Curry-Howard Correspondence; Existential Types

CS 1520 (Spring 2025)

Harvard University

Tuesday, March 25, 2025

Today, we will learn about

- Curry-Howard Correspondence
- Existential types

Curry-Howard Correspondence

Curry-Howard Correspondence

- propositions as types
- proofs as programs
- proof normalization as program evaluation

Curry-Howard Correspondence

- a well-typed program demonstrates that there is at least one value for that typed
  - i.e. that type is inhabited
  - a program is a proof that the type is inhabited
- a proof demonstrates that there is at least one way of deriving a formula
  - i.e. that the formula is provable by manipulating assumptions and doing inference
  - a proof is a program that manipulates evidence

Curry-Howard: Implication

$$\text{T-Var}\ \frac{}{\Gamma \vdash x : \tau}\ \Gamma(x) = \tau$$

$$\text{T-Abs}\ \frac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x : \tau.\, e : \tau \to \tau'}$$

$$\text{T-App}\ \frac{\Gamma \vdash e_1 : \tau \to \tau' \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1\ e_2 : \tau'}$$

Curry-Howard: Implication

$$\text{T-Var}\ \frac{}{\Gamma_1, x : A, \Gamma_2 \vdash x : A}$$

$$\text{T-Abs}\ \frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x : A.\, e : A \to B}$$

$$\text{T-App}\ \frac{\Gamma \vdash e_1 : A \to B \quad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1\ e_2 : B}$$

Curry-Howard: Implication

$$\textsc{T-Var}\ \frac{}{\Gamma_1, x : A\ , \Gamma_2 \vdash x : A}$$

$$\textsc{T-Abs}\ \frac{\Gamma, x : A\ \vdash e : B}{\Gamma \vdash \lambda x : A.\ e : A \to B}$$

$$\textsc{T-App}\ \frac{\Gamma \vdash e_1 : A \to B \quad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1\ e_2 : B}$$

Curry-Howard: Implication

$$\textsc{T-Var}\ \frac{}{\overline{\Gamma}_1, \quad A \ , \overline{\Gamma}_2 \vdash \quad A}$$

$$\textsc{T-Abs}\ \frac{\overline{\Gamma}, \quad A \ \vdash \quad B}{\overline{\Gamma} \vdash \qquad\qquad A \Rightarrow B}$$

$$\textsc{T-App}\ \frac{\overline{\Gamma} \vdash \quad A \Rightarrow B \quad \overline{\Gamma} \vdash \quad A}{\overline{\Gamma} \vdash \qquad B}$$

Conjunction = Product

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \#1\ e : \tau_1} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \#2\ e : \tau_2}$$

Conjunction = Product

$$\frac{\Gamma \vdash e_1 : A \quad \Gamma \vdash e_2 : B}{\Gamma \vdash (e_1, e_2) : A \times B}$$

$$\frac{\Gamma \vdash e : A \times B}{\Gamma \vdash \#1\ e : A} \qquad \frac{\Gamma \vdash e : A \times B}{\Gamma \vdash \#2\ e : B}$$

Conjunction = Product

$$\frac{\Gamma \vdash e_1 : A \quad \Gamma \vdash e_2 : B}{\Gamma \vdash (e_1, e_2) : A \times B}$$

$$\frac{\Gamma \vdash e : A \times B}{\Gamma \vdash \#1\ e : A} \qquad \frac{\Gamma \vdash e : A \times B}{\Gamma \vdash \#2\ e : B}$$

Conjunction = Product

$$\frac{\overline{\Gamma} \vdash \quad A \qquad \overline{\Gamma} \vdash \quad B}{\overline{\Gamma} \vdash \qquad A \wedge B}$$

$$\frac{\overline{\Gamma} \vdash \quad A \wedge B}{\overline{\Gamma} \vdash \qquad A} \qquad \frac{\overline{\Gamma} \vdash \quad A \wedge B}{\overline{\Gamma} \vdash \qquad B}$$

Disjunction = Sum

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathsf{inl}_{\tau_1+\tau_2}\ e : \tau_1 + \tau_2} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathsf{inr}_{\tau_1+\tau_2}\ e : \tau_1 + \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \quad \Gamma \vdash e_1 : \tau_1 \to \tau \quad \Gamma \vdash e_2 : \tau_2 \to \tau}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ e_1 \mid e_2 : \tau}$$

Disjunction = Sum

$$\frac{\Gamma \vdash e : A}{\Gamma \vdash \mathsf{inl}_{A+B}\ e : A + B} \qquad \frac{\Gamma \vdash e : B}{\Gamma \vdash \mathsf{inr}_{A+B}\ e : A + B}$$

$$\frac{\Gamma \vdash e : A + B \quad \Gamma \vdash e_1 : A \to C \quad \Gamma \vdash e_2 : B \to C}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ e_1 \mid e_2 : C}$$

Disjunction = Sum

$$\frac{\Gamma \vdash e : A}{\Gamma \vdash \mathsf{inl}_{A+B}\ e : A + B} \qquad \frac{\Gamma \vdash e : B}{\Gamma \vdash \mathsf{inr}_{A+B}\ e : A + B}$$

$$\frac{\Gamma \vdash e : A + B \quad \Gamma \vdash e_1 : A \to C \quad \Gamma \vdash e_2 : B \to C}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ e_1 \mid e_2 : C}$$

Disjunction = Sum

$$\frac{\overline{\Gamma} \vdash \quad A}{\overline{\Gamma} \vdash \qquad A \vee B} \qquad \frac{\overline{\Gamma} \vdash \quad B}{\overline{\Gamma} \vdash \qquad A \vee B}$$

$$\frac{\overline{\Gamma} \vdash \quad A \vee B \quad \overline{\Gamma} \vdash \quad A \Rightarrow C \quad \overline{\Gamma} \vdash \quad B \Rightarrow C}{\overline{\Gamma} \vdash \qquad C}$$

Parametric Polymorphism

What about False?

$$\textsc{Ex Falso Quodlibet}\ \frac{\overline{\Gamma} \vdash \bot}{\overline{\Gamma} \vdash A}$$

Example 1: From Formula to Type

$$\forall \phi_1, \phi_2, \phi_3.\ ((\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_3)) \Rightarrow (\phi_1 \Rightarrow \phi_3).$$

Example 1: From Formula to Type

$$\forall \phi_1, \phi_2, \phi_3.\ ((\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_3)) \Rightarrow (\phi_1 \Rightarrow \phi_3).$$

$$\forall X, Y, Z.\ ((X \to Y) \times (Y \to Z)) \to (X \to Z).$$

Example 1: From Formula to Type

$$\forall \phi_1, \phi_2, \phi_3.\ ((\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_3)) \Rightarrow (\phi_1 \Rightarrow \phi_3).$$

$$\forall X, Y, Z.\ ((X \to Y) \times (Y \to Z)) \to (X \to Z).$$

$$\Lambda X, Y, Z.\, \lambda f : (X \to Y) \times (Y \to Z).\, \lambda x : X.\, (\#2\ f)\ ((\#1\ f)\ x)$$

Example 2: From Type to Formula

$$\lambda f : (\tau_1 \times \tau_2) \to \tau_3 .\, \lambda x : \tau_1 .\, \lambda y : \tau_2 .\, f\ (x, y)$$

Example 2: From Type to Formula

$$\lambda f : (\tau_1 \times \tau_2) \to \tau_3 .\, \lambda x : \tau_1 .\, \lambda y : \tau_2 .\, f\ (x, y)$$

$$((\tau_1 \times \tau_2) \to \tau_3) \to (\tau_1 \to \tau_2 \to \tau_3)$$

Example 2: From Type to Formula

$$\lambda f : (\tau_1 \times \tau_2) \to \tau_3 .\, \lambda x : \tau_1 .\, \lambda y : \tau_2 .\, f\ (x, y)$$

$$((\tau_1 \times \tau_2) \to \tau_3) \to (\tau_1 \to \tau_2 \to \tau_3)$$

$$(\phi_1 \wedge \phi_2 \Rightarrow \phi_3) \Rightarrow (\phi_1 \Rightarrow (\phi_2 \Rightarrow \phi_3))$$

Negation

$\neg\tau$ equivalent to $\tau \Rightarrow$ **False**
If $\neg\tau$ is true, then if you give me a proof of τ, I can give you a proof of **False**.

Which are tautologies?

1. $A \Rightarrow B \Rightarrow A$
2. $(A \Rightarrow B) \Rightarrow A \Rightarrow B$
3. $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
4. $A \vee \neg A$
5. $\neg\neg(A \vee \neg A)$
6. $A \Rightarrow \neg\neg A$
7. $\neg\neg A \Rightarrow A$

Which are tautologies?

1. First axiom of sentential logic: $A \Rightarrow B \Rightarrow A$
2. Modus Ponens: $(A \Rightarrow B) \Rightarrow A \Rightarrow B$
3. Second axiom of sentential logic: $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
4. Excluded Middle (does not hold in intuitionistic logic, only in classical logic): $A \vee \neg A$
5. Excluded Middle is not wrong (holds in intuitionistic logic too): $\neg\neg(A \vee \neg A)$
6. If A is right, then it's not wrong: $A \Rightarrow \neg\neg A$
7. If A is not wrong, then it's right (does not hold in intuitionistic logic): $\neg\neg A \Rightarrow A$

Double negation

- Double negation $\neg\neg P$ reads as "P is not wrong".
- In classical logic, $\neg\neg P$ is equivalent to P, and "it's not wrong" is equivalent to "it's true".
- In intuitionistic logic, P implies $\neg\neg P$ but the converse does not hold (no double negation elimination). Hence, "it's not wrong" is weaker than "it's true".
- However, the intuitionistic "it's not wrong" behaves much like the classical "it's true". In particular, excluded middle is not wrong!

Continuations

Answer: "return type" of continuations
Continuation type: $\tau \longrightarrow$ **Answer**.
Assume **Answer** is uninhabited – like **False**.
Then, continuation corresponds to negation!

Continuation-Passing Style

The type of $\mathcal{CPS}[\![e]\!]$ is
$([\![\tau]\!] \to \textbf{Answer}) \to \textbf{Answer}$.
This type corresponds to $\neg(\neg[\![\tau]\!])$.
Double negation.

Proof Normalization is Program Evaluation

β-reduction corresponds to cut elimination

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma, x : A \vdash e : B}}{\Gamma \vdash \lambda x : A.\, e : A \to B} \qquad \dfrac{\vdots}{\Gamma \vdash e_2 : A}}{\Gamma \vdash (\lambda x : A.\, e)\, e_2 : B}$$

becomes

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash e_2 : A}}{\vdots}}{\Gamma \vdash e\{e_2/x\} : B}$$

β-reduction corresponds to cut elimination

$$\dfrac{\dfrac{\dfrac{\vdots}{\overline{\Gamma},\quad A \vdash \quad B}}{\overline{\Gamma} \vdash \qquad\qquad A \Rightarrow B} \qquad \dfrac{\vdots}{\overline{\Gamma} \vdash \quad A}}{\overline{\Gamma} \vdash \qquad\qquad B}$$

becomes

$$\dfrac{\dfrac{\dfrac{\vdots}{\overline{\Gamma} \vdash \quad A}}{\vdots}}{\overline{\Gamma} \vdash \qquad B}$$

Cuts

A cut is an intermediate statement (a lemma) that we prove even though it is not a subformula of the final statement (the theorem).

Example (A proof with a cut). We show that P, then $P \Rightarrow Q$, and we conclude Q.
P is a cut.

Example (A proof without cuts). We show P, then Q, and we conclude $P \wedge Q$.
P and Q are subformulas of $P \wedge Q$ and therefore not cuts.

Curry-Howard Correspondence

Existential Types

Syntax

$$
\begin{aligned}
e ::= {} & x \mid \lambda x{:}\tau.\, e \mid e_1\ e_2 \mid n \mid e_1 + e_2 \\
& \mid \{\ l_1 = e_1, \ldots, l_n = e_n\ \} \mid e.l \\
& \mid \mathsf{pack}\ \{\tau_1, e\}\ \mathsf{as}\ \exists X.\ \tau_2 \\
& \mid \mathsf{unpack}\ \{X, x\} = e_1\ \mathsf{in}\ e_2
\end{aligned}
$$

$$
\begin{aligned}
v ::= {} & n \mid \lambda x{:}\tau.\, e \mid \{\ l_1 = v_1, \ldots, l_n = v_n\ \} \\
& \mid \mathsf{pack}\ \{\tau_1, v\}\ \mathsf{as}\ \exists X.\ \tau_2
\end{aligned}
$$

$$
\tau ::= \mathbf{int} \mid \tau_1 \to \tau_2 \mid \{\ l_1{:}\tau_1, \ldots, l_n{:}\tau_n\ \} \mid X \mid \exists X.\ \tau
$$

Example: Counter ADT

$$
\begin{array}{l}
\text{let } \mathit{counterADT} = \\
\quad \text{pack} \\
\quad\quad \{\mathbf{int}, \{\ \text{new} = 0, \\
\quad\quad\quad\quad \text{get} = \lambda i : \mathbf{int}.\ i, \\
\quad\quad\quad\quad \text{inc} = \lambda i : \mathbf{int}.\ i + 1\ \}\ \} \\
\quad \text{as} \\
\quad\quad \exists \mathbf{Counter}.\ \{\ \text{new} : \mathbf{Counter}, \\
\quad\quad\quad\quad \text{get} : \mathbf{Counter} \to \mathbf{int}, \\
\quad\quad\quad\quad \text{inc} : \mathbf{Counter} \to \mathbf{Counter}\ \} \\
\text{in } \ldots
\end{array}
$$

Example: Counter ADT, continued

$$\mathsf{unpack}\ \{C, x\} = \mathit{counterADT}\ \mathsf{in}\ \mathsf{let}\ y : C = x.\mathsf{new}\ \mathsf{in}$$
$$x.\mathsf{get}\ (x.\mathsf{inc}\ (x.\mathsf{inc}\ y))$$

Operational Semantics

$$E ::= \cdots \mid \mathsf{pack}\ \{\tau_1, E\}\ \mathsf{as}\ \exists X.\ \tau_2$$
$$\mid \mathsf{unpack}\ \{X, x\} = E\ \mathsf{in}\ e$$

$$\mathsf{unpack}\ \{X, x\} = (\mathsf{pack}\ \{\tau_1, v\}\ \mathsf{as}\ \exists Y.\ \tau_2)\ \mathsf{in}\ e \longrightarrow e\{v/x\}\{\tau_1/X\}$$

Typing rules

$$\frac{\Delta, \Gamma \vdash e : \tau_2\{\tau_1/X\}}{\Delta, \Gamma \vdash \mathsf{pack}\ \{\tau_1, e\}\ \mathsf{as}\ \exists X.\ \tau_2 : \exists X.\ \tau_2}$$

$$\frac{\Delta, \Gamma \vdash e_1 : \exists X.\ \tau_1 \quad X \notin \Delta \qquad \Delta \cup \{X\}, \Gamma, x : \tau_1 \vdash e_2 : \tau_2 \quad \Delta \vdash \tau_2\ \mathsf{ok}}{\Delta, \Gamma \vdash \mathsf{unpack}\ \{X, x\} = e_1\ \mathsf{in}\ e_2 : \tau_2}$$

$$\frac{\Delta \cup \{X\} \vdash \tau\ \mathsf{ok}}{\Delta \vdash \exists X.\ \tau\ \mathsf{ok}}$$

Exisentials encoded with Universals

$$\begin{aligned} & \exists X.\ \tau \\ =& \forall Y.(\forall X.\tau \to Y) \to Y \end{aligned}$$

$$\begin{aligned} & \text{pack } \{\tau_0, e\} \text{ as } \exists X.\ \tau \\ =& \Lambda Y.\ \lambda f : (\forall X.\tau \to Y).\ f\ [\tau_0]e \end{aligned}$$

$$\begin{aligned} & \text{unpack } \{X, x\} = e_1 \text{ in } e_2 \\ =& e_1\ [\tau_2]\ (\Lambda X.\ \lambda x : \tau_{11}.\ e_2). \end{aligned}$$