

A Programming Language Based on Classical Logic

William Lovas
(with Karl Crary)

Motivation



“It is very, *very*
easy to design *bad*
programming
languages.”

(John Reynolds)

- ▶ Want to design *good* programming languages by building on *logical foundations*
- ▶ Today: explore one possibility, *classical logic*

Part 1: Proof theory boot camp

Logical foundations

- ▶ Curry-Howard correspondence

Logic	Programming
Propositions	Types
Proofs	Programs
Proof-checking	Type-checking
Simplification	Evaluation

- ▶ Classical logic as a programming language?
 - excluded middle, proof by contradiction, ...

Curry-Howard correspondence

- ▶ Propositions as types, proofs as programs

Curry-Howard correspondence

- ▶ Propositions as types, proofs as programs
- ▶ **Q:** What is a proof of a proposition?

Curry-Howard correspondence

- ▶ Propositions as types, proofs as programs
- ▶ **Q:** What is a proof of a proposition?
- ▶ **Q:** What is a proposition?

Curry-Howard correspondence

- ▶ Propositions as types, proofs as programs
- ▶ **Q:** What is a proof of a proposition?
- ▶ **Q:** What is a proposition?
- ▶ **A:** Something that can be judged true.
 - e.g. “ $2 + 3 = 6$ ”, or “it is raining”

Example

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .

Example

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ Conjunction (“and”)

Example

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ Conjunction (“and”)
- ▶ Disjunction (“or”)

Example

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ Conjunction (“and”)
- ▶ Disjunction (“or”)
- ▶ Implication (“if ... then ...”)

Example

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ Conjunction (“and”)
- ▶ Disjunction (“or”)
- ▶ Implication (“if ... then ...”)
- ▶ A , B , and C : whatever you like...

Example

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ Conjunction (“and”)
- ▶ Disjunction (“or”)
- ▶ Implication (“if ... then ...”)
- ▶ A , B , and C : whatever you like...
- ▶ Symbolically: $A \wedge (B \vee C) \Rightarrow (A \wedge B) \vee (A \wedge C)$

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose C :
 - have A (since A and [...])
 - have C (by assumption)
 - thus A and C (since we have both)
 - thus either A and B , or A and C (in particular, the second)

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - ...
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - ...
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C

Example Proof

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - ...
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
 - thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C
 - have B or C (since [...] a
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

$$\frac{A \text{ true} \quad B \text{ true}}{A \wedge B \text{ true}} \wedge\text{-I}$$

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and $(B \text{ or } C)$)
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and $(B \text{ or } C)$)
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and $(B \text{ or } C)$)
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and $(B \text{ or } C)$)
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

$$\begin{array}{c}
 \frac{A \wedge B \text{ true}}{A \text{ true}} \wedge\text{-E}_1 \qquad \frac{A \wedge B \text{ true}}{B \text{ true}} \wedge\text{-E}_2
 \end{array}$$

Formalizing Proof, take 1

- ▶ Judgement: $A \text{ true}$. (“ A is provable.”)
- ▶ Inference rules: grouped into “Introductions”:

$$\frac{A \text{ true} \quad B \text{ true}}{A \wedge B \text{ true}} \wedge\text{-I}$$

- ▶ ... and “Eliminations”:

$$\frac{A \wedge B \text{ true}}{A \text{ true}} \wedge\text{-E}_1 \quad \frac{A \wedge B \text{ true}}{B \text{ true}} \wedge\text{-E}_2$$

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and (B or C))
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and $(B \text{ or } C)$)
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

- Suppose A and either B or C :
 - have B or C (since [...] and $(B \text{ or } C)$)
 - suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
 - suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
 - thus either A and B , or A and C
- thus, if A and either B or C , then either A and B , or A and C

A closer look...

▶ Proof:

◦ Suppose A and either B or C :

- have B or C (since [...] and $(B \text{ or } C)$)
- suppose B :
 - have A (since A and [...])
 - have B (by assumption)
 - thus A and B (since we have both)
 - thus either A and B , or A and C (in particular, the first)
- suppose C :
 - ...
 - thus either A and B , or A and C (in particular, the second)
- thus either A and B , or A and C

◦ thus, if A and either B or C , then either A and B , or A and C

A closer look...

► Proof:

◦ Suppose *A* and either *B* or *C*:

- have *B* or *C* (since [...] and [...])
- suppose *B*:
 - have *A* (since *A* and [...])
 - have *B* (by assumption)
 - thus *A* and *B* (since we have both)
 - thus either *A* and *B*, or *A* and *C* (in particular, the first)
- suppose *C*:
 - ...
 - thus either *A* and *B*, or *A* and *C* (in particular, the second)
- thus either *A* and *B*, or *A* and *C*

$$\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \Rightarrow B \text{ true}} \Rightarrow \text{-I}$$

◦ thus, if *A* and either *B* or *C*, then either *A* and *B*, or *A* and *C*

Formalizing Proof, take 2

- ▶ Judgement: $\Gamma \vdash A \text{ true}$. (“ A is provable assuming Γ ”)
- ▶ Γ is a list of assumptions: $A_1 \text{ true}, \dots, A_n \text{ true}$
- ▶ Implication: one introduction rule:

$$\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \Rightarrow B \text{ true}} \Rightarrow\text{-I}$$

Formalizing Proof, take 2

- ▶ Judgement: $\Gamma \vdash A \text{ true}$. (“ A is provable assuming Γ ”)
- ▶ Γ is a list of assumptions: $A_1 \text{ true}, \dots, A_n \text{ true}$
- ▶ Implication: one introduction rule:

$$\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \Rightarrow B \text{ true}} \Rightarrow\text{-I}$$

- ▶ ... and one elimination rule:

$$\frac{\Gamma \vdash A \Rightarrow B \text{ true} \quad \Gamma \vdash A \text{ true}}{\Gamma \vdash B \text{ true}} \Rightarrow\text{-E}$$

Reasoning from assumptions

- ▶ Hypothesis rule:

$$\Gamma, A \text{ true} \vdash A \text{ true}$$

Reasoning from assumptions

- ▶ Hypothesis rule:

$$\frac{}{\Gamma, A \text{ true} \vdash A \text{ true}}$$

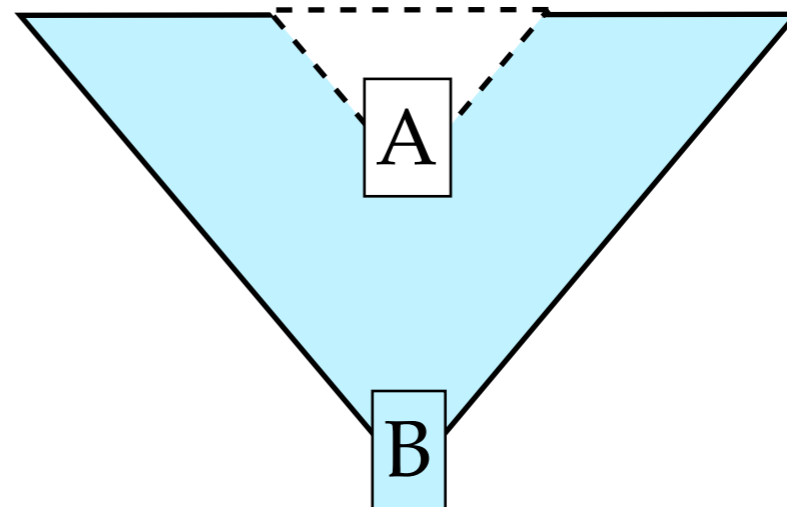
- ▶ **Substitution Principle:** if $\Gamma, A \text{ true} \vdash B \text{ true}$ and $\Gamma \vdash A \text{ true}$, then $\Gamma \vdash B \text{ true}$

Reasoning from assumptions

- ▶ Hypothesis rule:

$$\frac{}{\Gamma, A \text{ true} \vdash A \text{ true}}$$

- ▶ **Substitution Principle:** if $\Gamma, A \text{ true} \vdash B \text{ true}$ and $\Gamma \vdash A \text{ true}$, then $\Gamma \vdash B \text{ true}$

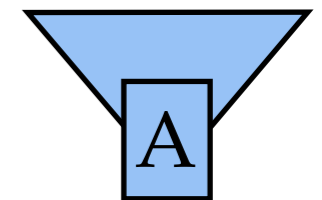
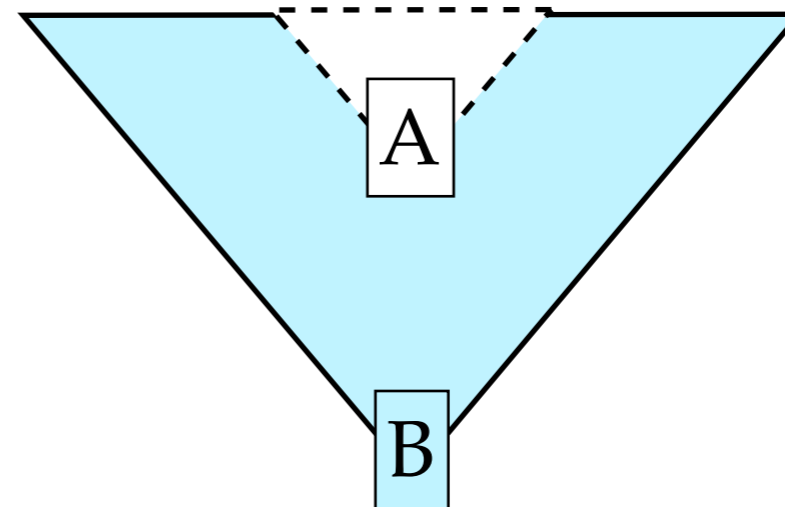


Reasoning from assumptions

- ▶ Hypothesis rule:

$$\frac{}{\Gamma, A \text{ true} \vdash A \text{ true}}$$

- ▶ **Substitution Principle:** if $\Gamma, A \text{ true} \vdash B \text{ true}$ and $\Gamma \vdash A \text{ true}$, then $\Gamma \vdash B \text{ true}$

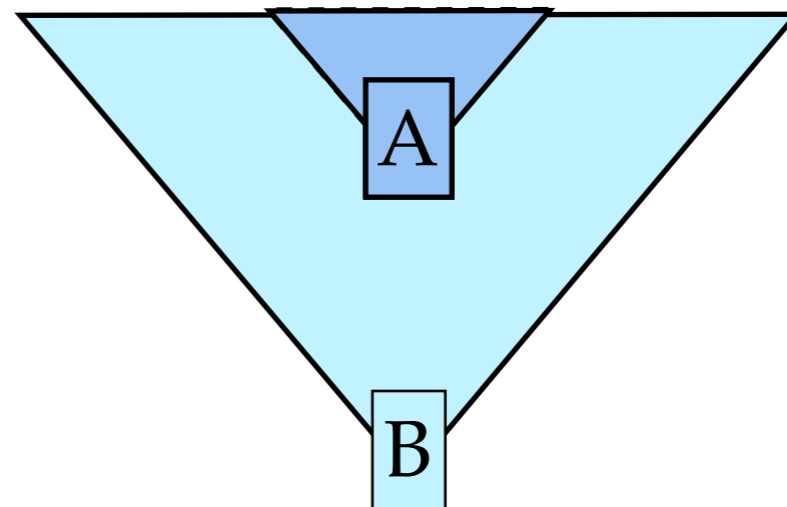


Reasoning from assumptions

- ▶ Hypothesis rule:

$$\frac{}{\Gamma, A \text{ true} \vdash A \text{ true}}$$

- ▶ **Substitution Principle:** if $\Gamma, A \text{ true} \vdash B \text{ true}$ and $\Gamma \vdash A \text{ true}$, then $\Gamma \vdash B \text{ true}$



Formalizing Proof, take 3

- ▶ Disjunction: two introduction rules:

$$\frac{\Gamma \vdash A \text{ true}}{\Gamma \vdash A \vee B \text{ true}} \vee\text{-I}_1 \qquad \frac{\Gamma \vdash B \text{ true}}{\Gamma \vdash A \vee B \text{ true}} \vee\text{-I}_2$$

Formalizing Proof, take 3

- ▶ Disjunction: two introduction rules:

$$\frac{\Gamma \vdash A \text{ true}}{\Gamma \vdash A \vee B \text{ true}} \vee\text{-I}_1 \quad \frac{\Gamma \vdash B \text{ true}}{\Gamma \vdash A \vee B \text{ true}} \vee\text{-I}_2$$

- ▶ ... and one elimination rule:

$$\frac{\Gamma \vdash A \vee B \text{ true} \quad \Gamma, A \text{ true} \vdash C \text{ true} \quad \Gamma, B \text{ true} \vdash C \text{ true}}{\Gamma \vdash C \text{ true}} \vee\text{-E}$$

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:
- ▶ Suppose A true:

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:
- ▶ Suppose A true:
 - Hmm... tricky...

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:
- ▶ Suppose A true:
 - Hmm... tricky...
 - Well, we also have B true...

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:
- ▶ Suppose A true:
 - Hmm... tricky...
 - Well, we also have B true...
 - A-ha! By \wedge -I, we have $A \wedge B$ true.

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:
- ▶ Suppose A true:
 - Hmm... tricky...
 - Well, we also have B true...
 - A-ha! By \wedge -I, we have $A \wedge B$ true.
 - And then by \wedge -E₁, we have A true.

Proof simplification

- ▶ Easy to make detours. Consider proving $A \Rightarrow A$:
- ▶ Suppose A true:
 - Hmm... tricky...
 - Well, we also have B true...
 - A-ha! By \wedge -I, we have $A \wedge B$ true.
 - And then by \wedge -E₁, we have A true.
 - *phew*

Proof simplification

- ▶ Eliminate “redundant” steps

$$\begin{array}{c} \mathcal{D} \qquad \mathcal{E} \\ \Gamma \vdash A \text{ true} \quad \Gamma \vdash B \text{ true} \\ \hline \Gamma \vdash A \wedge B \text{ true} \\ \hline \Gamma \vdash A \text{ true} \end{array}$$

Proof simplification

- ▶ Eliminate “redundant” steps

$$\frac{\frac{\mathcal{D} \quad \mathcal{E}}{\Gamma \vdash A \text{ true} \quad \Gamma \vdash B \text{ true}}}{\Gamma \vdash A \wedge B \text{ true}}}{\Gamma \vdash A \text{ true}} \quad \Rightarrow \quad \frac{\mathcal{D}}{\Gamma \vdash A \text{ true}}$$

Proof simplification

- ▶ Using Substitution Principle:

$$\frac{\frac{\mathcal{D}}{\Gamma, A \text{ true} \vdash B \text{ true}}}{\Gamma \vdash A \Rightarrow B \text{ true}} \quad \mathcal{E}}{\Gamma \vdash B \text{ true}} \Gamma \vdash A \text{ true}$$

Proof simplification

- ▶ Using Substitution Principle:

$$\frac{\frac{\mathcal{D}}{\Gamma, A \text{ true} \vdash B \text{ true}}}{\Gamma \vdash A \Rightarrow B \text{ true}} \quad \Gamma \vdash A \text{ true} \quad \mathcal{F}}{\Gamma \vdash B \text{ true}}$$



$$\begin{array}{l} \mathcal{F} \\ \Gamma \vdash A \text{ true} \\ : \\ \Gamma \vdash B \text{ true} \end{array}$$

Proof terms

$$\frac{\Gamma \vdash A \text{ true} \quad \Gamma \vdash B \text{ true}}{\Gamma \vdash A \wedge B \text{ true}} \quad \wedge\text{-I}$$

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash A \text{ true}} \quad \wedge\text{-E}_1$$

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash B \text{ true}} \quad \wedge\text{-E}_2$$

Proof terms

$$\frac{\Gamma \vdash e_1 : A \text{ true} \quad \Gamma \vdash e_2 : B \text{ true}}{\Gamma \vdash (e_1, e_2) : A \wedge B \text{ true}} \wedge\text{-I}$$

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash A \text{ true}} \wedge\text{-E}_1$$

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash B \text{ true}} \wedge\text{-E}_2$$

Proof terms

$$\frac{\Gamma \vdash e_1 : A \text{ true} \quad \Gamma \vdash e_2 : B \text{ true}}{\Gamma \vdash (e_1, e_2) : A \wedge B \text{ true}} \wedge\text{-I}$$

$$\frac{\Gamma \vdash e : A \wedge B \text{ true}}{\Gamma \vdash \#1 e : A \text{ true}} \wedge\text{-E}_1$$

$$\frac{\Gamma \vdash e : A \wedge B \text{ true}}{\Gamma \vdash \#2 e : B \text{ true}} \wedge\text{-E}_2$$

Proof terms

- ▶ Conjunction: pairing!

$$\frac{\Gamma \vdash e_1 : A \text{ true} \quad \Gamma \vdash e_2 : B \text{ true}}{\Gamma \vdash (e_1, e_2) : A \wedge B \text{ true}} \wedge\text{-I}$$

$$\frac{\Gamma \vdash e : A \wedge B \text{ true}}{\Gamma \vdash \#1 e : A \text{ true}} \wedge\text{-E}_1$$

$$\frac{\Gamma \vdash e : A \wedge B \text{ true}}{\Gamma \vdash \#2 e : B \text{ true}} \wedge\text{-E}_2$$

Proof terms

$$\frac{\Gamma, x : A \text{ true} \vdash e : B \text{ true}}{\Gamma \vdash \lambda x. e : A \Rightarrow B \text{ true}} \Rightarrow\text{-I}$$

$$\frac{\Gamma \vdash e_1 : A \Rightarrow B \text{ true} \quad \Gamma \vdash e_2 : A \text{ true}}{\Gamma \vdash e_1 e_2 : B \text{ true}} \Rightarrow\text{-E}$$

Proof terms

- ▶ Implication: functions!
- ▶ (Note: assumptions now labelled)

$$\frac{\Gamma, x : A \text{ true} \vdash e : B \text{ true}}{\Gamma \vdash \lambda x. e : A \Rightarrow B \text{ true}} \Rightarrow\text{-I}$$

$$\frac{\Gamma \vdash e_1 : A \Rightarrow B \text{ true} \quad \Gamma \vdash e_2 : A \text{ true}}{\Gamma \vdash e_1 e_2 : B \text{ true}} \Rightarrow\text{-E}$$

Proof terms

Proof terms

- ▶ Disjunction: datatypes and pattern matching!

Proof terms

- ▶ Disjunction: datatypes and pattern matching!

Proof terms

- ▶ Disjunction: datatypes and pattern matching!
- ▶ (rules elided)

Proof terms

- ▶ Simplification: evaluation!

$$\#1 (e_1, e_2) \longrightarrow e_1$$

$$\#2 (e_1, e_2) \longrightarrow e_2$$

$$(\lambda x. e_1) e_2 \longrightarrow [e_2 / x] e_1$$

- ▶ Basic programming language: the simply-type lambda calculus.
 - data structures, functions

Example Proof, revisited

Example Proof, revisited

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .

Example Proof, revisited

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**

Example Proof, revisited

▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .

▶ **Proof:**

◦ **fn** $x : A \wedge (B \vee C) \Rightarrow$

case #2 x **of** $\text{inl } y \Rightarrow \text{inl } (\#1 x, y)$

| $\text{inr } z \Rightarrow \text{inr } (\#1 x, z)$

Example Proof, revisited

- ▶ **Proposition:** If A and either B or C , then either A and B , or else A and C .
- ▶ **Proof:**
 - **fn** $x : A \wedge (B \vee C) \Rightarrow$
 case #2 x **of** $\text{inl } y \Rightarrow \text{inl } (\#1 x, y)$
 | $\text{inr } z \Rightarrow \text{inr } (\#1 x, z)$
- ▶ Computational content of proof: a simple input-shuffling program

Classical Logic

- ▶ What I've shown you: *intuitionistic logic*
- ▶ Classical logic: *proof-by-contradiction*

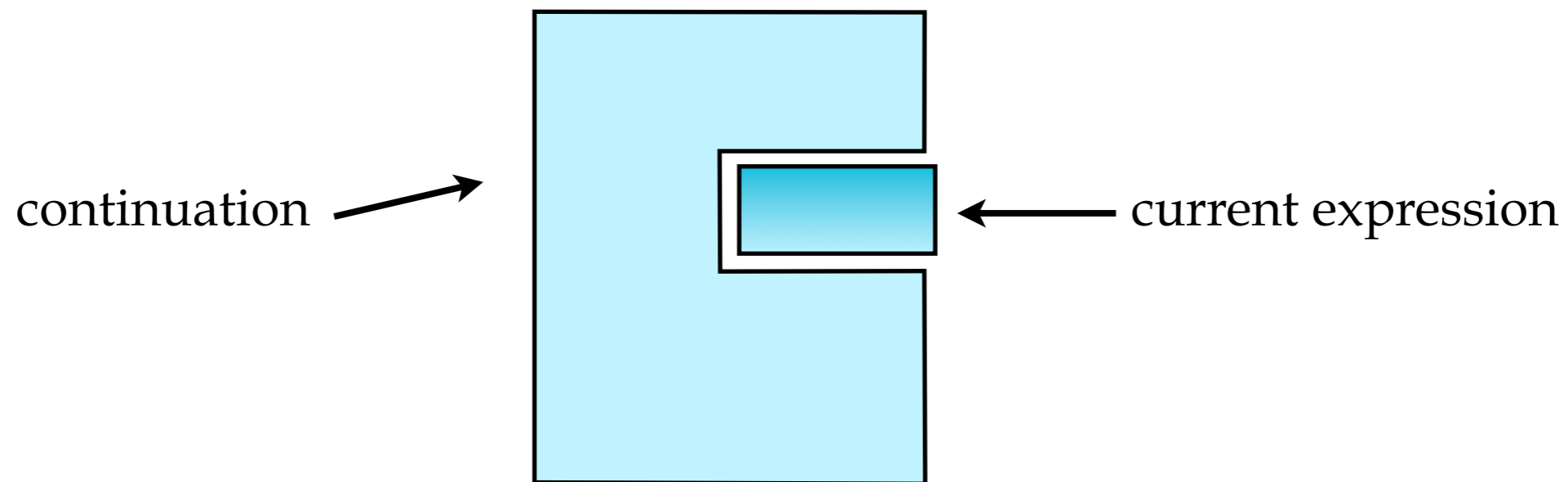
$$\frac{\Gamma, A \text{ false} \vdash \text{contra}}{\Gamma \vdash A \text{ true}}$$
$$\frac{\Gamma \vdash C \text{ true} \quad \Gamma \vdash C \text{ false}}{\Gamma \vdash \text{contra}}$$

- ▶ What is *contra*?
- ▶ What is *false*?
- ▶ Computational interpretation?

Continuations

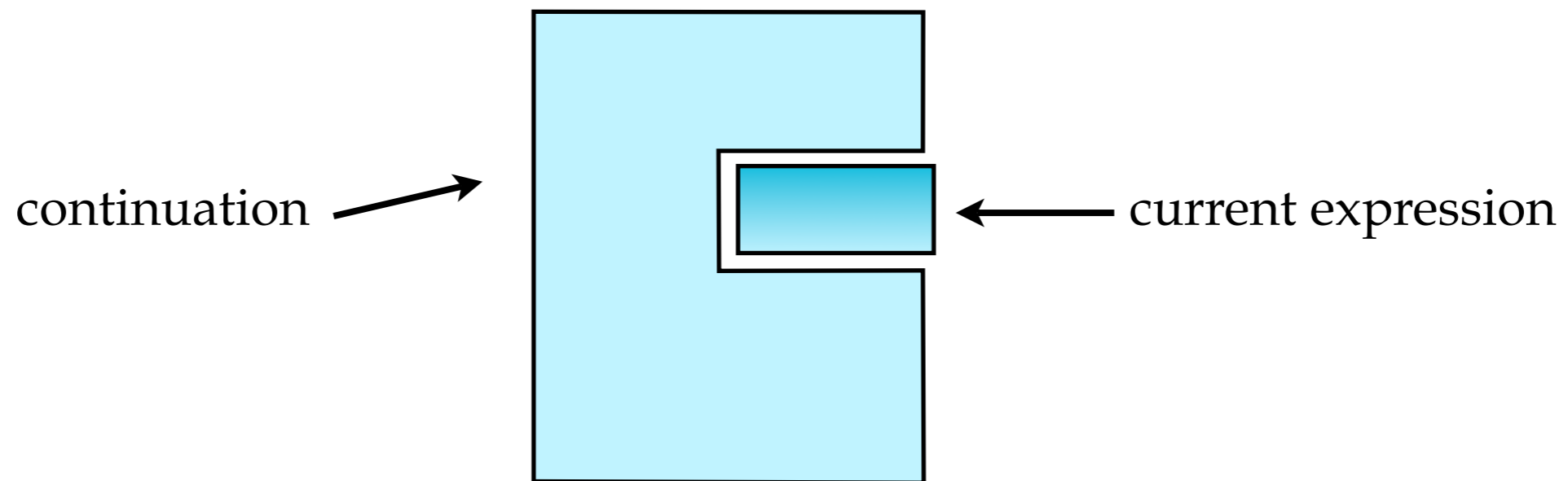
Continuations

- ▶ Intuition: separate a program into *what's happening now* and *what happens next*...
 - what's happening now: expression currently being evaluated
 - what happens next: the *continuation*: the rest of the program



Current continuation

- ▶ “**letcc** u in e ”: bind current continuation to u , run e
- ▶ “**throw** e to u ”: restore continuation u with expr. e
 - like a **goto** with an argument



Example: early exit

- ▶ `letcc` example: early exit

Example: early exit

- ▶ letcc example: early exit

```
fun product [] = 1  
  | product (x::xs) = x * product xs
```

Example: early exit

- ▶ letcc example: early exit

```
fun product [] = 1
  | product (0::_) = 0
  | product (x::xs) = x * product xs
```

Example: early exit

- ▶ **letcc** example: early exit

```
fun product nums =  
  letcc u in  
    let fun prod [] = 1  
      | prod (0::_) = throw 0 to u  
      | prod (x::xs) = x * prod xs  
    in  
      prod nums  
    end
```

What *are* continuations?

- ▶ Like a “partial program”: given a value of the right type, it becomes a complete program.
- ▶ “**A cont**”: type of a continuation expecting an *A*

What *are* continuations?

- ▶ Like a “partial program”: given a value of the right type, it becomes a complete program.
- ▶ “A **cont**”: type of a continuation expecting an A
- ▶ in “early exit” example:
 $u : \mathbf{int\ cont}$, since
“product” should
return an **int**

```
fun product nums =  
  letcc  $u$  in  
    let fun prod ...  
      in  
        prod nums  
      end
```

What can we *do* with them?

- ▶ Given an *A cont*, pass it an *A*

What can we *do* with them?

- ▶ Given an A **cont**, pass it an A
- ▶ Given an A **cont**, construct an $A \wedge B$ **cont**
 - accept a pair : $A \wedge B$
 - project the first component : A
 - pass it to original continuation

What can we *do* with them?

- ▶ Given an A **cont**, pass it an A
- ▶ Given an A **cont**, construct an $A \wedge B$ **cont**
 - accept a pair : $A \wedge B$
 - project the first component : A
 - pass it to original continuation

A **cont** $\Rightarrow A \wedge B$ **cont**

B **cont** $\Rightarrow A \wedge B$ **cont**

What can we *do* with them?

- ▶ Given an A **cont**, pass it an A
- ▶ Given an A **cont**, construct an $A \wedge B$ **cont**
 - accept a pair : $A \wedge B$
 - project the first component : A
 - pass it to original continuation
- ▶ Given an A **cont** and a B **cont**, make an $A \vee B$ **cont**
 - accept a sum : $A \vee B$
 - case analyze it to get either an A or a B
 - if A , pass to the A **cont**; if B , pass to the B **cont**

A **cont** \Rightarrow $A \wedge B$ **cont**

B **cont** \Rightarrow $A \wedge B$ **cont**

What can we *do* with them?

- ▶ Given an A **cont**, pass it an A
- ▶ Given an A **cont**, construct an $A \wedge B$ **cont**
 - accept a pair : $A \wedge B$
 - project the first component : A
 - pass it to original continuation
- ▶ Given an A **cont** and a B **cont**, make an $A \vee B$ **cont**
 - accept a sum : $A \vee B$
 - case analyze it to get either an A or a B
 - if A , pass to the A **cont**; if B , pass to the B **cont**

A **cont** \Rightarrow $A \wedge B$ **cont**

B **cont** \Rightarrow $A \wedge B$ **cont**

A **cont**, B **cont** \Rightarrow $A \vee B$ **cont**

What can we *do* with them?

- ▶ Given an A **cont**, pass it an A
- ▶ Given an A **cont**, construct an $A \wedge B$ **cont**
 - accept a pair : $A \wedge B$
 - project the first component : A
 - pass it to original continuation
- ▶ Given an A **cont** and a B **cont**, make an $A \vee B$ **cont**
 - accept a sum : $A \vee B$
 - case analyze it to get either an A or a B
 - if A , pass to the A **cont**; if B , pass to the B **cont**

$A \text{ false} \vdash A \wedge B \text{ false}$

$B \text{ false} \vdash A \wedge B \text{ false}$

$A \text{ false}, B \text{ false} \vdash A \vee B \text{ false}$

Continuations are refutations!

$$\frac{\Gamma \vdash k : A \text{ false}}{\Gamma \vdash (\#1; k) : A \wedge B \text{ false}} \wedge\text{-F}_1$$

$$\frac{\Gamma \vdash k : B \text{ false}}{\Gamma \vdash (\#2; k) : A \wedge B \text{ false}} \wedge\text{-F}_2$$

$$\frac{\Gamma \vdash k_1 : A \text{ false} \quad \Gamma \vdash k_2 : B \text{ false}}{\Gamma \vdash [k_1, k_2] : A \vee B \text{ false}} \vee\text{-F}$$

Classical Curry-Howard

- ▶ Expressions: $e : A \text{ true}$
- ▶ Continuations: $k : A \text{ false}$

Classical Curry-Howard

- ▶ Expressions: $e : A \text{ true}$
- ▶ Continuations: $k : A \text{ false}$
- ▶ Programs: $k \triangleleft e$

$$\frac{\Gamma \vdash e : C \text{ true} \quad \Gamma \vdash k : C \text{ false}}{\Gamma \vdash k \triangleleft e : \text{contra}}$$

Classical Curry-Howard

- ▶ Evaluate programs $k \triangleleft e$:

$\#1; k \triangleleft (e_1, e_2) \longrightarrow k \triangleleft e_1$

$\#2; k \triangleleft (e_1, e_2) \longrightarrow k \triangleleft e_2$

$[k_1, k_2] \triangleleft \text{inl } e \longrightarrow k_1 \triangleleft e$

$[k_1, k_2] \triangleleft \text{inr } e \longrightarrow k_2 \triangleleft e$

Proof-by-contradiction redux

$$\frac{\Gamma, A \text{ false} \vdash \text{contra}}{\Gamma \vdash A \text{ true}}$$

Proof-by-contradiction redux

$$\Gamma, u : A \text{ false} \vdash k \triangleleft e : \text{contra}$$

$$\Gamma \vdash \mathbf{letcc} \ u \ \mathbf{in} \ k \triangleleft e : A \ \text{true}$$

Proof-by-contradiction redux

$$\frac{\Gamma, u : A \text{ false} \vdash k \triangleleft e : \text{contra}}{\Gamma \vdash \text{letcc } u \text{ in } k \triangleleft e : A \text{ true}}$$

$k' \triangleleft \text{letcc } u \text{ in } k \triangleleft e \quad \longrightarrow \quad [k' / u] (k \triangleleft e)$

Classical proof terms

$\square e ::= x \mid \text{letcc } u:A \text{ false in } c)$	$\frac{A \text{ true}}{A \wedge B, \top}$
$(e_1, e_2) \mid ()$	$A \vee B$
$\text{inl}(e) \mid \text{inr}(e)$	$A \Rightarrow B$
$\lambda x:A. e$	$\neg A$
$\text{not}(k)$	
$\square k ::= u \mid \text{let } x:A \text{ true in } c$	$\frac{A \text{ false}}{A \wedge B}$
$\#1; k \mid \#2; k$	$A \vee B, \perp$
$[k_1, k_2] \mid []$	$A \Rightarrow B$
$e; k$	$\neg A$
$\text{not}(e)$	

Normalization

- ▶ **Theorem:** Every contradiction has a normal form.
 - “normal”: cannot reduce any further
- ▶ **Proof:** By nested induction on the *type* at which a contradiction occurs and the *terms* undergoing evaluation.

Normalization

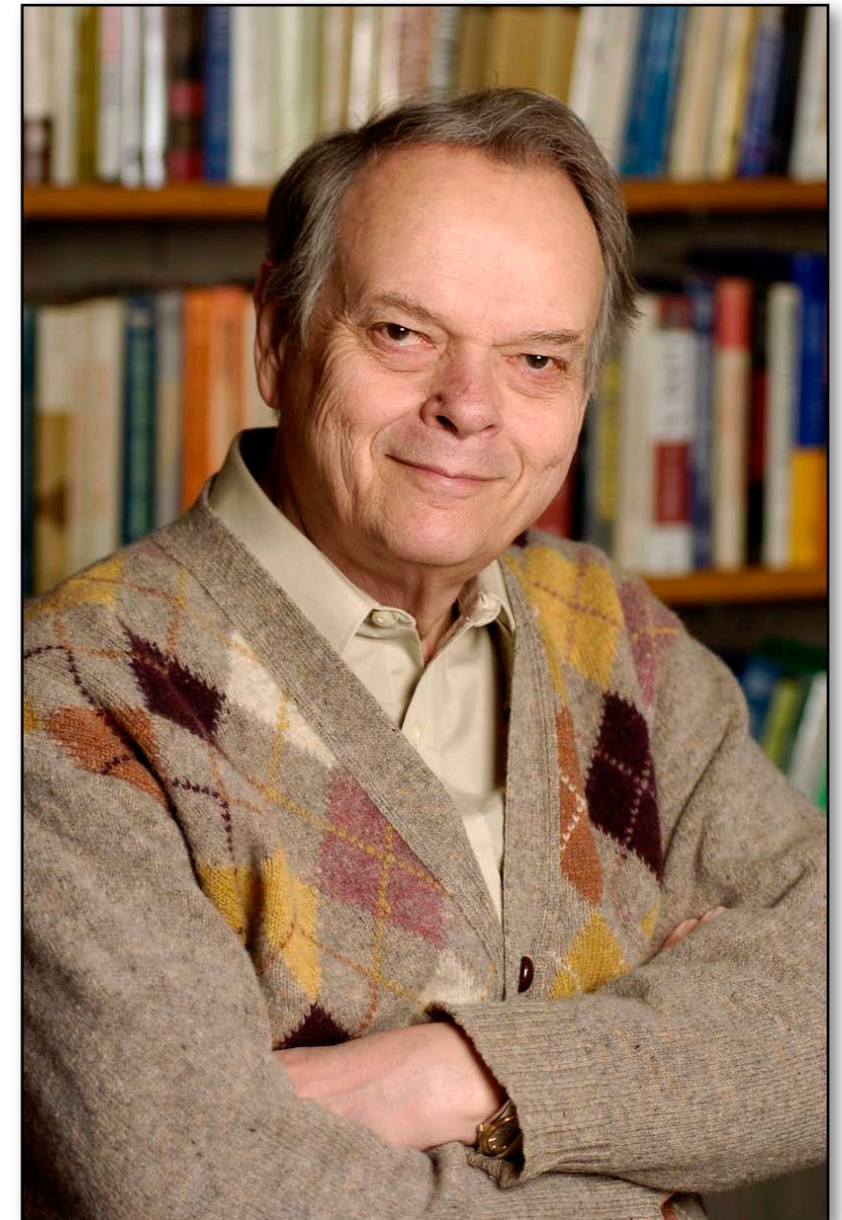
- ▶ **Theorem:** Every contradiction has a normal form.
 - “normal”: cannot reduce any further
- ▶ **Proof:** By nested induction on the *type* at which a contradiction occurs and the *terms* undergoing evaluation.
- ▶ **Corollary:** Classical logic is consistent, since there are no closed, normal contradictions

Prior work

- ▶ Standing on many giants' shoulders:
 - Andrzej Filinski
 - Michel Parigot
 - Timothy Griffin
 - Chetan Murthy
 - Pierre-Louis Curien and Hugo Herbelin
 - Aleksandar Nanevski
 - Philip Wadler
- ▶ But one of the first -- and simplest -- proofs of normalization.

Conclusion

- ▶ Observed that *continuations* embody *refutations* of propositions
- ▶ Constructed a *programming language* with *continuations*, based on *proof-by-contradiction*
- ▶ Proved the language *terminating*, establishing the *consistency* of classical logic



(John Reynolds approves)

To truth through proof

- ▶ *Q*: What is a proof of a proposition *A*?
- ▶ *A*: Depends on *A*...
- ▶ How about $A \wedge B$?
- ▶ A proof of $A \wedge B$ is a proof of *A* and a proof of *B*.

$$\frac{A \text{ true} \quad B \text{ true}}{A \wedge B \text{ true}}$$

To truth through proof

- ▶ *Q*: What can we *do* with a proof?
- ▶ *A*: From a proof of $A \wedge B$, we can get a proof of A .
(Also, a proof of B .)

$$A \wedge B \text{ true}$$

$$A \text{ true}$$
$$A \wedge B \text{ true}$$

$$B \text{ true}$$

Reasoning from hypotheses

- ▶ Refine judgement: *A true* becomes $\Gamma \vdash A \text{ true}$, with Γ an unordered list of hypotheses A_1, \dots, A_n
- ▶ Hypothesis rule:

$$\frac{}{\Gamma, A \text{ true} \vdash A \text{ true}}$$

- ▶ **Substitution Principle:** if $\Gamma, A \text{ true} \vdash B \text{ true}$ and $\Gamma \vdash A \text{ true}$, then $\Gamma \vdash B \text{ true}$

Intros and Elims

▶ Introduction

$$\frac{\Gamma \vdash A \text{ true} \quad \Gamma \vdash B \text{ true}}{\Gamma \vdash A \wedge B \text{ true}}$$

▶ Elimination

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash A \text{ true}}$$

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash B \text{ true}}$$

Proof simplification

- ▶ Eliminate “redundant” steps

$$\frac{\frac{\frac{\Gamma \vdash A \text{ true} \quad \Gamma \vdash B \text{ true}}{\Gamma \vdash A \wedge B \text{ true}}}{\Gamma \vdash A \text{ true}}}{\Gamma \vdash A \text{ true}} \quad \Rightarrow \quad \Gamma \vdash A \text{ true}$$

Implication

- ▶ *Q*: What is a proof of $A \Rightarrow B$?
- ▶ *A*: A proof of B , conditioned on a proof of A .
- ▶ *Q*: What can you do with a proof of $A \Rightarrow B$?
- ▶ *A*: Given a proof of A , make a proof of B .

Implication rules

- ▶ A little more interesting...

$$\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \Rightarrow B \text{ true}}$$

$$\frac{\Gamma \vdash A \Rightarrow B \text{ true} \quad \Gamma \vdash A \text{ true}}{\Gamma \vdash B \text{ true}}$$

Implication simplification

- ▶ Using Substitution Principle:

$$\frac{\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \Rightarrow B \text{ true}} \quad \Gamma \vdash A \text{ true}}{\Gamma \vdash B \text{ true}}$$



$$\begin{array}{l} \Gamma \vdash A \text{ true} \\ : \\ \Gamma \vdash B \text{ true} \end{array}$$

$$\Gamma \vdash A \Rightarrow B$$

Proof terms

- ▶ Compact *representation* of derivations

$$\frac{\Gamma \vdash M : A \text{ true} \quad \Gamma \vdash N : B \text{ true}}{\Gamma \vdash (M, N) : A \wedge B \text{ true}}$$

$$\frac{\Gamma \vdash M : A \wedge B \text{ true}}{\Gamma \vdash \pi_1 M : A \text{ true}}$$

$$\frac{\Gamma \vdash M : A \wedge B \text{ true}}{\Gamma \vdash \pi_2 M : B \text{ true}}$$

Proof terms

- ▶ Hypothesis get labels: now Γ is $x_1 : A_1, \dots, x_n : A_n$
- ▶ Hypothesis rule:

$$\frac{}{\Gamma, x : A \text{ true} \vdash x : A \text{ true}}$$

- ▶ **Substitution Principle:** if $\Gamma, x : A \text{ true} \vdash M : B \text{ true}$
and $\Gamma \vdash N : A \text{ true}$, then $\Gamma \vdash [N/x] M : B \text{ true}$

Proof terms

- ▶ Abstraction and application

$$\Gamma, x : A \text{ true} \vdash M : B \text{ true}$$

$$\Gamma \vdash \lambda v. \Delta \quad \lambda M. \Delta \rightarrow R \text{ true}$$
$$\Gamma \vdash M : A \Rightarrow B \text{ true} \quad \Gamma \vdash N : A \text{ true}$$

$$\Gamma \vdash M N : B \text{ true}$$

Proof term simplification

- ▶ Reduction on trees \Rightarrow reduction on terms

$$\pi_1 (M, N) \Longrightarrow M$$

$$\pi_2 (M, N) \Longrightarrow N$$

$$(\lambda x:A. M) N \Longrightarrow [N/x] M$$

- ▶ This is a *programming language!*

Classical proof terms

⊠ $e ::= x \mid \text{letcc}(u \div A. c)$
| $(e_1, e_2) \mid ()$
| $\text{inl}(e) \mid \text{inr}(e)$
| $\lambda x:A. e$
| $\text{not}(k)$

$A \text{ true}$
 $A \wedge B, \top$
 $A \vee B$
 $A \Rightarrow B$
 $\neg A$

⊠ $k ::= u \mid \text{let}(x:A. c)$
| $k \circ \pi_1 \mid k \circ \pi_2$
| $[k_1, k_2] \mid []$
| $e; k$
| $\text{not}(e)$

$A \text{ false}$
 $A \wedge B$
 $A \vee B, \perp$
 $A \Rightarrow B$
 $\neg A$