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

3

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

5

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"

5

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")

6

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 …”)

6

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 […])

7

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 $\boxed{A}$ (since A and […])
      - have $\boxed{B}$ (by assumption)
      - thus $\boxed{A \text{ and } B}$ (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

9

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

$$\frac{A\ \mathit{true} \qquad B\ \mathit{true}}{A \wedge B\ \mathit{true}}\ \wedge\textbf{-I}$$

9

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 […] 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 eith
    - thus eithe
  - thus, if *A* and either *B* or *C*, then either *A* and *B*, or *A* and *C*

$$\frac{A \wedge B\ \mathit{true}}{A\ \mathit{true}}\ \mathbf{\wedge\text{-}E_1} \qquad \frac{A \wedge B\ \mathit{true}}{B\ \mathit{true}}\ \mathbf{\wedge\text{-}E_2}$$

9

Formalizing Proof, take 1

- Judgement: *A true*. ("*A* is provable.")
- Inference rules: grouped into "Introductions":

$$\frac{A\ \mathit{true} \qquad B\ \mathit{true}}{A \wedge B\ \mathit{true}}\ \mathbf{\wedge\text{-}I}$$

- … and "Eliminations":

$$\frac{A \wedge B\ \mathit{true}}{A\ \mathit{true}}\ \mathbf{\wedge\text{-}E_1} \qquad \frac{A \wedge B\ \mathit{true}}{B\ \mathit{true}}\ \mathbf{\wedge\text{-}E_2}$$

10

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

11

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

11

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
    - 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

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

Formalizing Proof, take 2

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

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

Formalizing Proof, take 2

- Judgement: $\Gamma \vdash A\ \mathit{true}$. ("$A$ is provable assuming Γ")
- Γ is a list of assumptions: $A_1\ \mathit{true}, \ldots, A_n\ \mathit{true}$
- Implication: one introduction rule:

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

- … and one elimination rule:

$$\frac{\Gamma \vdash \mathrm{A} \Rightarrow \mathrm{B}\ \mathit{true} \qquad \Gamma \vdash \mathrm{A}\ \mathit{true}}{\Gamma \vdash \mathrm{B}\ \mathit{true}}\ \mathbf{\Rightarrow\text{-}E}$$

Reasoning from assumptions

- Hypothesis rule:

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

Reasoning from assumptions

- Hypothesis rule:

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

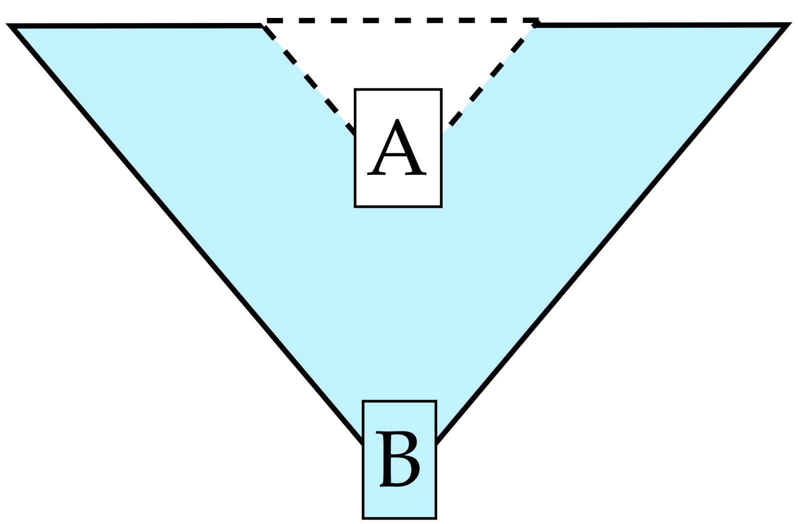
- **Substitution Principle**: if $\Gamma, A\ \mathit{true} \vdash B\ \mathit{true}$ and $\Gamma \vdash A\ \mathit{true}$, then $\Gamma \vdash B\ \mathit{true}$

Reasoning from assumptions

- Hypothesis rule:

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

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

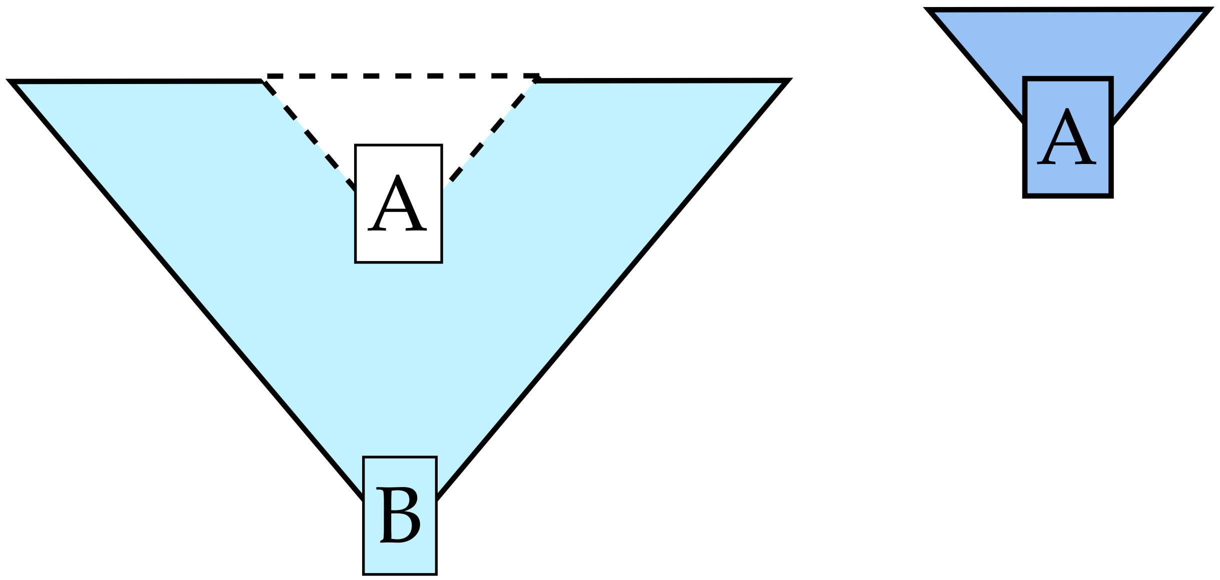


Reasoning from assumptions

- Hypothesis rule:

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

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



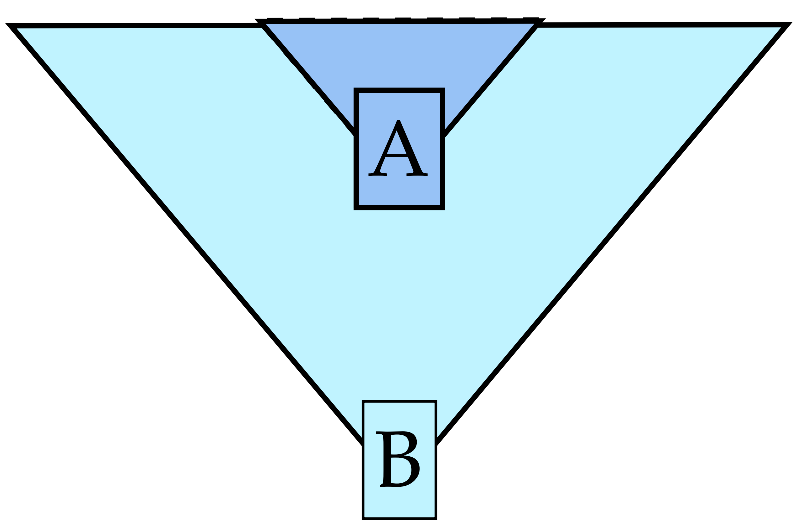
13

Reasoning from assumptions

- Hypothesis rule:

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

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



Formalizing Proof, take 3

- Disjunction: two introduction rules:

$$\frac{\Gamma \vdash A\ \mathit{true}}{\Gamma \vdash A \vee B\ \mathit{true}}\ \vee\text{-}\mathbf{I_1} \qquad \frac{\Gamma \vdash B\ \mathit{true}}{\Gamma \vdash A \vee B\ \mathit{true}}\ \vee\text{-}\mathbf{I_2}$$

14

Formalizing Proof, take 3

- Disjunction: two introduction rules:

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

- … and one elimination rule:

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

14

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*.

15

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 **∧-I**, we have $A \wedge B$ *true*.
  - And then by **∧-E$_1$**, 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 **∧-I**, we have $A \wedge B$ *true*.
  - And then by **∧-E$_1$**, we have A *true*.
  - *phew*

Proof simplification

- Eliminate “redundant” steps

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D} \\ \Gamma \vdash A\ \mathit{true}\end{array} \quad \begin{array}{c}\mathcal{E} \\ \Gamma \vdash B\ \mathit{true}\end{array}}{\Gamma \vdash A \wedge B\ \mathit{true}}}{\Gamma \vdash A\ \mathit{true}}$$

16

Proof simplification

- Eliminate "redundant" steps

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D} \\ \Gamma \vdash A\ \mathit{true}\end{matrix} \qquad \begin{matrix}\mathcal{E} \\ \Gamma \vdash B\ \mathit{true}\end{matrix}}{\Gamma \vdash A \wedge B\ \mathit{true}}}{\Gamma \vdash A\ \mathit{true}} \quad \Rightarrow \quad \begin{matrix}\mathcal{D} \\ \Gamma \vdash A\ \mathit{true}\end{matrix}$$

Proof simplification

- Using Substitution Principle:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \Gamma, A\ \mathit{true} \vdash B\ \mathit{true}\end{array}}{\Gamma \vdash A \Rightarrow B\ \mathit{true}} \quad \begin{array}{c}\mathcal{E}\\ \Gamma \vdash A\ \mathit{true}\end{array}}{\Gamma \vdash B\ \mathit{true}}$$

Proof simplification

- Using Substitution Principle:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \Gamma, A\ \mathit{true} \vdash B\ \mathit{true}\end{array}}{\Gamma \vdash A \Rightarrow B\ \mathit{true}} \qquad \begin{array}{c}\mathcal{E}\\ \Gamma \vdash A\ \mathit{true}\end{array}}{\Gamma \vdash B\ \mathit{true}} \quad \Longrightarrow \quad \begin{array}{c}\mathcal{E}\\ \Gamma \vdash A\ \mathit{true}\\ \vdots\\ \Gamma \vdash B\ \mathit{true}\end{array}$$

17

Proof terms

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

$$\frac{\Gamma \vdash A \wedge B\ \mathit{true}}{\Gamma \vdash A\ \mathit{true}}\ \mathbf{\wedge\text{-}E_1} \qquad \frac{\Gamma \vdash A \wedge B\ \mathit{true}}{\Gamma \vdash B\ \mathit{true}}\ \mathbf{\wedge\text{-}E_2}$$

18

Proof terms

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

$$\frac{\Gamma \vdash A \wedge B\ \mathit{true}}{\Gamma \vdash A\ \mathit{true}}\ \mathbf{\wedge\text{-}E_1} \qquad \frac{\Gamma \vdash A \wedge B\ \mathit{true}}{\Gamma \vdash B\ \mathit{true}}\ \mathbf{\wedge\text{-}E_2}$$

18

Proof terms

$$\frac{\Gamma \vdash e_1 : A\ \mathit{true} \qquad \Gamma \vdash e_2 : B\ \mathit{true}}{\Gamma \vdash (e_1, e_2) : A \wedge B\ \mathit{true}}\ \wedge\textbf{-I}$$

$$\frac{\Gamma \vdash e : A \wedge B\ \mathit{true}}{\Gamma \vdash \#1\ e : A\ \mathit{true}}\ \wedge\textbf{-E}_1 \qquad \frac{\Gamma \vdash e : A \wedge B\ \mathit{true}}{\Gamma \vdash \#2\ e : B\ \mathit{true}}\ \wedge\textbf{-E}_2$$

Proof terms

- Conjunction: pairing!

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

$$\frac{\Gamma \vdash e : A \wedge B\ \mathit{true}}{\Gamma \vdash \#1\ e : A\ \mathit{true}}\ \wedge\text{-}\mathbf{E_1} \qquad \frac{\Gamma \vdash e : A \wedge B\ \mathit{true}}{\Gamma \vdash \#2\ e : B\ \mathit{true}}\ \wedge\text{-}\mathbf{E_2}$$

18

Proof terms

$$\frac{\Gamma, x : \mathrm{A}\ \mathit{true} \vdash \mathrm{e} : \mathrm{B}\ \mathit{true}}{\Gamma \vdash \lambda x.\ \mathrm{e} : \mathrm{A} \Rightarrow \mathrm{B}\ \mathit{true}}\ \textbf{⇒-I}$$

$$\frac{\Gamma \vdash \mathrm{e}_1 : \mathrm{A} \Rightarrow \mathrm{B}\ \mathit{true} \qquad \Gamma \vdash \mathrm{e}_2 : \mathrm{A}\ \mathit{true}}{\Gamma \vdash \mathrm{e}_1\ \mathrm{e}_2 : \mathrm{B}\ \mathit{true}}\ \textbf{⇒-E}$$

19

Proof terms

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

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

$$\frac{\Gamma \vdash \mathrm{e}_1 : \mathrm{A} \Rightarrow \mathrm{B}\ \textit{true} \qquad \Gamma \vdash \mathrm{e}_2 : \mathrm{A}\ \textit{true}}{\Gamma \vdash \mathrm{e}_1\ \mathrm{e}_2 : \mathrm{B}\ \textit{true}} \Rightarrow\textbf{-E}$$

Proof terms

20

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!

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

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

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

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

Example Proof, revisited

22

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 ∧ (B ∨ C) ⇒

    case #2 x **of** inl y ⇒ inl (#1 x, y)

    | inr z ⇒ 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 ∧ (B ∨ C) ⇒

    case #2 x **of** inl y ⇒ inl (#1 x, y)

    | inr z ⇒ inr (#1 x, z)
- Computational content of proof: a simple input-shuffling program

22

Classical Logic

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

$$\frac{\Gamma, A\ \mathit{false} \vdash \mathit{contra}}{\Gamma \vdash A\ \mathit{true}} \qquad \frac{\Gamma \vdash C\ \mathit{true} \qquad \Gamma \vdash C\ \mathit{false}}{\Gamma \vdash \mathit{contra}}$$

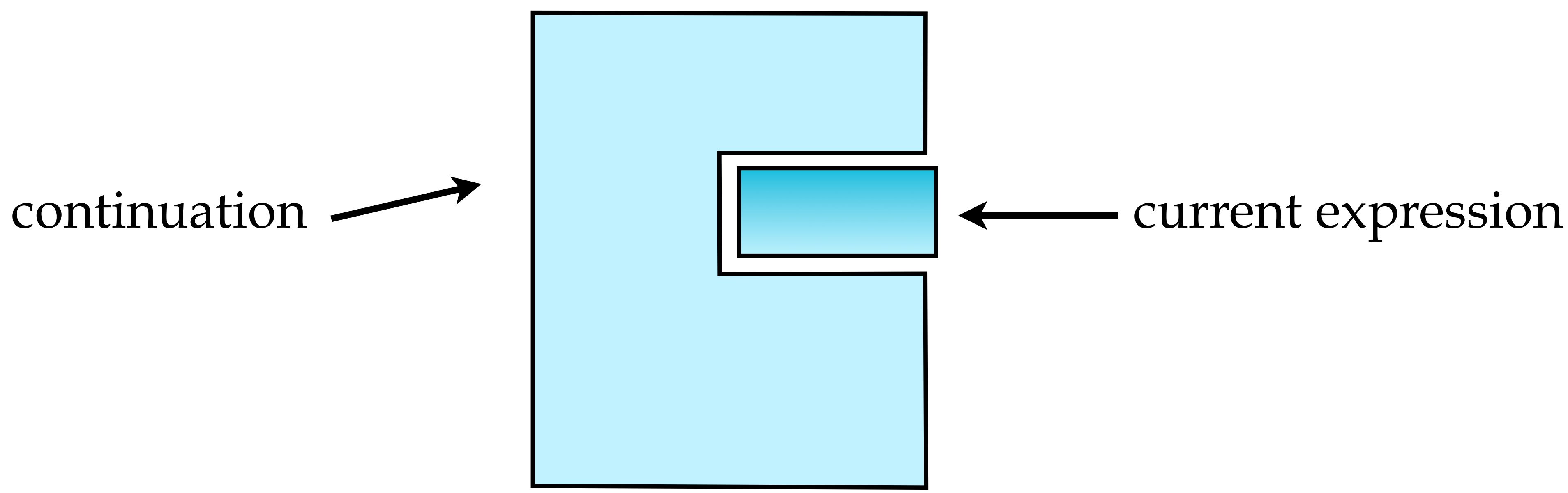
- What is *contra*?
- What is *false*?
- Computational interpretation?

23

Continuations

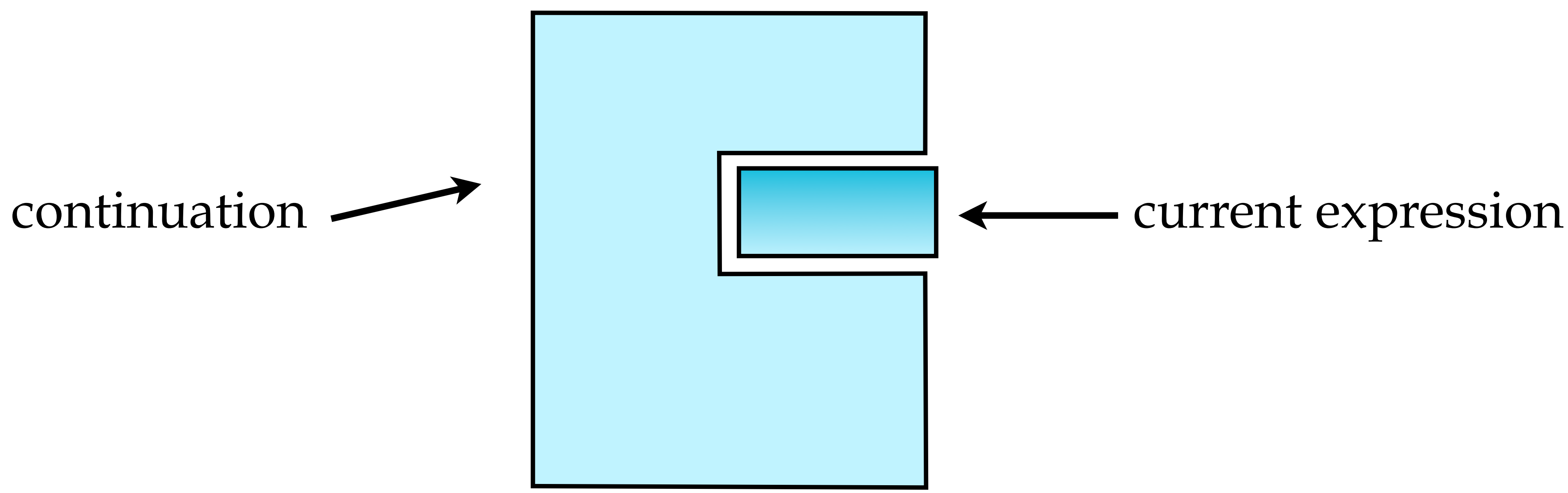
Continuations

- 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



26

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
```


27

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*

28

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* : **int cont**, since "product" should return an **int**

```
fun product nums =
   letcc u in
   let fun prod …
   in
        prod nums
   end
```


28

What can we *do* with them?

- Given an A **cont**, pass it an A

29

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

29

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

$$\boxed{A\ \textbf{cont} \Rightarrow A \wedge B\ \textbf{cont}}$$

$$\boxed{B\ \textbf{cont} \Rightarrow A \wedge B\ \textbf{cont}}$$

29

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**

- 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**

29

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**

- 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**, 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

$$\boxed{A \textit{ false} \vdash A \wedge B \textit{ false}}$$

$$\boxed{B \textit{ false} \vdash A \wedge B \textit{ false}}$$

- 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**

$$\boxed{A \textit{ false}, B \textit{ false} \vdash A \vee B \textit{ false}}$$

29

Continuations are refutations!

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

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

$$\frac{\Gamma \vdash k_1 : A\ \mathit{false} \qquad \Gamma \vdash k_2 : B\ \mathit{false}}{\Gamma \vdash [k_1, k_2] : A \vee B\ \mathit{false}}\ \textbf{∨-F}$$

30

Classical Curry-Howard

- Expressions: e : *A true*
- Continuations: k : *A false*

31

Classical Curry-Howard

- Expressions: $\mathrm{e} : A\ \mathit{true}$
- Continuations: $\mathrm{k} : A\ \mathit{false}$
- Programs: $\mathrm{k} \triangleleft \mathrm{e}$

$$\frac{\Gamma \vdash \mathrm{e} : \mathrm{C}\ \mathit{true} \qquad \Gamma \vdash \mathrm{k} : \mathrm{C}\ \mathit{false}}{\Gamma \vdash \mathrm{k} \triangleleft \mathrm{e} : \mathit{contra}}$$

31

Classical Curry-Howard

- Evaluate programs $k \triangleleft e$:

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

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

$$[k_1, k_2] \triangleleft \mathrm{inl}\ e \Longrightarrow k_1 \triangleleft e$$

$$[k_1, k_2] \triangleleft \mathrm{inr}\ e \Longrightarrow k_2 \triangleleft e$$

32

Proof-by-contradiction redux

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

33

Proof-by-contradiction redux

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

Proof-by-contradiction redux

$$\frac{\Gamma, u : \mathrm{A}\ \mathit{false} \vdash \mathrm{k} \lhd \mathrm{e} : \mathit{contra}}{\Gamma \vdash \mathbf{letcc}\ u\ \mathbf{in}\ \mathrm{k} \lhd \mathrm{e} : \mathrm{A}\ \mathit{true}}$$

$$\mathrm{k}' \lhd \mathbf{letcc}\ u\ \mathbf{in}\ \mathrm{k} \lhd \mathrm{e} \quad \Longrightarrow \quad [\mathrm{k}'/u]\,(\mathrm{k} \lhd \mathrm{e})$$

33

Classical proof terms

- e ::= x | **letcc** u:A *false* **in** c)
  | (e_1, e_2) | **()**
  | **inl**(e) | **inr**(e)
  | λx:A. e
  | **not**(k)

- k ::= u | **let** x:A *true* **in** c
  | #1; k | #2; k
  | [k_1, k_2] | **[]**
  | e; k
  | **not**(e)

<u>A *true*</u>

$A \wedge B$, $\top$

$A \vee B$

$A \Rightarrow B$

$\neg A$

<u>A *false*</u>

$A \wedge B$

$A \vee B$, $\bot$

$A \Rightarrow B$

$\neg A$

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

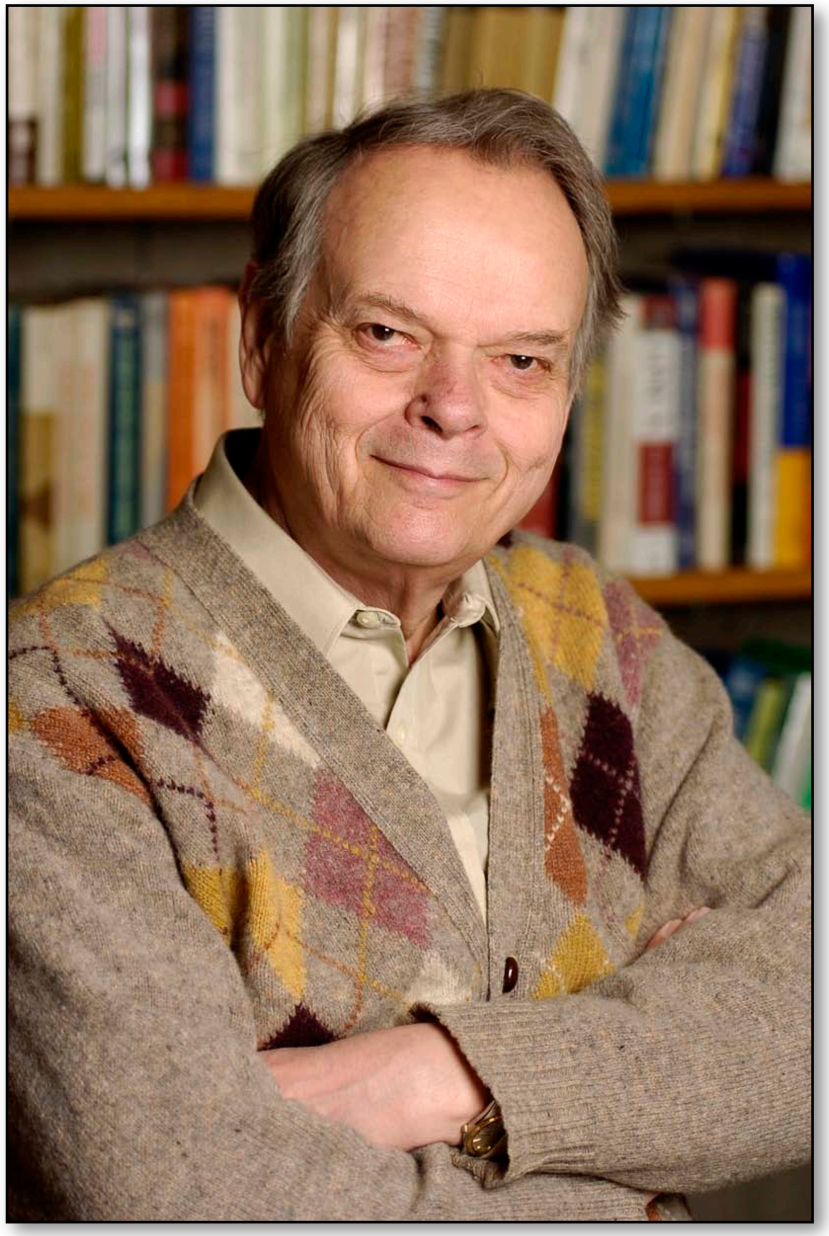
Prior work

- 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.

36

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)

37

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.

$$\frac{\mathrm{A}\ \mathit{true} \qquad \mathrm{B}\ \mathit{true}}{\mathrm{A} \wedge \mathrm{B}\ \mathit{true}}$$

38

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.)

$$\frac{A \wedge B \ \mathit{true}}{A \ \mathit{true}} \qquad \frac{A \wedge B \ \mathit{true}}{B \ \mathit{true}}$$

Reasoning from hypotheses

- Refine judgement: A *true* becomes $\Gamma \vdash A$ *true*, with Γ an unordered list of hypotheses $A_1, \ldots, A_n$

- Hypothesis rule:

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

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

40

Intros and Elims

- Introduction

$$\frac{\Gamma \vdash A \ \mathit{true} \quad \Gamma \vdash B \ \mathit{true}}{\Gamma \vdash A \land B \ \mathit{true}}$$

- Elimination

$$\frac{\Gamma \vdash A \land B \ \mathit{true}}{\Gamma \vdash A \ \mathit{true}} \qquad \frac{\Gamma \vdash A \land B \ \mathit{true}}{\Gamma \vdash B \ \mathit{true}}$$

41

Proof simplification

- Eliminate "redundant" steps

$$\dfrac{\dfrac{\Gamma \vdash A\ \textit{true} \qquad \Gamma \vdash B\ \textit{true}}{\Gamma \vdash A \wedge B\ \textit{true}}}{\Gamma \vdash A\ \textit{true}} \quad \Longrightarrow \quad \Gamma \vdash A\ \textit{true}$$

42

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…

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

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

44

Implication simplification

- Using Substitution Principle:

$$\dfrac{\dfrac{\Gamma, A\ \mathit{true} \vdash B\ \mathit{true}}{\Gamma \vdash A \Rightarrow B\ \mathit{true}} \qquad \Gamma \vdash A\ \mathit{true}}{\Gamma \vdash B\ \mathit{true}} \quad \Rightarrow \quad \begin{array}{c} \Gamma \vdash A\ \mathit{true} \\ \vdots \\ \Gamma \vdash B\ \mathit{true} \end{array}$$

$\Gamma \vdash A \Rightarrow B$

Proof terms

- Compact *representation* of derivations

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

$$\frac{\Gamma \vdash M : A \wedge B\ \mathit{true}}{\Gamma \vdash \pi_1\ M : A\ \mathit{true}} \qquad \frac{\Gamma \vdash M : A \wedge B\ \mathit{true}}{\Gamma \vdash \pi_2\ M : B\ \mathit{true}}$$

Proof terms

- Hypothesis get labels: now Γ is $x_1 : A_1, \ldots, x_n : A_n$

- Hypothesis rule:

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

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

Proof terms

- Abstraction and application

$$\frac{\Gamma, x : A\ \mathit{true} \vdash M : B\ \mathit{true}}{\Gamma \vdash \lambda x{:}A.\ M : A \Rightarrow B\ \mathit{true}}$$

$$\frac{\Gamma \vdash M : A \Rightarrow B\ \mathit{true} \quad \Gamma \vdash N : A\ \mathit{true}}{\Gamma \vdash M\,N : B\ \mathit{true}}$$

48

Proof term simplification

- Reduction on trees ⇒ reduction on terms

$$\pi_1\ (M, N) \ \texttt{====>}\ M$$

$$\pi_2\ (M, N) \ \texttt{====>}\ N$$

$$(\lambda x{:}A.\ M)\ N \ \texttt{====>}\ [N/x]\ M$$

- This is a *programming language!*

Classical proof terms

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

<u>A *true*</u>

$A \wedge B, \top$

$A \vee B$

$A \Rightarrow B$

$\neg A$

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

<u>A *false*</u>

$A \wedge B$

$A \vee B, \bot$

$A \Rightarrow B$

$\neg A$

50