Introduction to Proof Theory

Lecture 1
Proof Theory and Proof Systems

Marianna Girlando, Sonia Marin
Universtiy of Amsterdam, University of Birmingham

OPLSS 2023
Eugene, Oregon, June 26 - July 8, 2023
Outline

Introduction

Propositional and first order syntax

Proof systems
Some history

1. Formalize statements/proofs
2. Prove consistency ✓
3. Independent & Completeness ✓
4. Decision problem X

Hilbert

Frege Peano Russell

1879 1890 1906

1928 1929

Classical predicate logic

H & Ackermann

1930

1935

Church undecidability

social

proof evidence

object

formalization

consistent

meta

eta
Outline

1. History: Why it started?
   Formal language & First proof system
   Classical logic & Arithmetic

2. Intuitionistic / Constructive
   Related to prog lang theory

3. Natural Deduction
   Sequent calculus

4. Cut Elimination / Normalization

5. Back to arithmetic
   Etc.
Outline

Introduction

Propositional and first order syntax

Proof systems
Propositional logic: syntax

Language

- Countably many propositional variables:
  \[ \text{Var}_\rho = \{ p, q, r, \ldots \} \]

- Propositional constants: \( \bot \) (false)

- Connectives: \( \lor \) (disjunction), \( \land \) (conjunction), \( \rightarrow \) (implication)

Formulas (Form\(_\rho\)) \( A, B, C, \ldots \) are inductively generated as follows:

- Propositional variables and constants are formulas
- If \( A, B \) are formulas then \( A \lor B, A \land B, A \rightarrow B \) are formulas.

\[ \neg A := A \rightarrow \bot \quad \top := p \lor (p \rightarrow \bot) \]
How do we interpret propositional formulas?

- **Propositional assignment**: assigns \{0, 1\} to propositional variables \(\alpha : \text{Var}_p \rightarrow \{0, 1\}\)

- Extend the assignment to formulas

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>A \land B</th>
<th></th>
<th></th>
<th>A \lor B</th>
<th></th>
<th></th>
<th>A \rightarrow B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Equivalently: Define \(\alpha \models A\) “\(\alpha\) satisfies \(A\)”

- \(\alpha \not\models \bot\)
- \(\alpha \models p\) iff \(\alpha(p) = 1\)
- \(\alpha \models A \land B\) iff \(\alpha \models A\) and \(\alpha \models B\)
- \(\alpha \models A \lor B\) iff \(\alpha \models A\) or \(\alpha \models B\)
- \(\alpha \models A \rightarrow B\) iff \(\alpha \not\models A\) or \(\alpha \models B\)
Predicate logic: language

We define a **predicate language** $\mathcal{L}^= \equiv$ as follows:

- **Countably many variables**: $\text{Var} = \{x, y, z, \ldots\}$
- **A set of function symbols**: $\text{Fun} = \{f, g, h, \ldots\}$
  
  Each function symbol has a fixed *arity* (n of arguments it takes)
  
  0-ary function symbols are called *constants*

- **A set of predicate symbols**: $\text{Pred} = \{P, Q, R, \ldots\}$
  
  Each predicate symbol has a fixed *arity* (n of arguments it takes)
  
  Propositional variables are 0-ary predicates

- **The equality symbol** $= \ (2\text{-ary predicate})$

- **Propositional constants**: $\bot$

- **Connectives** $\lor, \land, \rightarrow$.

- **Quantifiers**: $\exists \ (\text{existential})$ and $\forall \ (\text{universal})$
Predicate logic: terms

Terms (Ter) $s, t, u, \ldots$ are inductively generated as follows:

- Variables are terms
- If $f \in \text{Fun}$ is a $k$-ary function symbol and $t_1, \ldots, t_k$ are terms, then the following is a term:

$$f(t_1, \ldots, t_k)$$

Any constant is a term.

Informally, terms denote individual entities.
Predicate logic: formulas

Atomic formulas $P(t_1, \ldots, t_k)$ are inductively generated as follows:

- If $s, t$ are terms, then $s = t$ is an atomic formula.
- If $P$ is a predicate symbol or arity $k$ and $t_1, \ldots, t_k$ are terms, then the following is an atomic formula:

  $$P(t_1, \ldots, t_k)$$

Formulas (Form) $P, Q, R, \ldots$ are inductively generated as follows:

- Atomic formulas are formulas
- $\bot$ is a formula
- If $A, B$ are formulas then $A \lor B$, $A \land B$ and $A \rightarrow B$ are formulas
- If $A$ is a formula then $\exists x A$ and $\forall x A$ are formulas.
Outline

Introduction

Propositional and first order syntax

Proof systems
Proof systems, informally

A proof system consists of:

- Set of axioms;
- Set of inference rules.

A proof of a formula $A$ is constructed by chaining together axioms, inference rules, and objects generated from axioms and inference rules, until $A$ is reached.

A logic can be identified with the set of provable formulas.
Various kinds of proof systems

▷ Hilbert-Frege proof systems, or axiom systems, or reductive systems (Prawitz, 1971)
▷ Gentzen-style proof systems

Today:
▷ Axiom system for propositional logic
▷ Axiom system for first-order logic
▷ First-order theories and Peano Arithmetic
An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

$A, B, C$ formulas of $\mathcal{L}_p$

PL1. $A \rightarrow (B \rightarrow A)$
PL2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
PL3. $(A \land B) \rightarrow A$
PL4. $(A \land B) \rightarrow B$
PL5. $A \rightarrow (B \rightarrow (A \land B))$
PL6. $A \rightarrow (A \lor B)$
PL7. $B \rightarrow (A \lor B)$
PL8. $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \lor B) \rightarrow C))$
PL9. $\bot \rightarrow A$
PL10. $A \lor (A \rightarrow \bot)$

\[
\begin{array}{c}
\text{mp} \quad \frac{A \quad A \rightarrow B}{B}
\end{array}
\]
Prove the following:

\[ \vdash_{H_{cp}} \overline{A \rightarrow A} \]

\[ \{A \rightarrow B, B \rightarrow C\} \vdash_{H_{cp}} A \rightarrow C \]

**PL1.**  \[ A \rightarrow (B \rightarrow A) \]

**PL2.**  \[ (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \]

\[
\begin{align*}
A \quad A \rightarrow B \\
\text{mp} & \quad B \\
\hline
A \rightarrow ((B \rightarrow A) \rightarrow A)
\end{align*}
\]

\[
\begin{align*}
\text{1} & \quad A \rightarrow ((B \rightarrow A) \rightarrow A) \\
\text{2} & \quad (A \rightarrow ((B \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \\
\text{3} & \quad (A \rightarrow (B \rightarrow A)) \rightarrow (A \rightarrow A) \\
\text{4} & \quad A \rightarrow (B \rightarrow A) \\
\text{5} & \quad A \rightarrow A
\end{align*}
\]

\[ \text{mp} 1, 2 \]

\[ \text{PL1} \]

\[ \text{mp} 3, 4 \]
Proofs in $\mathcal{H}_{cp}$

For $A$ formula of $\mathcal{L}_p$, $\Gamma$ set of formulas of $\mathcal{L}_p$:

A $\mathcal{H}_{cp}$ derivation of $A$ from assumptions in $\Gamma$ is a list of $\mathcal{L}_p$ formulas

$$
A_1, A_2, \ldots, A_n
$$

where $A_n = A$ and for each $A_i$, for $i \leq n$, we have that either:

- $A_i$ is an axiom of $\mathcal{H}_{cp}$;
- $A_i \in \Gamma$;
- $A_i$ is obtained by applying (mp) to formulas in $A_1, \ldots, A_{i-1}$.

We write $\Gamma \vdash_{\mathcal{H}_{cp}} A$ if there is a derivation of $A$ from formulas in $\Gamma$.

A proof of $A$ is a derivation of $A$ from $\emptyset$. We write $\vdash_{\mathcal{H}_{cp}} A$ if there is a proof of $A$.

Classical propositional logic CPL is defined as $\{ A | \vdash_{\mathcal{H}_{cp}} A \}$.
Deduction Theorem

For a formula of $\mathcal{L}_p$, $\Gamma$ set of formulas of $\mathcal{L}_p$:

$\not\Gamma \vdash_{H_{cp}} A \rightarrow B \iff \Gamma \cup \{A\} \vdash_{H_{cp}} B$

\[
\Gamma \\
\{ : \} \\
A \rightarrow B
\]

$\Gamma \cup \{A\}$

$\Gamma \cup \{B\}$

$\Rightarrow$

$\Gamma$

\[
\{ : \} \\
A \rightarrow B
\]

$\Rightarrow$ RanCosu, Galvan, Zach

An Introduction to Proof Theory