Curry-Howard From The Ground Up

Michael Arntzenius

July 31, 2014

What's this talk about?

1. A simple formal logic (natural deduction)
2. A simple programming language (λ-calculus)
3. How they're *secretly the same thing!!*
4. Where is this relevant?

What's formal logic?

Inference rules

$$\frac{\mathit{premise}_1 \quad \mathit{premise}_2 \quad \cdots \quad \mathit{premise}_n}{\mathit{conclusion}}\ \texttt{RuleName}$$

If *premise*$_1$, *premise*$_2$, ..., and *premise*$_n$,
then *conclusion*,
by rule `RuleName`.

Defining inference rules

$$\frac{a < b \quad b < c}{a < c}\ \texttt{transitivity}$$

This **defines** a rule of inference called `transitivity`.

Using inference rules to prove things

$$\frac{a < b \quad b < c}{a < c}\ \texttt{transitivity}$$

Suppose we know that $a < b$, $b < c$, and $c < d$. We'd like to **prove** that $a < d$ using our `transitivity` rule.

Using inference rules to prove things

$$\frac{a < b \quad b < c}{a < c}\ \texttt{transitivity}$$

Suppose we know that $a < b$, $b < c$, and $c < d$. We'd like to **prove** that $a < d$ using our `transitivity` rule.

$$\dfrac{\overset{\textit{(assumed)}}{a < b} \qquad \dfrac{\overset{\textit{(assumed)}}{b < c} \quad \overset{\textit{(assumed)}}{c < d}}{b < d}\ \texttt{transitivity}}{a < d}\ \texttt{transitivity}$$

A simple logic

Connective	Meaning
$A \wedge B$	"A and B"
$A \supset B$	"A implies B", or "given A then B"
$A \vee B$	"A or B (*and I know which one*)"
$\neg A$	"not A" or "refutation of A"

A simple logic: $\wedge$

$$\frac{A \quad B}{A \wedge B}\wedge\mathtt{I}$$

A simple logic: $\wedge$

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathtt{I} \qquad \frac{A \wedge B}{A}\,\wedge\mathtt{E}_1$$

A simple logic: $\wedge$

$$\frac{A \quad B}{A \wedge B} \wedge\mathtt{I} \qquad \frac{A \wedge B}{A} \wedge\mathtt{E}_1 \qquad \frac{A \wedge B}{B} \wedge\mathtt{E}_2$$

A simple logic: $\wedge$: example proofs

For reference:

$$\frac{A \quad B}{A \wedge B} \wedge\mathtt{I} \qquad \frac{A \wedge B}{A} \wedge\mathtt{E}_1 \qquad \frac{A \wedge B}{B} \wedge\mathtt{E}_2$$

A simple logic: $\wedge$: example proofs

$$\dfrac{\dfrac{\overset{\textit{(assumed)}}{A \wedge (B \wedge C)}}{B \wedge C}\,\wedge\mathtt{E}_2}{B}\,\wedge\mathtt{E}_1$$

For reference:

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathtt{I} \qquad \frac{A \wedge B}{A}\,\wedge\mathtt{E}_1 \qquad \frac{A \wedge B}{B}\,\wedge\mathtt{E}_2$$

A simple logic: $\wedge$: example proofs

$$\cfrac{\cfrac{\overset{\textit{(assumed)}}{A \wedge (B \wedge C)}}{B \wedge C}\,\wedge\mathtt{E}_2}{B}\,\wedge\mathtt{E}_1 \qquad \cfrac{\overset{\textit{(assumed)}}{A} \qquad \overset{\textit{(assumed)}}{A}}{A \wedge A}\,\wedge\mathtt{I}$$

For reference:

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathtt{I} \qquad \frac{A \wedge B}{A}\,\wedge\mathtt{E}_1 \qquad \frac{A \wedge B}{B}\,\wedge\mathtt{E}_2$$

A simple logic: $\wedge$: example proofs

$$\dfrac{\dfrac{\overset{(assumed)}{A \wedge (B \wedge C)}}{B \wedge C}\,\wedge\mathtt{E}_2}{B}\,\wedge\mathtt{E}_1 \qquad \dfrac{\overset{(assumed)}{A} \qquad \overset{(assumed)}{A}}{A \wedge A}\,\wedge\mathtt{I} \qquad \dfrac{\dfrac{\overset{(assumed)}{A \wedge B}}{B}\,\wedge\mathtt{E}_2 \qquad \overset{(assumed)}{C}}{B \wedge C}\,\wedge\mathtt{I}$$

For reference:

$$\dfrac{A \quad B}{A \wedge B}\,\wedge\mathtt{I} \qquad \dfrac{A \wedge B}{A}\,\wedge\mathtt{E}_1 \qquad \dfrac{A \wedge B}{B}\,\wedge\mathtt{E}_2$$

A simple logic: $\supset$

$$\frac{???}{A \supset B} \supset\texttt{I} \qquad \frac{A \supset B \quad A}{B} \supset\texttt{E}$$

A simple logic: $\supset$: example proof (1)

Let's try to prove C given A, B, and $(A \wedge B) \supset C$:

For reference:

$$\frac{???}{A \supset B} \supset \mathtt{I} \qquad \frac{A \supset B \quad A}{B} \supset \mathtt{E}$$

A simple logic: $\supset$: example proof (1)

Let's try to prove C given A, B, and $(A \wedge B) \supset C$:

$$\dfrac{\overset{\textit{(assumed)}}{(A \wedge B) \supset C} \qquad \dfrac{\overset{\textit{(assumed)}}{A} \qquad \overset{\textit{(assumed)}}{B}}{A \wedge B}\,\wedge\mathtt{I}}{C}\,\supset\mathtt{E}$$

For reference:

$$\dfrac{???}{A \supset B}\,\supset\mathtt{I} \qquad\qquad \dfrac{A \supset B \quad A}{B}\,\supset\mathtt{E}$$

A simple logic: $\supset$

$$\frac{???}{A \supset B} \supset \mathtt{I} \qquad \frac{A \supset B \quad A}{B} \supset \mathtt{E}$$

A simple logic: $\supset$

$$\dfrac{\begin{array}{c} A \\ \vdots \\ B \end{array}}{A \supset B} \supset\texttt{I} \qquad\qquad \dfrac{A \supset B \quad A}{B} \supset\texttt{E}$$

A simple logic: $\supset$

$$
\dfrac{\begin{array}{c} \overset{x}{\overline{A}} \\ \vdots \\ B \end{array}}{A \supset B} \supset\mathtt{I}_x
\qquad\qquad
\dfrac{A \supset B \quad A}{B} \supset\mathtt{E}
$$

A simple logic: $\supset$: example proof (2)

Let's try to prove, given $A \supset B$, that $A \supset (A \wedge B)$:

For reference:

$$\dfrac{\begin{matrix}\overset{\text{-----}\,x}{A} \\ \vdots \\ B\end{matrix}}{A \supset B} \supset \mathtt{I}_x \qquad \dfrac{A \supset B \quad A}{B} \supset \mathtt{E}$$

A simple logic: $\supset$: example proof (2)

Let's try to prove, given $A \supset B$, that $A \supset (A \wedge B)$:

$$\cfrac{\cfrac{\overset{\text{- - - - -}\,x}{A} \qquad \cfrac{\overset{(\textit{assumed})}{A \supset B} \qquad \overset{\text{- - - - -}\,x}{A}}{B}\ \supset\mathtt{E}}{A \wedge B}\ \wedge\mathtt{I}}{A \supset (A \wedge B)}\ \supset\mathtt{I}_x$$

For reference:

$$\cfrac{\begin{array}{c}\overset{\text{- - - - -}\,x}{A} \\ \vdots \\ B\end{array}}{A \supset B}\ \supset\mathtt{I}_x \qquad\qquad \cfrac{A \supset B \qquad A}{B}\ \supset\mathtt{E}$$

A simple logic: $\supset$: example proof (3)

What's wrong with this "proof" of $(A \supset A) \wedge A$?

For reference:

$$\dfrac{\begin{matrix} \overline{A}^{\,x} \\ \vdots \\ B \end{matrix}}{A \supset B} \supset \mathtt{I}_x \qquad \dfrac{A \supset B \quad A}{B} \supset \mathtt{E}$$

A simple logic: $\supset$: example proof (3)

What's wrong with this "proof" of $(A \supset A) \wedge A$?

$$\dfrac{\dfrac{\overline{A}^{\,x}}{A \supset A}\supset\mathtt{I}_x \qquad \overline{A}^{\,x}}{(A \supset A) \wedge A}\wedge\mathtt{I}$$

For reference:

$$\dfrac{\begin{matrix}\overline{A}^{\,x}\\ \vdots \\ B\end{matrix}}{A \supset B}\supset\mathtt{I}_x \qquad \dfrac{A \supset B \quad A}{B}\supset\mathtt{E}$$

A simple logic: $\supset$: example proof (4)

Let's try to prove $(A \wedge B) \supset C$ from $A \supset (B \supset C)$:

For reference:

$$\dfrac{\begin{array}{c}\overset{\text{-----}\,x}{A} \\ \vdots \\ B\end{array}}{A \supset B}\supset\mathtt{I}_x \qquad \dfrac{A \supset B \quad A}{B}\supset\mathtt{E}$$

A simple logic: $\supset$: example proof (4)

Let's try to prove $(A \wedge B) \supset C$ from $A \supset (B \supset C)$:

$$\cfrac{\cfrac{\cfrac{\overset{\textit{(assumed)}}{A \supset (B \supset C)} \qquad \cfrac{\overline{A \wedge B}^{\,x}}{A}\,\wedge\mathtt{E}_1}{B \supset C}\,\supset\mathtt{E} \qquad \cfrac{\overline{A \wedge B}^{\,x}}{B}\,\wedge\mathtt{E}_2}{C}\,\supset\mathtt{E}}{(A \wedge B) \supset C}\,\supset\mathtt{I}_x$$

For reference:

$$\cfrac{\begin{matrix}\overline{A}^{\,x} \\ \vdots \\ B\end{matrix}}{A \supset B}\,\supset\mathtt{I}_x \qquad\qquad \cfrac{A \supset B \quad A}{B}\,\supset\mathtt{E}$$

A simple logic (all together now)

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathtt{I} \qquad \frac{A \wedge B}{A}\,\wedge\mathtt{E}_1 \qquad \frac{A \wedge B}{B}\,\wedge\mathtt{E}_2$$

$$\frac{\begin{array}{c}\overset{\text{-----}x}{A}\\ \vdots \\ B\end{array}}{A \supset B}\,\supset\mathtt{I}_x \qquad \frac{A \supset B \quad A}{B}\,\supset\mathtt{E}$$

A simple programming language (λ-calculus)

A simple λ-calculus

Syntax	What is it?	in Python
$\lambda x.a$	Anonymous function	`lambda x: a`
$a\ b$	Function application	`a(b)`
$\langle a, b \rangle$	Make a pair	`(a,b)`
$\pi_1\ a$	First element of pair	`a[0]`
$\pi_2\ a$	Second element of pair	`a[1]`

What does $\pi_1\,(\lambda x.x)$ do?

In Python: `(lambda x: x)[0]`

A simple λ-calculus: types

Type	What is it?	in Haskell
$A \to B$	Functions from A to B	`A -> B`
$A \times B$	Pairs of As and Bs	`(A,B)`

How do I know what type an expression has?

How do I know what type an expression has?

Maybe we can we use inference rules!

A simple λ-calculus: typing rules

A simple λ-calculus: typing rules

$$\frac{a : A \quad b : B}{\langle a, b \rangle : A \times B}\ \texttt{pair}$$

A simple λ-calculus: typing rules

$$\frac{a : A \quad b : B}{\langle a, b \rangle : A \times B}\ \texttt{pair} \qquad \frac{a : A \times B}{\pi_1\, a : A}\ \texttt{proj}_1 \qquad \frac{a : A \times B}{\pi_2\, a : B}\ \texttt{proj}_2$$

A simple λ-calculus: typing rules

$$\frac{a : A \quad b : B}{\langle a, b \rangle : A \times B}\ \texttt{pair} \qquad \frac{a : A \times B}{\pi_1\, a : A}\ \texttt{proj}_1 \qquad \frac{a : A \times B}{\pi_2\, a : B}\ \texttt{proj}_2$$

$$\frac{f : A \to B \quad a : A}{f\, a : B}\ \texttt{app}$$

A simple λ-calculus: typing rules

$$\frac{a : A \quad b : B}{\langle a, b \rangle : A \times B}\ \texttt{pair} \qquad \frac{a : A \times B}{\pi_1\, a : A}\ \texttt{proj}_1 \qquad \frac{a : A \times B}{\pi_2\, a : B}\ \texttt{proj}_2$$

$$\frac{f : A \to B \quad a : A}{f\, a : B}\ \texttt{app} \qquad \frac{\begin{array}{c} x : A \\ \vdots \\ b : B \end{array}}{\lambda x.b : A \to B}\ \texttt{lam}$$

A simple λ-calculus: typing rules

$$\frac{A \qquad B}{A \times B}\ \texttt{pair} \qquad \frac{A \times B}{A}\ \texttt{proj}_1 \qquad \frac{A \times B}{B}\ \texttt{proj}_2$$

$$\frac{A \to B \qquad A}{B}\ \texttt{app} \qquad \frac{\begin{array}{c} A \\ \vdots \\ B \end{array}}{A \to B}\ \texttt{lam}$$

A simple λ-calculus: typing rules

$$\frac{A \qquad B}{A \wedge B} \wedge \mathtt{I} \qquad \frac{A \wedge B}{A} \wedge \mathtt{E}_1 \qquad \frac{A \wedge B}{B} \wedge \mathtt{E}_2$$

$$\frac{A \supset B \qquad A}{B} \supset \mathtt{E} \qquad \frac{\begin{array}{c} A \\ \vdots \\ B \end{array}}{A \supset B} \supset \mathtt{I}$$

Our programming language was a logic in disguise!

Our programming language was a logic in disguise!

... are expressions proofs in disguise?

Curry-Howard: programs as proofs

Here's a simple proof:

$$\dfrac{\dfrac{\overset{\textit{(assumed)}}{A \land B}}{B}\ \land\mathtt{E}_2 \qquad \dfrac{\overset{\textit{(assumed)}}{A \land B}}{A}\ \land\mathtt{E}_1}{B \land A}\ \land\mathtt{I}$$

This shows that $\land$ is commutative.

Curry-Howard: programs as proofs

Here's a simple ~~proof~~ program:

$$\dfrac{\dfrac{a : A \times B}{\pi_2\, a : B}\,\texttt{proj}_2 \qquad \dfrac{a : A \times B}{\pi_1\, a : A}\,\texttt{proj}_1}{\langle \pi_2\, a, \pi_1\, a \rangle : B \times A}\,\texttt{pair}$$

In Python: `(a[1], a[0])`. This swaps a pair.

Commutativity of $\wedge$

=

Swapping a pair!

What have we just learned?

propositions	=	types
proofs	=	expressions
"and", $\wedge$	=	pairs, $\times$
"implies", $\supset$	=	functions, $\rightarrow$
rules of logic	=	rules of typing
commutativity of $\wedge$	=	swapping a pair

We have found a bridge between logic and programming languages.

What use is this?

- Prove theorems by writing code: Coq, Agda, Idris, ...
- Prove theorems **about** programming:
  can apply a hundred years of work in formal logic!
- Serendipity: Take a thing in {logic,PL},
  ask "what does this mean in {PL,logic}?"
- Design of programming language features

Serendipity

What logical concepts have meaning in programming-land?

- Connectives: $\vee$, $\neg$, $\forall$, $\exists$, ...
- Properties: Constructivity, consistency, ...
- Systems: Modal logic, linear logic, ...

What PL concepts have meaning in logic-land?

- **Evaluation**, Turing-completeness, laziness, mutation, subtyping, exceptions, monads, ...

BONUS ROUND: Evaluating our λ-calculus

Read "$a \mapsto b$" as "a steps to b" or "whenever you see a, you can replace it with b".

Rule	in Python
$\pi_1 \langle a, b \rangle \mapsto a$	`(a,b)[0]` evaluates to `a`
$\pi_2 \langle a, b \rangle \mapsto b$	`(a,b)[1]` evaluates to `b`

BONUS ROUND: Evaluating our λ-calculus

Read "$a \mapsto b$" as "a steps to b" or "whenever you see a, you can replace it with b".

Rule	in Python
$\pi_1 \langle a, b \rangle \mapsto a$	`(a,b)[0]` evaluates to `a`
$\pi_2 \langle a, b \rangle \mapsto b$	`(a,b)[1]` evaluates to `b`
$(\lambda x.a)\, b \mapsto [b/x]\, a$	

$[b/x]\, a$ means "substitute b for x in a".

BONUS ROUND: Evaluating our λ-calculus

Read "$a \mapsto b$" as "a steps to b" or "whenever you see a, you can replace it with b".

Rule	in Python
$\pi_1 \langle a, b \rangle \mapsto a$	`(a,b)[0]` evaluates to `a`
$\pi_2 \langle a, b \rangle \mapsto b$	`(a,b)[1]` evaluates to `b`
$(\lambda x.a)\, b \mapsto [b/x]\, a$	`(lambda x: a)(b)` evaluates to

$[b/x]\, a$ means "substitute b for x in a".

BONUS ROUND: Evaluating our λ-calculus

Read "$a \mapsto b$" as "a steps to b" or "whenever you see a, you can replace it with b".

Rule	in Python
$\pi_1 \langle a, b \rangle \mapsto a$	`(a,b)[0]` evaluates to `a`
$\pi_2 \langle a, b \rangle \mapsto b$	`(a,b)[1]` evaluates to `b`
$(\lambda x.a)\, b \mapsto [b/x]\, a$	`(lambda x: a)(b)` evaluates to
	... well, it's complicated

$[b/x]\, a$ means "substitute b for x in a".