CIS 624: Structure of Programming Languages
The Curry-Howard Isomorphism

Philip Johnson-Freyd

November 18, 2015
The Lecture of Three Lies

Today we learn about the “Curry-Howard Isomorphism”
The Lecture of Three Lies

Today we learn about the “Curry-Howard Isomorphism”

1. Not invented by Haskell Curry
The Lecture of Three Lies

Today we learn about the “Curry-Howard Isomorphism”

1. Not invented by Haskell Curry

2. Not invented by William A. Howard
The Lecture of Three Lies

Today we learn about the “Curry-Howard Isomorphism”

1. Not invented by Haskell Curry

2. Not invented by William A. Howard

3. And, not an isomorphism!

So we are off to a good start...
Curry-Howard Isomorphism

Logics $\equiv$ Programming Languages

- “Proofs as Programs”
- “Propositions as Types”
Curry-Howard Isomorphism

Logics $\equiv$ Programming Languages

- “Proofs as Programs”
- “Propositions as Types”

Extends to “Curry-Howard-Lambek” Correspondence

- Logics = Programming = Algebra/Categories
- “The central dogma of computational trinitarianism holds that Logic, Languages, and Categories are but three manifestations of one divine notion of computation.” (Bob Harper)
What Logicians Do

Define a syntax for a logic

\[ P, Q, R \in \text{Propositions} ::= P \lor Q \mid P \land Q \mid P \supset Q \]

- Semantics
- Proof Rules
Natural Deduction

Rules “as close as possible to actual reason” (Gentzen, 1934)

Introduction Rules

\[
\begin{align*}
\frac{P}{P \lor Q} & \quad \frac{Q}{P \lor Q} \\
\end{align*}
\]
Natural Deduction

Rules “as close as possible to actual reason” (Gentzen, 1934)

Introduction Rules

\[
\begin{align*}
P & \quad Q \\
\hline
P \lor Q & \\
\hline
P \lor Q & \\
\hline
P \land Q & \\
\hline
P \land Q & \\
\hline
\end{align*}
\]

what about implication?
Natural Deduction

Rules “as close as possible to actual reason” (Gentzen, 1934)

Introduction Rules

\[
\begin{align*}
P & \quad Q \\
\hline
& P \lor Q \\
& P \lor Q \\

& P \\
& \hline
& Q \\
& P \land Q
\end{align*}
\]

what about implication?
Hypothetical Judgments

\[ \Gamma \vdash P \] where \( \Gamma \) is a set of formula

\[
P \in \Gamma \\
\hline
\Gamma \vdash P
\]
Hypothetical Judgments

\[ \Gamma \vdash P \text{ where } \Gamma \text{ is a set of formula} \]

\[ \frac{P \in \Gamma}{\Gamma \vdash P} \]

\[ \frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \lor Q} \]

\[ \frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \land Q} \]

\[ \frac{\Gamma, P \vdash Q}{\Gamma \vdash P \supset Q} \]
Elimination Rules

\[
\frac{\Gamma \vdash P \land Q}{\Gamma \vdash P} \quad \frac{\Gamma \vdash P \land Q}{\Gamma \vdash Q}
\]
Elimination Rules

\[ \Gamma \vdash P \land Q \quad \Gamma \vdash P \land Q \]

\[ \Gamma \vdash P \quad \Gamma \vdash Q \]

\[ \Gamma \vdash P \supset Q \quad \Gamma \vdash P \]

\[ \Gamma \vdash Q \]
Elimination Rules

\[
\frac{\Gamma \vdash P \land Q}{\Gamma \vdash P} \quad \frac{\Gamma \vdash P \land Q}{\Gamma \vdash Q}
\]

\[
\frac{\Gamma \vdash P \supset Q \quad \Gamma \vdash P}{\Gamma \vdash Q}
\]

\[
\frac{\Gamma \vdash P \lor Q \quad \Gamma, P \vdash R \quad \Gamma, Q \vdash R}{\Gamma \vdash R}
\]
Local Soundness

Intro, then elim

\[
\begin{align*}
\text{proof1} & : \Gamma \vdash P \\
\text{proof2} & : \Gamma \vdash P \\
\hline
P \land Q & \\
\hline
P &
\end{align*}
\]
Local Soundness

Intro, then elim

\[
\frac{\text{proof1}}{\Gamma \vdash P} \quad \frac{\text{proof2}}{\Gamma \vdash P} \\
\hline
P \land Q \quad \Rightarrow \quad \text{proof1} \\
\hline
\frac{P}{\Gamma \vdash P}
\]

(we can use this to prove consistency!)
Local Soundness

Intro, then elim

\[
\frac{\text{proof1}}{\Gamma \vdash P} \quad \frac{\text{proof2}}{\Gamma \vdash P}
\]

\[
\frac{}{P \land Q}
\]

\[
\frac{}{P}
\]

\[
\Rightarrow \quad \frac{}{\Gamma \vdash P}
\]

\[
\frac{\text{proof1}}{\Gamma, P \vdash Q} \quad \frac{\text{proof2}}{\Gamma \vdash P}
\]

\[
\frac{\Gamma \vdash P \supset Q \quad \Gamma \vdash P}{\Gamma \vdash Q}
\]

(we can use this to prove consistency!)
Local Soundness

Intro, then elim

\[
\begin{align*}
&\text{proof1} \quad \text{proof2} \\
&\frac{}{\Gamma \vdash P} \quad \frac{}{\Gamma \vdash P} \\
&\frac{}{P \land Q} \quad \frac{}{P \Rightarrow \Gamma \vdash P}
\end{align*}
\]

\[
\begin{align*}
&\text{proof1} \quad \text{proof2} \\
&\frac{}{\Gamma, P \vdash Q} \quad \frac{}{\Gamma \vdash P} \\
&\frac{}{\Gamma \vdash P \supset Q} \quad \frac{}{\Gamma \vdash P} \\
&\text{proof1 with proof2 for } P \\
&\frac{}{\Gamma \vdash Q} \quad \frac{}{\Gamma \vdash Q}
\end{align*}
\]

(we can use this to prove consistency!)
STLC

\[
T, S, R \in \text{Type} ::= T + S \mid T \times S \mid T \rightarrow S
\]

\[
e, e_1, e_2 \in \text{Terms} ::= \lambda x^T : e \mid e_1 \ e_2 \mid \langle e_1, e_2 \rangle \mid e.1 \mid e.2
\]

\[
\mid A(e) \mid B(e) \mid \text{match } e \text{ with } A(x).e_1; B(x).e_2
\]

\[
\frac{\Gamma \vdash e_1 : T \quad \Gamma \vdash e_2 : S}{\Gamma \vdash \langle e_1, e_2 \rangle : T \times S}
\quad
\frac{\Gamma \vdash e : T \times S}{\Gamma \vdash e.1 : T}
\quad
\frac{\Gamma \vdash e : T \times S}{\Gamma \vdash e.2 : S}
\]

\[
\frac{\Gamma \vdash e : T \quad \Gamma \vdash e : S}{\Gamma \vdash A(e) : T + S}
\quad
\frac{\Gamma \vdash e : T \quad \Gamma \vdash e : S}{\Gamma \vdash B(e) : T + S}
\quad
\frac{\Gamma \vdash e : T + S \quad \Gamma, x : T \vdash e_1 : R \quad \Gamma, x : S \vdash e_1 : R}{\Gamma \vdash \text{match } e \text{ with } A(x).e_1; B(x).e_2 : R}
\]

\[
\frac{\Gamma, x : T \vdash e : S}{\Gamma \vdash \lambda x^T . e : T \rightarrow S}
\quad
\frac{\Gamma \vdash e_1 : T \rightarrow S \quad \Gamma \vdash e_2 : T}{\Gamma \vdash x : T \in \Gamma}
\quad
\frac{\Gamma \vdash e_1 \ e_2 : S}{\Gamma \vdash x : T}
\]
Remarkable Coincidence–Forget the Terms

\[ T, S, R \in \text{Type} ::= T + S \mid T \times S \mid T \rightarrow S \]

\[
\begin{align*}
\Gamma \vdash T & \quad \Gamma \vdash S \\
\quad & \quad \Gamma \vdash T \times S \\
\quad & \quad \Gamma \vdash T \\
\quad & \quad \Gamma \vdash S
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash T & \quad \Gamma \vdash T \\
\quad & \quad \Gamma \vdash T + S \\
\quad & \quad \Gamma \vdash T + S \\
\quad & \quad \Gamma \vdash T \rightarrow S
\end{align*}
\]

\[
\begin{align*}
\Gamma, T \vdash S & \quad \Gamma \vdash T \rightarrow S \\
\quad & \quad \Gamma \vdash T \\
\quad & \quad \Gamma \vdash T \\
\quad & \quad \Gamma \vdash T
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash R & \quad \Gamma, T \vdash R \\
\quad & \quad \Gamma, S \vdash R \\
\quad & \quad \Gamma \vdash R \\
\end{align*}
\]
Remarkable Coincidence–Change Notation

\[ P, Q, R \in \text{Propositions} ::= P \lor Q \mid P \land Q \mid P \supset Q \]

\[ \Gamma \vdash P \quad \Gamma \vdash Q \quad \Gamma \vdash P \land Q \quad \Gamma \vdash P \quad \Gamma \vdash Q \]

\[ \Gamma \vdash P \quad \Gamma \vdash Q \quad \Gamma \vdash P \lor Q \quad \Gamma \vdash P \lor Q \quad \Gamma \vdash P \lor Q \]

\[ \Gamma, P \vdash R \quad \Gamma, Q \vdash R \quad \Gamma \vdash P \lor Q \quad \Gamma \vdash P \lor Q \quad \Gamma \vdash P \lor Q \quad P \in \Gamma \]

\[ \Gamma \vdash P \supset Q \quad \Gamma \vdash P \supset Q \quad \Gamma \vdash P \quad \Gamma \vdash Q \quad \Gamma \vdash P \]

Its Propositional Logic! and local soundness is just reduction!!!
Remarkable Coincidence–Change Notation

\[ P, Q, R \in \text{Propositions ::= } P \lor Q \mid P \land Q \mid P \supset Q \]

\[
\begin{align*}
\Gamma \vdash P \quad \Gamma \vdash Q & \quad \frac{\Gamma \vdash P \land Q}{\Gamma \vdash P} \quad \frac{\Gamma \vdash P \land Q}{\Gamma \vdash Q} \\
\Gamma \vdash P \quad \Gamma \vdash Q & \quad \frac{\Gamma \vdash P \lor Q}{\Gamma \vdash P} \quad \frac{\Gamma \vdash P \lor Q}{\Gamma \vdash Q} \\
\Gamma, P \vdash Q & \quad \frac{\Gamma, P \vdash Q}{\Gamma \vdash P \supset Q} \quad \frac{\Gamma \vdash P}{\Gamma \vdash P \supset Q} \\
\Gamma \vdash Q & \quad \frac{\Gamma \vdash P}{\Gamma \vdash Q} \quad \frac{\Gamma \vdash P}{\Gamma \vdash P}
\end{align*}
\]

Its Propositional Logic!
Remarkable Coincidence–Change Notation

\[ P, Q, R \in \text{Propositions} ::= P \lor Q \mid P \land Q \mid P \supset Q \]

\[
\begin{align*}
\Gamma \vdash P & \quad \Gamma \vdash Q \\
\hline
\Gamma \vdash P \land Q
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash P & \quad \Gamma \vdash P \land Q \\
\hline
\Gamma \vdash P
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash P \land Q & \quad \Gamma \vdash P \land Q \\
\hline
\Gamma \vdash P
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash P & \quad \Gamma \vdash Q \\
\hline
\Gamma \vdash P \lor Q
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash P & \quad \Gamma \vdash P \lor Q \\
\hline
\Gamma \vdash P
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash P & \quad \Gamma \vdash P \lor Q \\
\hline
\Gamma \vdash P
\end{align*}
\]

\[
\begin{align*}
\Gamma, P \vdash Q & \quad \Gamma, P \vdash Q \\
\hline
\Gamma \vdash P \supset Q
\end{align*}
\]

\[
\begin{align*}
\Gamma, Q \vdash R & \quad \Gamma, Q \vdash R \\
\hline
\Gamma \vdash R
\end{align*}
\]

\[
\begin{align*}
\Gamma, P \vdash Q & \quad \Gamma, P \vdash Q \\
\hline
\Gamma \vdash P \supset Q
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash P & \quad \Gamma \vdash P \\
\hline
\Gamma \vdash P \supset Q
\end{align*}
\]

\[
\begin{align*}
P \in \Gamma & \\
\hline
\Gamma \vdash P
\end{align*}
\]

Its Propositional Logic! and local soundness is just reduction!!!!!
A magic trick

Which of the following types are inhabited by a closed term?

\[
\begin{align*}
b_{17} &\to b_{17} \\
b_1 &\to (b_1 \to b_2) \to b_2 \\
(b_1 \to b_2 \to b_3) &\to b_2 \to b_1 \to b_3 \\
b_1 &\to ((b_1 + b_7) \times (b_1 + b_4)) \\
(b_1 \to b_3) &\to (b_2 \to b_3) \to (b_1 + b_2) \to b_3 \\
(b_1 \times b_2) &\to b_3 \to ((b_3 \times b_1) \times b_2) \\
b_1 &\to b_2 \\
b_1 &\to (b_2 \to b_1) \to b_2
\end{align*}
\]
Which of the following types are inhabited by a closed term?

\[
\begin{align*}
    b_{17} & \rightarrow b_{17} \\
    b_1 & \rightarrow (b_1 \rightarrow b_2) \rightarrow b_2 \\
    (b_1 \rightarrow b_2 \rightarrow b_3) & \rightarrow b_2 \rightarrow b_1 \rightarrow b_3 \\
    b_1 & \rightarrow ((b_1 + b_7) \times (b_1 + b_4)) \\
    (b_1 \rightarrow b_3) & \rightarrow (b_2 \rightarrow b_3) \rightarrow (b_1 + b_2) \rightarrow b_3 \\
    (b_1 \times b_2) & \rightarrow b_3 \rightarrow ((b_3 \times b_1) \times b_2) \\
    b_1 & \rightarrow b_2 \\
    b_1 & \rightarrow (b_2 \rightarrow b_1) \rightarrow b_2 \\
\end{align*}
\]

(Curry-Howard means you can ask your computer)
A problem

Aristotle says that everything is either true or false (Excluded Middle).

\[ P \lor \neg P = P \lor (P \rightarrow \text{False}). \]
A problem

Aristotle says that everything is either true or false (Excluded Middle).
\[ P \lor \neg P = P \lor (P \rightarrow \text{False}) . \]
Since false implies anything, that is the same as saying that for all \( P \) and \( Q \):
\[ \vdash P \lor (P \rightarrow Q) \]
A problem

Aristotle says that everything is either true or false (Excluded Middle).

\[ P \lor \neg P = P \lor (P \rightarrow \text{False}). \]

Since false implies anything, that is the same as saying that for all \( P \) and \( Q \):

\[ \vdash P \lor (P \rightarrow Q) \]

can we construct a proof of that? Equivalently, give me an \( e \) of type

\[ P + (P \rightarrow Q) \]

which works no matter what \( P \) and \( Q \) are
Aristotle’s logic with excluded middle is called “Classical Logic”
Logic without exclude middle is called “Intuitionistic” or “Constructive”
STLC = Intuitionistic propositional logic
Intuitionism part 1

Some questions we don’t know, can’t know

- This program either halts or it doesn’t.
- $P = NP$ or $P \neq NP$.
- The axiom of choice is true or it is not.
Intuitionism part 2

Classical Proof is Weird!
Theorem: there is a student in this class such that if that student gets an A everyone gets an A.
Intuitionism part 2

Classical Proof is Weird!
Theorem: there is a student in this class such that if that student gets an A everyone gets an A. Proof:
  ▶ either everyone in this class gets an A or not everyone gets an A.
Classical Proof is Weird!
Theorem: there is a student in this class such that if that student gets an A everyone gets an A. Proof:

- either everyone in this class gets an A or not everyone gets an A.
- in the first case the student is you.
Classical Proof is Weird!
Theorem: there is a student in this class such that if that student gets an A everyone gets an A. Proof:

▶ either everyone in this class gets an A or not everyone gets an A.
▶ in the first case the student is you.
▶ in the second case the student is one of the ones who won’t get an A
Intuitionism part 2

Classical Proof is Weird!
Theorem: there is a student in this class such that if that student gets an A everyone gets an A. Proof:

- either everyone in this class gets an A or not everyone gets an A.
- in the first case the student is you.
- in the second case the student is one of the ones who won’t get an A (so not you).
Tarski Style Semantics

What does “$P$ is true” mean?

Induction on $P$:

- “$P \land Q$ true” means $P$ is true and $Q$ is true.
- “$P \lor Q$ true” means $P$ is true or $Q$ is true.
- “$P \supset Q$ true” means that if $P$ is true then $Q$ is true.

“The meaning of Broccoli is Broccoli” (Girard)
Brouwer-Heyting-Kolmogorov Interpretation

Idea: “$P$ is true” means “I have a $P$ widget and I know why.”

▶ A “$P \land Q$ widget” is a “$P$ widget” and a “$Q$ widget”
▶ A “$P \lor Q$ widget” is a “$P$ widget” or it is a “$Q$ widget.”
▶ A “$P \supset Q$ widget” is a widget such that if you give it a “$P$ widget” it will give you back a “$Q$ widget”

must be a widget! That is, some sort of *thing* called a construction.
Idea: “$P$ is true” means “I have a $P$ widget and I know why.”

- A “$P \land Q$ widget” is a pair of a “$P$ widget” and a “$Q$ widget”
- A “$P \lor Q$ widget” is a sum containing a “$P$ widget” or a “$Q$ widget”
- A “$P \supset Q$ widget” is a (computable) function from “$P$ widgets” to “$Q$ widgets.”

This was the original motivation for intuitionism and Kleene was the first to put it in the lambda calculus. Why Curry and Howard shouldn’t get credit.

But, BHK is about semantics while Curry-Howard is about syntax.
Other Types

All propositions are types.
Not all types are propositions: what is the logical content of \texttt{nat}?

\[
\begin{align*}
\Gamma \vdash e : \text{nat} \\
\Gamma \vdash Z : \text{nat} & \quad \Gamma \vdash S(e) : \text{nat} \\
\Gamma \vdash e_1 : T & \quad \Gamma, x : T \vdash e_2 : T \Rightarrow \\
& \quad \Gamma \vdash \text{rec } e \text{ with } Z.e_1; S(x).e_2 : T
\end{align*}
\]

not an isomorphism
Recursion

What is the logical interpretation of \texttt{fix}?

\[ \Gamma \vdash \text{fix} : (T \rightarrow T) \rightarrow T \]

Moral: consistency depends on termination.
Recursion

What is the logical interpretation of \texttt{fix}?

\[
\Gamma \vdash \texttt{fix} : (T \rightarrow T) \rightarrow T
\]

Inconsistent

\[
\begin{align*}
\Gamma, x : T & \vdash x : T \\
\Gamma & \vdash \lambda x^T.x : T \rightarrow T \\
\Gamma & \vdash \texttt{fix} : (T \rightarrow T) \rightarrow T \\
\Gamma & \vdash \texttt{fix} (\lambda x^T.x) : T
\end{align*}
\]

Lets us prove anything!
Moral: consistency depends on termination.
Curry-Howard Generalized

We have seen a correspondence between typed functional programming and intuitionistic propositional logic. But this isn’t the only correspondence. Every (well behaved) logic corresponds to a programming language.

- First order logic $=$ (limited) dependent types
- Second order logic $=$ polymorphism
Curry-Howard Goes Classical

Even classical logic has a Curry-Howard interpretation!
Curry-Howard Goes Classical

Even classical logic has a Curry-Howard interpretation!
And it involves goto
Curry-Howard Goes Classical

Even classical logic has a Curry-Howard interpretation!
And it involves goto but the good kind of goto (functional goto/goto with argument)
Curry-Howard Goes Classical

Even classical logic has a Curry-Howard interpretation!
And it involves goto but the good kind of goto (functional
goto/goto with argument)
Program of type $P + (P \rightarrow Q)$:

```
local label top becomes \( B(\lambda x. \text{goto} \ \text{top} \ \text{with} \ A(x)) \)
```
Curry-Howard Goes Classical

Even classical logic has a Curry-Howard interpretation!
And it involves goto but the good kind of goto (functional goto/goto with argument)
Program of type $P + (P \rightarrow Q)$:

$$\text{local label top becomes } B(\lambda x.\text{goto top with } A(x))$$

Many more logics beyond this relation not necessarily 1-to-1

- Modal logic S4 = runtime code generation
- Modal logic S5 = distributed systems with “mobile” code
- Linear logic = state/IO
- Linear logic = concurrency (session types)
- Temporal logic = “reactive” GUI programming
- etc
Conclusion

Math, Philosophy, and Computer Science are all studying the same thing.
Curry-Howard is a dictionary for translating between fields

- Sublime Facts: Modus Ponens is Function Application
- Confidence we have done things right: they were invented twice (or thrice)
- Talk to people from other fields
- Research approach:
  1. find idea in logic
  2. translate to CS
  3. profit
Acknowledgments

Some of this material came from Boyana’s original slides. Photos of Howard http://www.math.uic.edu/mugshots/wahow and Curry https://wiki.haskell.org/wikiupload/8/86/HaskellBCurry.jpg