

Curry-Howard Correspondence; Existential Types

CS 152 (Spring 2022)

Harvard University

Tuesday, March 22, 2022

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 type
 - ▶ 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

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

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

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

Curry-Howard: Implication

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

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

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

Curry-Howard: Implication

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

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

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

Curry-Howard: Implication

$$\text{T-VAR} \frac{}{\bar{\Gamma}_1, A, \bar{\Gamma}_2 \vdash A}$$

$$\text{T-ABS} \frac{\bar{\Gamma}, A \vdash B}{\bar{\Gamma} \vdash A \Rightarrow B}$$

$$\text{T-APP} \frac{\bar{\Gamma} \vdash A \Rightarrow B \quad \bar{\Gamma} \vdash A}{\bar{\Gamma} \vdash B}$$

Conjunction = Product

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

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

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

Conjunction = Product

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

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

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

Conjunction = Product

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

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

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

Conjunction = Product

$$\frac{\bar{\Gamma} \vdash A \quad \bar{\Gamma} \vdash B}{\bar{\Gamma} \vdash A \wedge B}$$

$$\frac{\bar{\Gamma} \vdash A \wedge B}{\bar{\Gamma} \vdash A}$$

$$\frac{\bar{\Gamma} \vdash A \wedge B}{\bar{\Gamma} \vdash B}$$

Disjunction = Sum

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \text{inl}_{\tau_1 + \tau_2} e : \tau_1 + \tau_2} \quad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \text{inr}_{\tau_1 + \tau_2} e : \tau_1 + \tau_2}$$
$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \quad \Gamma \vdash e_1 : \tau_1 \rightarrow \tau \quad \Gamma \vdash e_2 : \tau_2 \rightarrow \tau}{\Gamma \vdash \text{case } e \text{ of } e_1 \mid e_2 : \tau}$$

Disjunction = Sum

$$\frac{\Gamma \vdash e:A}{\Gamma \vdash \text{inl}_{A+B} e:A+B} \quad \frac{\Gamma \vdash e:B}{\Gamma \vdash \text{inr}_{A+B} e:A+B}$$
$$\frac{\Gamma \vdash e:A+B \quad \Gamma \vdash e_1:A \rightarrow C \quad \Gamma \vdash e_2:B \rightarrow C}{\Gamma \vdash \text{case } e \text{ of } e_1 \mid e_2:C}$$

Disjunction = Sum

$$\frac{\Gamma \vdash e:A}{\Gamma \vdash \text{inl}_{A+B} e:A+B} \quad \frac{\Gamma \vdash e:B}{\Gamma \vdash \text{inr}_{A+B} e:A+B}$$
$$\frac{\Gamma \vdash e:A+B \quad \Gamma \vdash e_1:A \rightarrow C \quad \Gamma \vdash e_2:B \rightarrow C}{\Gamma \vdash \text{case } e \text{ of } e_1 \mid e_2:C}$$

Disjunction = Sum

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

Parametric Polymorphism

What about False?

$$\text{EX FALSO QUODLIBET} \frac{\bar{\Gamma} \vdash \perp}{\bar{\Gamma} \vdash A}$$

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

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

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

Example 1: From Formula to Type

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

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

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

Example 2: From Type to Formula

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

Example 2: From Type to Formula

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

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

Example 2: From Type to Formula

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

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

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

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: $\tau \rightarrow$ **Answer**.

Assume **Answer** is uninhabited – like **False**.

Then, continuation corresponds to negation!

Continuation-Passing Style

The type of $\mathcal{CPS}\llbracket e \rrbracket$ is

$(\llbracket \tau \rrbracket \rightarrow \mathbf{Answer}) \rightarrow \mathbf{Answer}$.

This type corresponds to $\neg(\neg\llbracket \tau \rrbracket)$.

Double negation.

Proof Normalization is Program Evaluation

β -reduction corresponds to cut elimination

$$\frac{\frac{\frac{\vdots}{\Gamma, x:A \vdash e:B}}{\Gamma \vdash \lambda x:A. e:A \rightarrow B} \quad \frac{\vdots}{\Gamma \vdash e_2:A}}{\Gamma \vdash (\lambda x:A. e)e_2:B}}$$

becomes

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

β -reduction corresponds to cut elimination

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

becomes

$$\frac{\frac{\frac{\vdots}{\overline{\Gamma \vdash A}}}{\vdots}}{\overline{\Gamma \vdash B}}$$

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

$$e ::= x \mid \lambda x:\tau. e \mid e_1 e_2 \mid n \mid e_1 + e_2 \\ \mid \{ l_1 = e_1, \dots, l_n = e_n \} \mid e.l \\ \mid \text{pack } \{ \tau_1, e \} \text{ as } \exists X. \tau_2 \\ \mid \text{unpack } \{ X, x \} = e_1 \text{ in } e_2$$
$$v ::= n \mid \lambda x:\tau. e \mid \{ l_1 = v_1, \dots, l_n = v_n \} \\ \mid \text{pack } \{ \tau_1, v \} \text{ as } \exists X. \tau_2$$
$$\tau ::= \mathbf{int} \mid \tau_1 \rightarrow \tau_2 \mid \{ l_1:\tau_1, \dots, l_n:\tau_n \} \mid X \mid \exists X. \tau$$

Example: Counter ADT

```
let counterADT =  
  pack  
    { int, { new = 0,  
             get =  $\lambda i:\mathbf{int}. i$ ,  
             inc =  $\lambda i:\mathbf{int}. i + 1$  } }  
  as  
     $\exists$  Counter. { new : Counter,  
                  get : Counter  $\rightarrow$  int,  
                  inc : Counter  $\rightarrow$  Counter }  
in ...
```

Example: Counter ADT, continued

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

Operational Semantics

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

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

Typing rules

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

$$\frac{\begin{array}{l} \Delta, \Gamma \vdash e_1 : \exists X. \tau_1 \quad X \notin \Delta \\ \Delta \cup \{ X \}, \Gamma, x : \tau_1 \vdash e_2 : \tau_2 \quad \Delta \vdash \tau_2 \text{ ok} \end{array}}{\Delta, \Gamma \vdash \text{unpack } \{ X, x \} = e_1 \text{ in } e_2 : \tau_2}$$

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

Existentials encoded with Universals

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

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

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