Dependent Types & Theorem Proving

A short story about theorems and computers

Alice Sayutina

August 5, 2024

Proofs in real life

1. Alice lives in Netherlands
2. Everyone in Netherlands drives a bicycle

Proofs in real life

1. Alice lives in Netherlands
2. Everyone in Netherlands drives a bicycle
3. ($\Longrightarrow$) Alice drives a bicycle

Proofs in real life (2)

1. People living in Canada like poutine
2. Jane likes poutine
3. ($\Longrightarrow$) Jane lives in Canada

Proofs in real life (2)

1. People living in Canada like poutine
2. Jane likes poutine
3. ($\implies$) Jane lives in Canada

Probably doesn't sound right?

What should we do to make proofs more robust?

Introducing Propositional Logic

- Propositional logic deals with propositions (statements), which can be true or false.

Introducing Propositional Logic

- Propositional logic deals with propositions (statements), which can be true or false.
- We can derive propositions from other propositions which we have already proved.

Introducing Propositional Logic

- Propositional logic deals with propositions (statements), which can be true or false.
- We can derive propositions from other propositions which we have already proved.
- If $P \to Q$ (implies), and we know P, then Q.

Introducing Propositional Logic

- Propositional logic deals with propositions (statements), which can be true or false.
- We can derive propositions from other propositions which we have already proved.
- If $P \to Q$ (implies), and we know P, then Q.
- We can also write statements which are trivially true:

Introducing Propositional Logic

- Propositional logic deals with propositions (statements), which can be true or false.
- We can derive propositions from other propositions which we have already proved.
- If $P \to Q$ (implies), and we know P, then Q.
- We can also write statements which are trivially true:
- "My cat is cute and small $\to$ My cat is small".

Let's do things formally.

Axioms of Propositional Logic

1. THEN-1 $\varphi \to (\chi \to \varphi)$
2. THEN-2 $(\varphi \to (\chi \to \psi)) \to ((\varphi \to \chi) \to (\varphi \to \psi))$
3. AND-1 $\varphi \wedge \chi \to \varphi$
4. AND-2 $\varphi \wedge \chi \to \chi$
5. AND-3 $\varphi \to (\chi \to (\varphi \wedge \chi))$
6. OR-1 $\varphi \to \varphi \vee \chi$
7. OR-2 $\chi \to \varphi \vee \chi$
8. OR-3 $(\varphi \to \chi) \to ((\varphi \to \psi) \to (\varphi \vee \chi \to \psi))$
9. NOT-1 $(\varphi \to \chi) \to ((\varphi \to \neg\chi) \to \neg\varphi))$
10. NOT-2 $\varphi \to (\neg\varphi \to \chi)$
11. NOT-3 $\varphi \vee \neg\varphi$

MP (Modus Ponens rule): $P; P \to Q \models Q$.

Deriving theorem

Let Γ be set of formulas.
We say $\Gamma \models A$, if there exists sequence of formulas $\sigma_1, \sigma_2, \ldots, \sigma_n$, such that σ_i is

1. Either one of the axioms (substituting φ, χ, ψ with any formulas),
2. Or $\sigma_i \in \Gamma$
3. Or σ_i is obtained by Modus Ponens from σ_j and $\sigma_{j'}$, $j, j' < i$.

Proof example

Examples

Let's prove: $\neg a, b| = a \rightarrow b$

- $b \rightarrow (a \rightarrow b)$ *axiom*
- b *from* Γ
- $a \rightarrow b$ *MP*

Remarks

Deduction theorem

$$\Gamma \models \varphi \to \psi \iff \Gamma, \varphi \models \psi$$

(Exercise: prove this!).

Remarks

Deduction theorem

$$\Gamma \models \varphi \to \psi \iff \Gamma, \varphi \models \psi$$

(Exercise: prove this!).

Remark

Propositional logic lacks a lot in the expression power, for instance we don't have $\exists$, $\forall$, and all the objects are boolean-valued (compared to more generic domain-specific things like vectors, or polynomials).

Remarks

Deduction theorem

$\Gamma \models \varphi \to \psi \iff \Gamma, \varphi \models \psi$

(Exercise: prove this!).

Remark

Propositional logic lacks a lot in the expression power, for instance we don't have $\exists$, $\forall$, and all the objects are boolean-valued (compared to more generic domain-specific things like vectors, or polynomials).
There are extensions of propositional logic which take care of that (First-Order Logic, Second-Order Logic, ...)

Remarks

Deduction theorem

$\Gamma \models \varphi \to \psi \iff \Gamma, \varphi \models \psi$

(Exercise: prove this!).

Remark

Propositional logic lacks a lot in the expression power, for instance we don't have $\exists$, $\forall$, and all the objects are boolean-valued (compared to more generic domain-specific things like vectors, or polynomials).
There are extensions of propositional logic which take care of that (First-Order Logic, Second-Order Logic, ...)
Integers example:
$\forall f \exists g : ADD(f, g) = ZERO$

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$

By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$

By lemma of deduction, $\iff$ $a \to b \models \neg b \to \neg a$

By lemma of deduction, $\iff$ $a \to b, \neg b \models \neg a$.

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$

By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$

By lemma of deduction, $\iff a \to b, \neg b \models \neg a.$

- $a \to b$ (from Γ)

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$
By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$
By lemma of deduction, $\iff a \to b, \neg b \models \neg a.$

- $a \to b$ (from Γ)
- $\neg b$ (from Γ)

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$
By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$
By lemma of deduction, $\iff a \to b, \neg b \models \neg a$.

- $a \to b$ (from Γ)
- $\neg b$ (from Γ)
- $\neg b \to (a \to \neg b)$ $(axiom)$

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$

By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$

By lemma of deduction, $\iff a \to b, \neg b \models \neg a$.

- $a \to b$ (from Γ)
- $\neg b$ (from Γ)
- $\neg b \to (a \to \neg b)$ $(axiom)$
- $a \to \neg b$ (MP)

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$
By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$
By lemma of deduction, $\iff a \to b, \neg b \models \neg a$.

- $a \to b$ (from Γ)
- $\neg b$ (from Γ)
- $\neg b \to (a \to \neg b)$ $(axiom)$
- $a \to \neg b$ (MP)
- $(a \to b) \to (a \to \neg b) \to (\neg a)$ $(axiom)$

Using lemma of deduction

Examples

Let's prove: $\models (a \to b) \to (\neg b \to \neg a)$

By lemma of deduction, $\iff a \to b \models \neg b \to \neg a$

By lemma of deduction, $\iff a \to b, \neg b \models \neg a$.

- $a \to b$ (from Γ)
- $\neg b$ (from Γ)
- $\neg b \to (a \to \neg b)$ $(axiom)$
- $a \to \neg b$ (MP)
- $(a \to b) \to (a \to \neg b) \to (\neg a)$ $(axiom)$
- $\neg a$ $(MP, \ twice)$.

Propositional Logic is Sound and Complete

definition

A logical formula is tautology, if it evaluates to true for any assignment of variables.

$x \to x$, $x \lor \neg x$, $\ldots$

Propositional Logic is Sound and Complete

definition

A logical formula is tautology, if it evaluates to true for any assignment of variables.

$x \to x$, $x \vee \neg x$, ...

Theorem

φ is tautology $\iff$ $\varnothing \models \varphi$.

One very special axiom

There is one special axiom:

$$\text{NOT-3: } \varphi \lor \neg\varphi$$

(aka Law of Excluded Middle, L.E.M.)

One very special axiom

There is one special axiom:

$$\text{NOT-3: } \varphi \vee \neg\varphi$$

(aka Law of Excluded Middle, L.E.M.)

Some people really don't like it.

Example of L.E.M. proofs

Theorem

There exists irrational $x, y \in \mathbb{I}$, *such that* $x^y \in \mathbb{Q}$.

Example of L.E.M. proofs

Theorem

There exists irrational $x, y \in \mathbb{I}$, such that $x^y \in \mathbb{Q}$.

Proof.

1. Consider $\sqrt{2}^{\sqrt{2}}$. If $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, we're done

Example of L.E.M. proofs

Theorem

There exists irrational $x, y \in \mathbb{I}$, such that $x^y \in \mathbb{Q}$.

Proof.

1. Consider $\sqrt{2}^{\sqrt{2}}$. If $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, we're done
2. Otherwise $\sqrt{2}^{\sqrt{2}} \in \mathbb{I}$

Example of L.E.M. proofs

Theorem

There exists irrational $x, y \in \mathbb{I}$, such that $x^y \in \mathbb{Q}$.

Proof.

1. Consider $\sqrt{2}^{\sqrt{2}}$. If $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, we're done
2. Otherwise $\sqrt{2}^{\sqrt{2}} \in \mathbb{I}$
3. Then consider $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2 \in \mathbb{Q}$ $\square$

Example of L.E.M. proofs

Theorem

There exists irrational $x, y \in \mathbb{I}$, such that $x^y \in \mathbb{Q}$.

Proof.

1. Consider $\sqrt{2}^{\sqrt{2}}$. If $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, we're done
2. Otherwise $\sqrt{2}^{\sqrt{2}} \in \mathbb{I}$
3. Then consider $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2 \in \mathbb{Q}$ $\square$

The proof is not "fair": we don't actually know x, y.

Introducing Intuitionistic Logic

Want to work with "constructive" math, where we know where exactly things are coming from.

Introducing Intuitionistic Logic

Want to work with "constructive" math, where we know where exactly things are coming from.
Specifically, we forbid law of excluded middle ($x \vee \neg x$) and equivalently powerful statements (i.e. $\neg\neg x \to x$).

Intuitionistic Logic

$\neg\psi$ is a shorthand for $\psi \to \bot$

Intuitionistic Logic

$\neg\psi$ is a shorthand for $\psi \to \bot$

Axioms

1. $\mathrm{FALSE}\ \bot \to \phi$
2. $\mathrm{THEN\text{-}1}\ \psi \to (\phi \to \psi)$
3. $\mathrm{THEN\text{-}2}\ \big(\chi \to (\phi \to \psi)\big) \to \big((\chi \to \phi) \to (\chi \to \psi)\big)$
4. $\mathrm{AND\text{-}1}\ \phi \wedge \chi \to \phi$
5. $\mathrm{AND\text{-}2}\ \phi \wedge \chi \to \chi$
6. $\mathrm{AND\text{-}3}\ \phi \to \big(\chi \to (\phi \wedge \chi)\big)$
7. $\mathrm{OR\text{-}1}\ \phi \to \phi \vee \chi$
8. $\mathrm{OR\text{-}2}\ \chi \to \phi \vee \chi$
9. $\mathrm{OR\text{-}3}\ (\phi \to \psi) \to \Big((\chi \to \psi) \to \big((\phi \vee \chi) \to \psi\big)\Big)$

Modus Ponens: $P, P \to Q \models Q$

Not possible in Intuitionistic Logic

There are multiple statements which are correct in Classic Logic, but are not provable in Intuitionistic Logic.

1. $x \vee \neg x$
2. $\neg\neg x \to x$
3. $((x \to y) \to x) \to x$

Not possible in Intuitionistic Logic

There are multiple statements which are correct in Classic Logic, but are not provable in Intuitionistic Logic.

1. $x \vee \neg x$
2. $\neg\neg x \to x$
3. $((x \to y) \to x) \to x$

It's possible to analyze if the statement is provable in Intuitionistic Logic by using i.e. Heyting Algebras.

Curry-Howard isomorphism

- Let's think of logical formulas as types...

Curry-Howard isomorphism

- Let's think of logical formulas as types...
- To prove formula, is to show a member of the corresponding type.

Curry-Howard isomorphism

- Let's think of logical formulas as types...
- To prove formula, is to show a member of the corresponding type.
- $\psi \to (\phi \to \psi)$ is a function which takes type ψ and returns a function from type ϕ to ψ.

Curry-Howard isomorphism

- Let's think of logical formulas as types...
- To prove formula, is to show a member of the corresponding type.
- $\psi \to (\phi \to \psi)$ is a function which takes type ψ and returns a function from type ϕ to ψ.

```
then1 : {TypeA, TypeB : Type} -> (a : TypeA) -> (b : TypeB) -> TypeA
then1 a b = a
```

Curry-Howard Isomorphism

Formal Logic	Programs
Formula	Type
Proof	Element of the Type
Formula is true	There is an element of the type
$a \to b$	Functions from a to b
$a \wedge b$	
$a \vee b$	
$\perp$	
$\forall x : p(x)$	
$\exists x : p(x)$	

Curry-Howard Isomorphism

Formal Logic	Programs
Formula	Type
Proof	Element of the Type
Formula is true	There is an element of the type
$a \to b$	Functions from a to b
$a \wedge b$	$Pair(a, b)$
$a \vee b$	$Union(a, b)$
$\perp$	Void-type
$\forall x : p(x)$	
$\exists x : p(x)$	

Curry-Howard Isomorphism

Formal Logic	Programs
Formula	Type
Proof	Element of the Type
Formula is true	There is an element of the type
$a \to b$	Functions from a to b
$a \wedge b$	$Pair(a, b)$
$a \vee b$	$Union(a, b)$
$\perp$	Void-type
$\forall x : p(x)$	$\prod$-type (Product-Type)
$\exists x : p(x)$	Σ-type (Sum-type)

Curry-Howard Isomorphism

Formal Logic	Programs
Formula	Type
Proof	Element of the Type
Formula is true	There is an element of the type
$a \to b$	Functions from a to b
$a \wedge b$	$Pair(a, b)$
$a \vee b$	$Union(a, b)$
$\perp$	Void-type
$\forall x : p(x)$	$\prod$-type (Product-Type)
$\exists x : p(x)$	Σ-type (Sum-type)

Observe there is no natural LEM in programs, so all the programs operate by Intuitionistic-logic rules

Curry-Howard Isomorphism: Idris

```
%hide Left
%hide Right

data Pair a b = MakePair a b

data Union a b = Left a | Right b

data Null : Type

or1 : {TypeA, TypeB : Type} -> (a : TypeA) -> (Union TypeA TypeB)
or1 a = Left a

absurd : {Anything: Type} -> Null -> Anything
-- no need to pattern match, since there are 0 cases to match on!
```

What are Pi & Sigma Types

```
data MyNat = Zero | Succ (MyNat)

data Vect : MyNat -> Type -> Type where
   Nil  : Vect Zero a
   (::) : a -> Vect k a -> Vect (Succ k) a


makeVec : {T : Type} -> (a : T) -> (n : MyNat) -> Vect n T
makeVec a Zero = Nil
makeVec a (Succ n) = a :: (makeVec a n)

-- makeIntVec is a Pi-Type:
makeIntVec : (n : MyNat) -> Vect n MyNat
makeIntVec = makeVec {T=MyNat} Zero

-- (m ** Vec m a) is a sigma-type
filter : (a -> Bool) -> Vect n a -> (m ** Vect m a)
filter p Nil = (Zero ** Nil)
filter p (x :: xs) =
 let (m ** vec) = filter p xs
 in case p x of
   False => (m ** vec)
   True => (Succ m ** x :: vec)
```

What is the equality

```
Main>
Main> 0 = 0
0 = 0
Main> :t 0 = 0
0 = 0 : Type
Main> :t 0 = 1
0 = 1 : Type
Main> :t Refl
Builtin.Refl : x = x
Main> -- Refl is a constructor for equality (lies in any x=x type)
Main>
```

Overview is finished

Time to do some fun things!