Natural Deduction

Ethan Zhang

University of Illinois Urbana-Champaign

Outline

Section 1

Logic Refresher

Basic Notions From Logic

Definition

A *formula* is a string in a formal language. A *sentence* is a formula with no free variables. A *statement* is a sentence with a definite truth value.

Basic Notions From Logic

Definition

A *formula* is a string in a formal language. A *sentence* is a formula with no free variables. A *statement* is a sentence with a definite truth value.

Definition

An *argument* is a sequence of statements. We say an argument is *valid* if every statement deductively follows from previous statements, and call it *sound* if it is valid and its premises hold.

Propositional (or Zeroth-Order) Logic

Propositional Logic is the most basic form of symbolic logic; its language is inductively defined by atomic propositions and truth functions.

Propositional (or Zeroth-Order) Logic

Propositional Logic is the most basic form of symbolic logic; its language is inductively defined by atomic propositions and truth functions.

Definition

A *truth function* is a function that takes truth values to truth values. For example, a *logical connective* is a truth function since the truth of a statement $P \wedge Q$ depends solely on the truth values of the substatements P and Q.

Propositional (or Zeroth-Order) Logic

$$
\begin{array}{llcll}
\textsf{Formulae} & \mathscr{A} & ::= & P & \text{propositions} \\
 & & \mid & \top & \text{top (true)} \\
 & & \mid & \bot & \text{bottom (false)} \\
 & & \mid & \mathscr{P} \wedge \mathscr{Q} & \text{conjunction} \\
 & & \mid & \mathscr{P} \vee \mathscr{Q} & \text{disjunction} \\
 & & \mid & \mathscr{P} \Rightarrow \mathscr{Q} & \text{implication} \\
 & & \mid & \neg \mathscr{P} & \text{negation}
\end{array}
$$

First-Order Logic (FOL)

First-Order Logic extends propositional logic with predicates and quantification over objects.

First-Order Logic (FOL)

First-Order Logic extends propositional logic with predicates and quantification over objects.

Definition

An *object* belongs to the *domain* of a theory, e.g. natural numbers in number theory, sets in set theory, etc. A *predicate* can be thought of like a function on objects: it takes in an object and outputs a truth value depending on said object and the *interpretation* in the model.

First-Order Logic (FOL)

Terms	τ	$::=$	x	variables
Formulae	$\mathscr{A}$	$::=$	$P[\tau_1, \ldots, \tau_n]$	predicates
		$\mid$	$\top$	top (true)
		$\mid$	$\bot$	bottom (false)
		$\mid$	$\mathscr{P} \wedge \mathscr{Q}$	conjunction
		$\mid$	$\mathscr{P} \vee \mathscr{Q}$	disjunction
		$\mid$	$\mathscr{P} \Rightarrow \mathscr{Q}$	implication
		$\mid$	$\neg \mathscr{P}$	negation
		$\mid$	$\forall x. \mathscr{P}[x]$	universal quantification
		$\mid$	$\exists x. \mathscr{P}[x]$	existential quantification

Intuitionistic vs. Classical Logic

- **Truth is objectively real:** Logical sentences have definite truth values, even if we can't (or don't yet know how to) prove them.
- **Truth is a mental construct:** Truth is subjective and dependent on the mind of the mathematician. Hence, a sentence is only true or false when we can provide an explicit construction for an object that witnesses its truth value.

In particular, *Classical Logic* admits the *Law of Excluded Middle* whereas *Intuitionistic Logic* rejects it.

Section 2

A Bit of Proof Theory

Proof Theory?

Proof theory is the study of "formal" proofs as mathematical objects. Historically, it stemmed out of a desire to reduce mathematics to a *syntactical* game governed by axioms and inference rules.

Proof Theory?

In particular, it studies *proof systems* and their strengths:

- What is the “structure” of a proof?
- Can a particular proof system prove (derive) some sentence?
- Is there any extra “meaning”/“insight” we can get from a particular proof?

This is in contrast to *model theory*, which studies the semantics of mathematical theories, i.e. in what contexts are so-and-so sentences true?

Proof Calculi

Definition (from the Pr∞f Wiki)

A *proof system* $\mathscr{P}$ for a formal language $\mathscr{L}$ comprises:

- *Axioms* and/or *axiom schemata*;
- *Rules of inference* for deriving theorems.

Proof Calculi

Definition (from the Pr∞f Wiki)

A *proof system* $\mathscr{P}$ for a formal language $\mathscr{L}$ comprises:

- *Axioms* and/or *axiom schemata*;
- *Rules of inference* for deriving theorems.

We write $\Gamma \vdash_{\mathscr{P}} \varphi$ if the formula φ is *provable* (or *derivable*) in $\mathscr{P}$ from a set of assumptions Γ; that is, if there is some sequence of deductions (governed by $\mathscr{P}$) that starts at Γ and ends at φ.

“Examples”

Hilbert System:

- Many axioms.
- Few inference rules (typically only Modus Ponens).
- Reasoning is primarily justified by axioms.
- Proofs consist of a sequence of formulae, each of which either is an axiom or follows from a deduction.

Natural Deduction:

- Few (or no) axioms.
- “Natural” inference rules.
- Proofs are expressed by inference, mimicking the “natural” way of reasoning.
- **Fitch-style** proofs look similar to those in a Hilbert system. **Gentzen-style** proofs make use of “proof trees” (à la CS 421).

Applications

- **Formal verification** of software systems;
- **Proof automation** in mathematics;
- **Logic programming**;
- **Semantics** for programming (and natural!) languages.

Section 3

(Fitch-Style) Natural Deduction for (Intuitionistic) Propositional Logic

Idea/Motivation

- Truth tables don't give us very much insight into why an argument is true (and don't really work for intuitionistic logics). More importantly, verifying complex arguments becomes intractable!
- Arguments involving substituting logical equivalences are hard to come up with, hard to follow, and can hide implicit assumptions.

Idea/Motivation

What if we instead focus on reasoning in the "natural" way?

> *One of Gentzen's main motivations was to devise rules that model mathematical reasoning as directly as possible, although clearly in much more detail than in a typical mathematical argument.*
>
> — Frank Pfenning (2023)

Language

$$
\begin{array}{llll}
\textsf{Formulae} \quad \mathscr{A} & ::= & P & \text{propositions} \\
& \mid & \top & \text{top (true)} \\
& \mid & \bot & \text{bottom (false)} \\
& \mid & \mathscr{P} \wedge \mathscr{Q} & \text{conjunction} \\
& \mid & \mathscr{P} \vee \mathscr{Q} & \text{disjunction} \\
& \mid & \mathscr{P} \Rightarrow \mathscr{Q} & \text{implication} \\
& \mid & \neg \mathscr{P} & \text{negation}
\end{array}
$$

Reiteration

m	$\mathscr{A}$	
n	$\mathscr{A}$	R, m

Conjunction

Conjunction

m	$\mathscr{A}$	
n	$\mathscr{B}$	
l	$\mathscr{A} \wedge \mathscr{B}$	$\wedge$I, m, n

Conjunction

m	$\mathscr{A}$	
n	$\mathscr{B}$	
l	$\mathscr{A} \wedge \mathscr{B}$	$\wedge$I, m, n

m	$\mathscr{A} \wedge \mathscr{B}$	
n	$\mathscr{A}$	$\wedge$E, m

m	$\mathscr{A} \wedge \mathscr{B}$	
n	$\mathscr{B}$	$\wedge$E, m

Implication

Implication

m	$\quad \mathscr{A}$	
n	$\quad \mathscr{B}$	
l	$\mathscr{A} \Rightarrow \mathscr{B}$	$\Rightarrow$I, m–n

Implication

m	$\quad\mathscr{A}$ (assumption)	
n	$\quad\mathscr{B}$	
l	$\mathscr{A} \Rightarrow \mathscr{B}$	$\Rightarrow$I, m–n

m	$\mathscr{A} \Rightarrow \mathscr{B}$	
n	$\mathscr{A}$	
l	$\mathscr{B}$	$\Rightarrow$E, m, n

Disjunction

Disjunction

m	$\mathscr{A}$	
n	$\mathscr{A} \vee \mathscr{B}$	$\vee$I, m

m	$\mathscr{B}$	
n	$\mathscr{A} \vee \mathscr{B}$	$\vee$I, m

Disjunction

m	$\mathscr{A}$	
n	$\mathscr{A} \vee \mathscr{B}$	$\vee$I, m

m	$\mathscr{B}$	
n	$\mathscr{A} \vee \mathscr{B}$	$\vee$I, m

m	$\mathscr{A} \vee \mathscr{B}$	
i	$\quad \mathscr{A}$	
j	$\quad \mathscr{C}$	
k	$\quad \mathscr{B}$	
l	$\quad \mathscr{C}$	
n	$\mathscr{C}$	$\vee$E, m, i–j, k–l

Negation

Negation

m	$\quad \mathscr{A}$	
n	$\quad \bot$	
l	$\neg\mathscr{A}$	$\neg$I, m–n

Negation

m	$\quad \mathscr{A}$	
n	$\quad \bot$	
l	$\neg\mathscr{A}$	$\neg$I, m–n

m	$\neg\mathscr{A}$	
n	$\mathscr{A}$	
l	$\bot$	$\neg$E, m, n

Explosion

m	$\bot$	
n	$\mathscr{A}$	$\bot$E, m

Subsection 1

Examples

$$A \Rightarrow (B \Rightarrow C) \vdash (A \land B) \Rightarrow C$$

1	$A \Rightarrow (B \Rightarrow C)$	
2	$A \land B$	
3	A	$\land$E, 2
4	B	$\land$E, 2
5	$B \Rightarrow C$	$\Rightarrow$E, 1, 3
6	C	$\Rightarrow$E, 5, 4
7	$(A \land B) \Rightarrow C$	$\Rightarrow$I, 2–6

$$P \land (Q \lor R), P \Rightarrow \neg R \vdash Q \lor E$$

1	$P \land (Q \lor R)$	
2	$P \Rightarrow \neg R$	
3	P	$\land$E, 1
4	$Q \lor R$	$\land$E, 1
5	$\neg R$	$\Rightarrow$E, 2, 3
6	$\quad Q$	
7	$\quad Q \lor E$	$\lor$I, 6
8	$\quad R$	
9	$\quad \bot$	$\neg$E, 5, 8
10	$\quad Q \lor E$	$\bot$E, 9
11	$Q \lor E$	$\lor$E, 4, 6–7, 8–10

Section 4

Extending Natural Deduction to FOL

Language

Terms	τ	$::=$	x	variables
Formulae	$\mathscr{A}$	$::=$	$P[\tau_1, \ldots, \tau_n]$	predicates
		$\mid$	$\top$	top (true)
		$\mid$	$\bot$	bottom (false)
		$\mid$	$\mathscr{P} \wedge \mathscr{Q}$	conjunction
		$\mid$	$\mathscr{P} \vee \mathscr{Q}$	disjunction
		$\mid$	$\mathscr{P} \Rightarrow \mathscr{Q}$	implication
		$\mid$	$\neg\mathscr{P}$	negation
		$\mid$	$\forall x.\mathscr{P}[x]$	universal quantification
		$\mid$	$\exists x.\mathscr{P}[x]$	existential quantification

Universal Quantifier

Universal Quantifier

m	$\quad a$	
n	$\quad \mathscr{P}[a/x]$	
l	$\forall x.\mathscr{P}[x]$	$\forall$I, m–n

Universal Quantifier

$$
\begin{array}{l|l l}
m & \quad \begin{array}{|l} a \\ \hline \mathscr{P}[a/x] \end{array} & \\
n & & \\
l & \forall x.\mathscr{P}[x] & \forall\text{I},\ m\text{–}n
\end{array}
$$

$$
\begin{array}{l|l l}
m & \forall x.\mathscr{P}[x] & \\
n & \mathscr{P}[a/x] & \forall\text{E},\ m
\end{array}
$$

Existential Quantifier

Existential Quantifier

m	$\mathscr{P}[a]$	
n	$\exists x.\mathscr{P}[x/a]$	$\exists$I, m

Existential Quantifier

$$
\begin{array}{l|ll}
m & \mathscr{P}[a] & \\
n & \exists x.\mathscr{P}[x/a] & \exists\text{I},\ m
\end{array}
$$

$$
\begin{array}{l|ll}
m & \exists x.\mathscr{P}[x] & \\
i & \quad \mathscr{P}[a/x] & \\
j & \quad \mathscr{A} & \\
n & \mathscr{A} & \exists\text{E},\ m,\ i\text{–}j
\end{array}
$$

Subsection 1

Examples

$\forall x.Fx \Rightarrow Gx, \exists x.Fx \vdash \exists x.Gx$

1	$\forall x.Fx \Rightarrow Gx$	
2	$\exists x.Fx$	
3	$\quad Fa$	
4	$\quad Fa \Rightarrow Ga$	$\forall$E, 1
5	$\quad Ga$	$\Rightarrow$E, 4, 3
6	$\quad \exists x.Gx$	$\exists$I, 5
7	$\exists x.Gx$	$\exists$E, 2, 3–5

Section 5

“Upgrading” to Classical Logic

Excluded Middle (and Related Axioms)

i	$\quad\mathscr{A}$		
j	$\quad\mathscr{B}$		
k	$\quad\neg\mathscr{A}$		
l	$\quad\mathscr{B}$		
m	$\mathscr{B}$	LEM, i–j, k–l	

m	$\neg\neg\mathscr{A}$	
n	$\mathscr{A}$	DNE, m

m	$\quad\neg\mathcal{A}$	
n	$\quad\bot$	
l	$\mathcal{A}$	IP, m–n

Section 6

Exercises

$\mathbf{LEM} \vdash \mathbf{DNE}$

LEM $\vdash$ DNE

1	$\neg\neg P$	
2	$\quad P$	
3	$\quad P$	R, 2
4	$\quad \neg P$	
5	$\quad \bot$	$\neg$E, 1, 4
6	$\quad P$	$\bot$E, 5
7	P	LEM, 2–3, 4–6

$\mathbf{DNE} \vdash \mathbf{IP}$

DNE $\vdash$ IP

1	$\neg P \Rightarrow \bot$	
2	$\quad \neg P$	
3	$\quad \bot$	$\Rightarrow$E, 1, 2
4	$\neg\neg P$	$\neg$I, 2–3
5	P	DNE, 4

$\mathbf{IP} \vdash \mathbf{LEM}$

IP $\vdash$ LEM

1	$\neg(P \lor \neg P)$	
2	$\neg P$	
3	$P \lor \neg P$	$\lor$I, 2
4	$\bot$	$\neg$E, 1, 3
5	P	IP, 2–4
6	$P \lor \neg P$	$\lor$I, 5
7	$\bot$	$\neg$E, 1, 6
8	$P \lor \neg P$	IP, 1–7

$$\forall x.\neg Mx \vee Ljx, \forall x.Bx \Rightarrow Ljx, \forall x.Mx \vee Bx \vdash \forall x.Ljx$$

$$\forall x.\neg Mx \lor Ljx, \forall x.Bx \Rightarrow Ljx, \forall x.Mx \lor Bx \vdash \forall x.Ljx$$

1	$\forall x.\neg Mx \lor Ljx$	
2	$\forall x.Bx \Rightarrow Ljx$	
3	$\forall x.Mx \lor Bx$	
4	$\neg Ma \lor Lja$	$\forall$E, 1
5	$Ba \Rightarrow Lja$	$\forall$E, 2
6	$Ma \lor Ba$	$\forall$E, 3
7	Lja	
8	Lja	R, 7
9	$\neg Ma$	
10	Ma	
11	$\bot$	$\neg$E, 9, 10
12	Lja	$\bot$E, 11
13	Ba	
14	Lja	$\Rightarrow$E, 5, 13
15	Lja	$\lor$E, 3, 10–12, 13–14
16	Lja	$\lor$E, 1, 9–15, 7–8
17	$\forall x.Ljx$	$\forall$I, 16

Questions?

There's this thing which I like to call the . . . "Holy Trinity" here of computer science[:] . . . the correspondences between proof theory, which is . . . the theory of logic and proofs; algebra and category theory; and then the subject in the computer science side[, which are] programs, or type theory.

— Robert Harper (2012)

Bibliography

- Open Logic Project. 2023. *forall x: Calgary*. https://forallx.openlogicproject.org/.
- Frank Pfenning. 2023. *Lecture Notes on What is (Constructive) Logic?*. https://www.cs.cmu.edu/~fp/courses/15317-s23/lectures/01-whatis.pdf.
- Samuel Buss. 1998. *An Introduction to Proof Theory*. https://mathweb.ucsd.edu/~sbuss/ResearchWeb/handbookI/ChapterI.pdf.
- Pr∞f Wiki. 2024. *Pr∞f Wiki*. https://proofwiki.org/wiki/Definition:Proof_System.