Constructive Logic

Hype for Types

February 16, 2021

Proofs

Existence

I want to prove there exists a set with property P.
Is one of these more useful?

- Proof by contradiction: If such a set did not exist, we'd have a contradiction (insert proof here), therefore it must exist
- Direct proof: The set S has property P (insert proof here)

Existence

I want to prove there exists an algorithm to convert SML into x86 assembly.

Is one method more useful now?

- Proof by contradiction: If such a compiler did not exist, we'd have a contradiction (insert proof here), therefore it must exist
- Direct proof : CakeML (formally verified SML compiler)

To Construct or Not To Construct

Two kinds of proofs

- *Non-Constructive* : demonstrate the existence of a mathematical object, but *without* telling you what it is
- *Constructive* : demonstrate the existence of a mathematical object precisely by presenting an object and proving it has the desired properties

A Concrete but Boring Example

Does there exist $a, b : \mathbb{R}$ such that a, b irrational but a^b rational?

- *Non-Constructive*
  If $\sqrt{2}^{\sqrt{2}}$ rational, we're done. Otherwise $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = 2$.
- *Constructive*
  Take $a = \sqrt{2}$ and $b = log_2 9$. Then $\sqrt{2}^{log_2 9} = 9^{log_2 \sqrt{2}} = 9^{\frac{1}{2}} = 3$

Constructive proofs are useful to computer scientists

Constructive proofs provide *algorithms*!

A proof that all natural numbers have property P must describe a way to construct a proof of $P(n)$ for each $n : \mathbb{N}$

Formalization B)

- "You have to construct something" is pretty vague
- How do we formalize what it means for a proof to be constructive?
    1. Decide what kinds of proposition we want to talk about
    2. Inference rules!

Formalization B)

A few reasonable kinds of proposition

- $\top$
- $\bot$
- $A \land B$
- $A \lor B$
- $A \Rightarrow B$
- $\neg A$

Constructive Logic: Inference Rules

Conjunction ($\wedge$)

To get $A \wedge B$ (*introduction*), we need... a proof of A and a proof of B:

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge I)$$

Given $A \wedge B$, we can extract two facts (*elimination*)... A and B:

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ (\wedge E_1) \qquad\qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ (\wedge E_2)$$

Implication ($\Rightarrow$)

To get $A \Rightarrow B$, we need... a proof of B *assuming* a proof of A:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ (\Rightarrow I)$$

Given $A \Rightarrow B$ and A, we can extract... B:

$$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\ (\Rightarrow E)$$

Disjunction ($\vee$)

To get $A \vee B$, we need... a proof of A *or* a proof of B:

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee I_1) \qquad\qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee I_2)$$

Given $A \vee B$, we can extract... nothing? But if we *also* have "given A, then C" and "given B, then C," we can get C:

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee E)$$

Truth ($\top$) and Falsehood ($\bot$)

To get $\top$, we need... nothing!

$$\frac{}{\Gamma \vdash \top}\ (\top I)$$

Can we get any information out of $\top$? No!

How can we get $\bot$? We can't!
But given $\bot$, we can obtain a proof of... anything!

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ (\bot E)$$

What about Negation ($\neg$)?

What counts as a proof of $\neg A$?
We need to show something like "it's impossible to prove A"
Do we need new inference rules?
No!

$$\neg A \equiv A \Rightarrow \bot$$

$\neg A$ means "If we can prove A, we can do something impossible".

All the rules!

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge I) \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ (\wedge E_1) \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ (\wedge E_2)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ (\Rightarrow I) \qquad \frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\ (\Rightarrow E) \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee I_1)$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee I_2) \qquad \frac{}{\Gamma \vdash \top}\ (\top I) \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ (\bot E)$$

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee E)$$

Question

Does this seem familiar...?

Programs are Proofs

Back to the Simply-Typed Lambda Calculus

$$\frac{\Gamma \vdash e_1 : A \quad \Gamma \vdash e_2 : B}{\Gamma \vdash \langle e_1, e_2 \rangle : A \wedge B}\ (\wedge I) \qquad \frac{\Gamma \vdash e : A \wedge B}{\Gamma \vdash \mathbf{fst}(e) : A}\ (\wedge E_1)$$

$$\frac{\Gamma \vdash e : A \wedge B}{\Gamma \vdash \mathbf{snd}(e) : B}\ (\wedge E_2) \qquad \frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash (\lambda x : A.\ e) : A \Rightarrow B}\ (\Rightarrow I)$$

$$\frac{\Gamma \vdash e_1 : A \Rightarrow B \quad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1\ e_2 : B}\ (\Rightarrow E) \qquad \frac{\Gamma \vdash e : A}{\Gamma \vdash \mathbf{Left}\ e : A \vee B}\ (\vee I_1)$$

$$\frac{\Gamma \vdash e : B}{\Gamma \vdash \mathbf{Right}\ e : A \vee B}\ (\vee I_2) \qquad \frac{}{\Gamma \vdash \langle\rangle : \top}\ (\top I) \qquad \frac{\Gamma \vdash e : \bot}{\Gamma \vdash \mathbf{absurd}(e) : A}\ (\bot E)$$

$$\frac{\Gamma \vdash e_1 : A \vee B \quad \Gamma, x_1 : A \vdash e_2 : C \quad \Gamma, x_2 : B \vdash e_3 : C}{\Gamma \vdash \mathbf{case}\ e_1\ \mathbf{of}\ x_1 \Rightarrow e_2 \mid x_2 \Rightarrow e_3 : C}\ (\vee E)$$

Let's Prove Some Stuff!

Some Examples

Theorem: Identity

Prove $A \Rightarrow A$.

$\lambda x : A.\ x$

Theorem

Prove $A \wedge B \Rightarrow A$.

$\lambda x : A \times B.\ \mathbf{fst}(x)$

Theorem: Currying

Prove $(A \wedge B \Rightarrow C) \Rightarrow A \Rightarrow B \Rightarrow C$.

$\lambda f : A \times B \to C.\ \lambda a : A.\ \lambda b : B.\ f\ \langle a, b \rangle$, or `Fn.curry` in SML

More Examples

Theorem

Prove $\bot \vee \top$.

Right $\langle\rangle$

Theorem: Distributivity

Prove $A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$.

$\lambda x : A \times (B + C).$ **case snd**(x) **of** $x_1 \Rightarrow$ **Left** $\langle$**fst**$(x), x_1\rangle \mid x_2 \Rightarrow$
Right $\langle$**fst**$(x), x_2\rangle$
$\lambda x : (A \times B) + (A \times C).$ **case** x **of** $x_1 \Rightarrow \langle$**fst**(x_1), **Left snd**$(x_1)\rangle \mid x_2 \Rightarrow$
$\langle$**fst**(x_2), **Right snd**$(x_2)\rangle$

Even More

Theorem

Prove $A \wedge \neg A \Rightarrow B$. In other words, $A \wedge (A \Rightarrow \bot) \Rightarrow B$.

$\lambda x : A \times (A \rightarrow \textbf{void}).\ \textbf{absurd}(\textbf{snd}(x)\ \textbf{fst}(x))$

Theorem: Contrapositive

Prove $(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$.

$\lambda f : A \rightarrow B.\ \lambda g : B \rightarrow \textbf{void}.\ \lambda x : A.\ g\ (f\ x)$

Is there anything we can't prove constructively?

- Law of Excluded Middle : $P \vee \neg P$
- Double Negation Elimination : $\neg\neg P \Rightarrow P$

(These are actually equivalent)

So what?

Practical Implications

Practical Implications

- If a computer can check if a *program* has certain *type*, a computer can also check if a *proof* proves a certain *proposition*
- Computers can automatically check if proofs are correct!
- Manually grading Concepts homework? A thing of the past!
- Mathematician writes a long proof and someone finds an error years later? Fuggedaboutit!

Question

These proofs seem pretty boring, can a type system express more complicated propositions?