The Curry–Howard Isomorphism

Introduction

- An example

$$\frac{A \to B \quad A}{B}$$

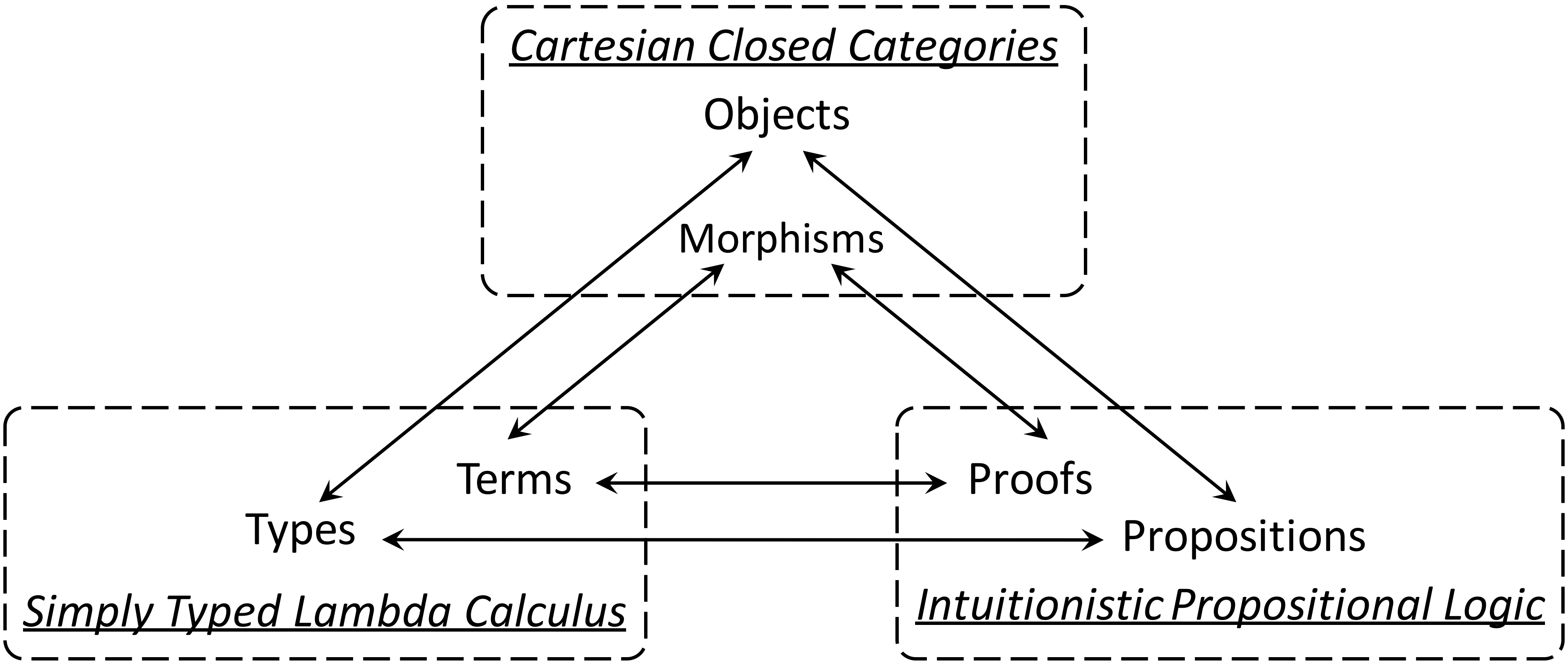
Modus ponens

$$\frac{f : A \to B \quad x : A}{f\,x : B}$$

Function application

- Propositions correspond to types
- Proofs correspond to programs

A Three-way Correspondence



Intuitionistic Logic

- Syntactically, *reductio ad absurdum* (*RAA*) is not a rule in the natural deduction system.
- The law of excluded middle $\varphi \vee \neg\varphi$ and double negation elimination $\neg\neg\varphi \to \varphi$ are no longer axioms.
- Semantically, the judgements about statements are based on the existence of a proof (or construction) of that statement

Simply Typed Lambda Calculus

- A family of prototype programming languages
- Simply typed lambda calculus with function types $\lambda^{\rightarrow}$
  - Types $\sigma \quad ::= \quad \iota \mid \sigma \rightarrow \sigma$
  - Terms $M \quad ::= \quad x \mid \lambda x.M \mid MM$
- Three kinds of equivalence
  - α-equivalence $\lambda x.M =_\alpha \lambda y.[y/x]M$
  - β-equivalence $(\lambda x.M)N =_\beta [N/x]M$
  - η-equivalence $\lambda x.Mx =_\eta M$

Cartesian Closed Categories

A *CCC* is a category with the following extra structure:

- A terminal object *unit* with unique arrow $\mathbf{One}^{\sigma}$ for every σ
- Object map $\times$, function $\langle\cdot,\cdot\rangle$, and arrows $\mathbf{Proj_1}$ and $\mathbf{Proj_2}$ satisfying $\mathbf{Proj}_{\boldsymbol{i}}\circ\langle f_1, f_2\rangle = f_i$ and $\langle \mathbf{Proj_1}\circ g, \mathbf{Proj_2}\circ g\rangle = g$
- Object map $\rightarrow$, function **Curry**, and arrow **App** satisfying $\mathbf{App}\circ\langle \mathbf{Curry}(h)\circ\mathbf{Proj_1}, \mathbf{Proj_2}\rangle = h$ and $\mathbf{Curry}(\mathbf{App}\circ\langle k\circ\mathbf{Proj_1}, \mathbf{Proj_2}\rangle) = k$

Encode Proofs in Typed Lambda Terms

- Every proof leads to a type derivation
- Every type-derivation bring a well-typed lambda term
- Therefore, every proof can be encoded in a well-typed lambda term
  - assumptions are encoded in type context
  - proofs of theorems are encoded in closed terms

Encode Proofs in Typed Lambda Terms

- Example – Theorem $\varphi \to (\psi \to \varphi)$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \vdash \varphi}}{\varphi,\ \psi \vdash \varphi}\,(add)}{\varphi \vdash \psi \to \varphi}\,(\to \mathrm{I})}{\vdash \varphi \to (\psi \to \varphi)}\,(\to \mathrm{I})$$

- The proof of theorem $\varphi \to (\psi \to \varphi)$ is encoded in combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

Encode Proofs in Typed Lambda Terms

- Example – Theorem $\varphi \to (\psi \to \varphi)$

$$\dfrac{\dfrac{\dfrac{\overline{x:\varphi \vdash x:\varphi}}{x:\varphi, y:\psi \vdash x:\varphi}(add)}{x:\varphi \vdash \lambda y:\psi.x : \psi \to \varphi}(\to \mathrm{I})}{\vdash \lambda x:\varphi.\lambda y:\psi.x : \varphi \to (\psi \to \varphi)}(\to \mathrm{I})$$

- The proof of theorem $\varphi \to (\psi \to \varphi)$ is encoded in combinator $\mathbf{K} \equiv \lambda x:\varphi.\lambda y:\psi.x : \varphi \to \psi \to \varphi$

Decode Well-typed Lambda Terms

- Every type derivation leads to a proof (by erasing terms)
- Every well-typed lambda term uniquely determines a type derivation
- By decoding a well-typed lambda term, we obtain a proof
  - Rebuild the type derivation of the term
  - Erase all the terms in the derivation

Decode Well-typed Lambda Terms

- Example – Combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

$$\dfrac{\dfrac{\dfrac{\overline{x{:}\varphi \vdash x{:}\varphi}}{x{:}\varphi, y{:}\psi \vdash x{:}\varphi}(add)}{x{:}\varphi \vdash \lambda y{:}\psi.x : \psi \to \varphi}(\to \mathrm{I})}{\vdash \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to (\psi \to \varphi)}(\to \mathrm{I})$$

- The proof of theorem $\varphi \to (\psi \to \varphi)$ is obtained by decoding the combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

Decode Well-typed Lambda Terms

- Example – Combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cancel{x:}\,\varphi \vdash \cancel{x:}\,\varphi}}{\cancel{x:}\,\varphi, \cancel{y:}\,\psi \vdash \cancel{x:}\,\varphi}\,(add)}{\cancel{x:}\,\varphi \vdash \cancel{\lambda y{:}\psi.x :}\,\psi \to \varphi}\,(\to \mathrm{I})}{\vdash \cancel{\lambda x{:}\varphi.\lambda y{:}\psi.x :}\,\varphi \to (\psi \to \varphi)}\,(\to \mathrm{I})$$

- The proof of theorem $\varphi \to (\psi \to \varphi)$ is obtained by decoding the combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

Decode Well-typed Lambda Terms

- Example – Combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

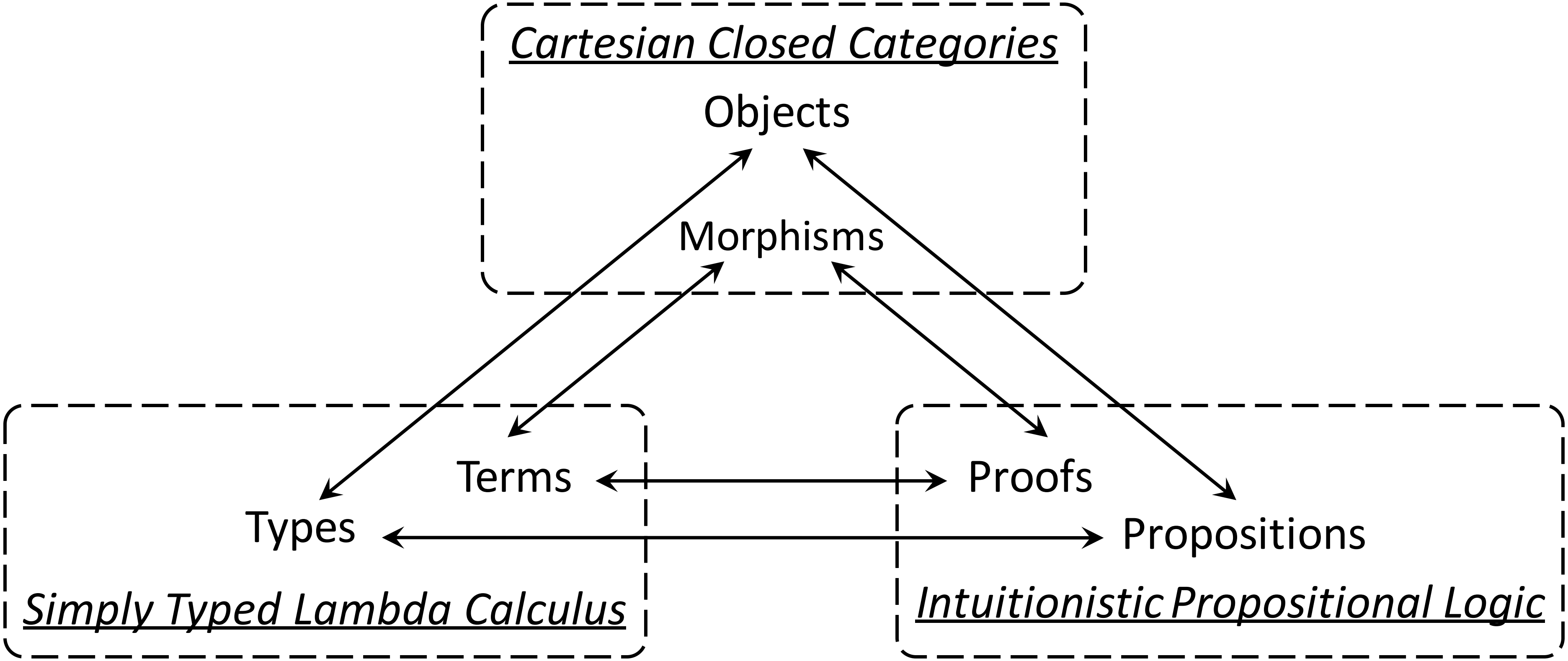
$$\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \vdash \varphi}}{\varphi, \ \psi \vdash \varphi}(add)}{\varphi \vdash \psi \to \varphi}(\to \mathrm{I})}{\vdash \varphi \to (\psi \to \varphi)}(\to \mathrm{I})$$

- The proof of theorem $\varphi \to (\psi \to \varphi)$ is obtained by decoding the combinator $\mathbf{K} \equiv \lambda x{:}\varphi.\lambda y{:}\psi.x : \varphi \to \psi \to \varphi$

Some Correspondence

- Propositions correspond to types
  - set of propositions ≌ set of types
  - propositional connectives ≌ type constructors
- Proofs correspond to terms
  - natural deduction system ≌ typing rules
  - normalization ≌ reduction
- Some typical questions
  - proof checking ≌ type checking
  - provability ≌ inhabitation

A Three-way Correspondence



Interpret Well-typed Terms as Morphisms

$\lambda^{unit,\times,\to}$	CCCs
terminal type $unit$	terminal object $unit$
product types $\sigma \times \tau$	products $A \times B$
function types $\sigma \to \tau$	function types $A \to B$

- $\lambda^{\to}$ is as expressive as $\lambda^{unit,\times,\to}$
  - product $\sigma_1 \times \ldots \times \sigma_n \triangleq$ sequence $\sigma_1, \ldots, \sigma_n$
  - function $f: \sigma \times \tau \to \rho \triangleq \bar{f}: \sigma \to (\tau \to \rho)$
  - function $g: \sigma \to (\tau \times \rho) \triangleq g_1: \sigma \to \tau$ and $g_2: \sigma \to \rho$

Interpret Well-typed Terms as Morphisms

Given a cartesian closed category $\mathcal{C}$ with some object constants

- the interpretation of type expressions
  - $\mathcal{C}[\![\iota]\!] = \hat{b}$
  - $\mathcal{C}[\![\sigma \to \tau]\!] = \mathcal{C}[\![\sigma]\!] \to \mathcal{C}[\![\tau]\!]$
- the interpretation of type expressions
  - $\mathcal{C}[\![\emptyset]\!] = unit$
  - $\mathcal{C}[\![\Gamma, x \colon \sigma]\!] = \mathcal{C}[\![\Gamma]\!] \times \mathcal{C}[\![\sigma]\!]$

Interpret Well-typed Terms as Morphisms

- the interpretation of well-typed terms
  - $\mathcal{C}[\![\Gamma, x{:}\sigma \rhd x{:}\sigma]\!] = \mathrm{Proj}_2^{\mathcal{C}[\![\Gamma]\!],\mathcal{C}[\![\sigma]\!]}$
  - $\mathcal{C}[\![\Gamma \rhd MN{:}\tau]\!] =$
    $$\mathrm{App}^{\mathcal{C}[\![\sigma]\!],\mathcal{C}[\![\tau]\!]} \circ \langle \mathcal{C}[\![\Gamma \rhd M{:}\sigma \to \tau]\!], \mathcal{C}[\![\Gamma \rhd N{:}\sigma]\!] \rangle$$
  - $\mathcal{C}[\![\Gamma \rhd \lambda x{:}\sigma.M{:}\sigma \to \tau]\!] = \mathrm{Curry}(\mathcal{C}[\![\Gamma, x{:}\sigma \rhd M{:}\tau]\!])$
  - $\mathcal{C}[\![\Gamma_1, x{:}\sigma, \Gamma_2 \rhd M{:}\tau]\!] = \mathcal{C}[\![\Gamma_1, \Gamma_2 \rhd M{:}\tau]\!] \circ \chi_f^{[\![\Gamma_1, x:\sigma, \Gamma_2]\!]}$

Soundness & Completeness

Theorem (Soundness) Given any well typed terms M and N,

if $\Gamma \rhd M \equiv_{\alpha\beta\eta} N : \sigma$, then $[\![\Gamma \rhd M : \sigma]\!] = [\![\Gamma \rhd N : \sigma]\!]$ in very CCC.

Proof

- α-equivalence:
  no term-variable name appears in the calculations
- β-equivalence:
  $[\![\Gamma \rhd (\lambda x : \sigma . M) N : \tau]\!] = [\![\Gamma \rhd [N/x]M : \tau]\!]$
- η-equivalence:
  $[\![\Gamma \rhd \lambda x : \sigma . Mx : \tau]\!] = [\![\Gamma \rhd M : \sigma \to \tau]\!]$

Soundness & Completeness (cont.)

Theorem (Completeness) Given any $\Gamma \rhd M : \sigma$ and $\Gamma \rhd N : \sigma$, there exists a CCC $\mathcal{C}$ such that if $\mathcal{C}[\![\Gamma \rhd M : \sigma]\!] = \mathcal{C}[\![\Gamma \rhd N : \sigma]\!]$, then $\Gamma \rhd M \equiv_{\alpha\beta\eta} N : \sigma$.

Proof

- Construct one category $\mathcal{C}$ by $\lambda^{\rightarrow}$
  - Objects – sequences of types
  - Morphisms – tuples of equivalence classes of terms
- $\mathcal{C}$ is cartesian closed (the most difficult part)
- If $\mathcal{C}$ satisfies $[\![\Gamma \rhd M : \sigma]\!] = [\![\Gamma \rhd N : \sigma]\!]$, then $\Gamma \rhd M \equiv_{\alpha\beta\eta} N : \sigma$.

Conclusions

Appendix

- Connectives VS. Type Constructors

Formulas in intuitionist propositional logic	Types in full simply-typed lambda calculus
False formula $\bot$	Initial type $null$
True formula $\top$ $(\neg\bot)$	Terminal type $unit$
Implication $\varphi_0 \rightarrow \varphi_1$	Function type $\sigma_0 \rightarrow \sigma_1$
Conjunction $\varphi_0 \wedge \varphi_1$	Product type $\sigma_0 \times \sigma_1$
Disjunction $\varphi_0 \vee \varphi_1$	Sum type $\sigma_0 + \sigma_1$

Appendix

- Natural Deduction System VS. Typing Rules

Natural deduction in intuitionist propositional logic	Typing rules in full simply-typed lambda calculus
$\dfrac{}{\Gamma_1, \varphi, \Gamma_2 \vdash \varphi} Axiom$	$\dfrac{}{\Gamma_1, x{:}\sigma, \Gamma_2 \vdash x{:}\sigma} Var$
$\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi} \to I$	$\dfrac{\Gamma, x{:}\sigma \vdash M{:}\tau}{\Gamma \vdash \lambda x.M{:}\sigma \to \tau} \to I$
$\dfrac{\Gamma \vdash \varphi \to \psi,\ \Gamma \vdash \varphi}{\Gamma \vdash \psi} \to E$	$\dfrac{\Gamma \vdash M{:}\sigma \to \tau,\ \Gamma \vdash N{:}\sigma}{\Gamma \vdash MN{:}\tau} \to E$
$\dfrac{\Gamma \vdash \varphi,\ \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} \wedge I$	$\dfrac{\Gamma \vdash M{:}\sigma,\ \Gamma \vdash N{:}\tau}{\Gamma \vdash \langle M, N\rangle{:}\sigma \times \tau} \times I$

Appendix

- Natural Deduction System VS. Typing Rules (cont.)

$\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \wedge E_1$	$\dfrac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \mathrm{Proj}_1^{\sigma,\tau} M : \sigma} \times E_1$
$\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi} \wedge E_2$	$\dfrac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \mathrm{Proj}_2^{\sigma,\tau} M : \tau} \times E_2$
$\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \vee I_1$	$\dfrac{\Gamma \vdash M : \sigma}{\Gamma \vdash \mathrm{Inleft}^{\sigma,\tau} M : \sigma + \tau} + I_1$
$\dfrac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} \vee I_2$	$\dfrac{\Gamma \vdash M : \tau}{\Gamma \vdash \mathrm{Inright}^{\sigma,\tau} M : \sigma + \tau} + I_1$
$\dfrac{\Gamma \vdash \varphi \vee \psi, \Gamma \vdash \varphi \to \chi, \Gamma \vdash \psi \to \chi}{\Gamma \vdash \chi} \vee E$	$\dfrac{\Gamma \vdash M : \sigma + \tau, \Gamma \vdash N : \sigma \to \rho, \Gamma \vdash P : \tau \to \rho}{\Gamma \vdash \mathrm{Case}^{\sigma,\tau,\rho} MNP : \rho} + E$