,[\sigma] {:} \tau\{\sigma/\alpha\}}\ \forall\text{-elim}$$

Propositions as Types

We can summarize the relationship between formulas and types using the following table:

Type Systems		**Formal Logic**	
$\rightarrow$	Function	$\rightarrow$	Implication
$\times$	Product	$\wedge$	Conjunction
$+$	Sum	$\vee$	Disjunction
$\forall$	Universal	$\forall$	Quantifier

Term Assignment

Given a natural deduction proof, there is a corresponding well-typed System F expression

The transformation from a proof to an expression is often called *term assignment*

Term assignment provides a convenient way to prove a theorem: find a λ-calculus term with the corresponding type!

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \phi \to \psi, \phi \vdash \phi \to \psi}\ \text{Axiom} \qquad \dfrac{}{\Gamma, \phi \to \psi, \phi \vdash \phi}\ \text{Axiom}}{\Gamma, \phi \to \psi, \phi \vdash \psi}\ {\to}\text{-elim}}{\Gamma, \phi \to \psi, \vdash \phi \to \psi}\ {\to}\text{-intro}}{\Gamma \vdash (\phi \to \psi) \to \phi \to \psi}\ {\to}\text{-intro}$$

Example

$$\dfrac{\dfrac{\dfrac{}{\Gamma, x : \sigma \to \tau, y : \sigma \vdash x : \sigma \to \tau}\ \text{Axiom} \qquad \dfrac{}{\Gamma, x : \sigma \to \tau, y : \sigma \vdash \sigma}\ \text{Axiom}}{\Gamma, x : \sigma \to \tau, y : \sigma \vdash x\, y : \tau}\ \to\text{-elim}}{\dfrac{\Gamma, x : \sigma \to \tau \vdash \lambda y : \sigma.\, x\, y : \sigma \to \tau}{\Gamma \vdash \lambda x : \sigma \to \tau.\, \lambda y : \sigma.\, x\, y : (\sigma \to \tau) \to \sigma \to \tau}}\ \to\text{-intro}$$

Example

Hence, the term

$$\lambda x{:}\sigma \to \tau.\, \lambda y{:}\sigma.\, x\, y$$

witnesses the proof of

$$(\phi \to \psi) \to \phi \to \psi$$

Classical Logic

The problem of extending propositions-as-types to classical logic was an open question for many years

It was not at all obvious how to do this, as the usual constructive interpretation of proofs does not readily extend to rules such as excluded middle

In the late 1980s, Griffin showed that continuation-passing style corresponds to a way of embedding classical logic into constructive logic

Negation and Continuations

Given an expression *e* of type τ, we can think of the continuation-passing style transformation as converting the expression into one of type $(\tau \to \bot) \to \bot$

Intuitively, the $\tau \to \bot$ is the type of the continuation, which "never returns"

Since $\neg\phi$ is just an abbreviation for $\phi \to \bot$, this corresponds to the following classical rule:

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \neg\neg\phi}$$

This yields a way to prove any formula that is classically valid in constructive logic using the double-negation embedding