Hi

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

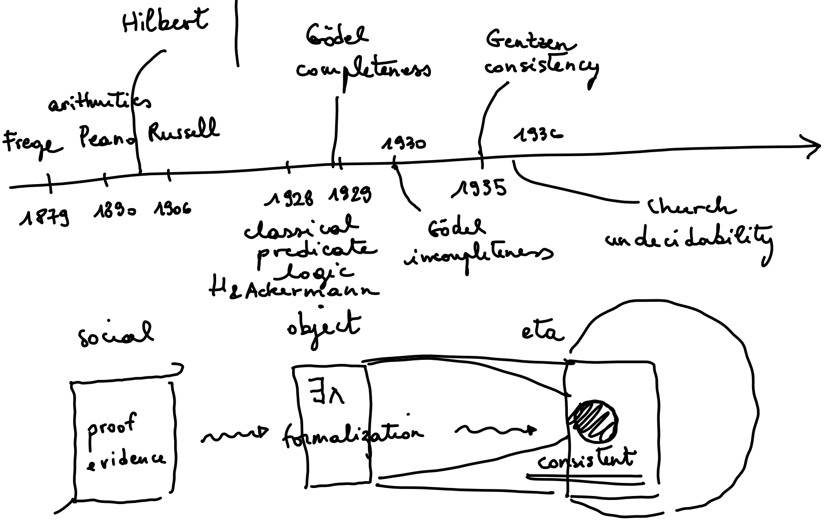
Introduction

Propositional and first order syntax

Proof systems

Some history

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



Some history

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 arithmetics

   Etc.

Outline

Introduction

Propositional and first order syntax

Proof systems

Propositional logic: syntax

Propositional logic: syntax

Language

Propositional logic: syntax

Language

- Countably many propositional variables:

$$\mathrm{Var}_p = \{p, q, r, \dots\}$$

Propositional logic: syntax

Language

- Countably many propositional variables:

$$\mathsf{Var}_p = \{p, q, r, \ldots\}$$

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

Propositional logic: syntax

Language

- Countably many propositional variables:

$$\mathrm{Var}_p = \{p, q, r, \ldots\}$$

- Propositional constants: $\bot$ (*false*)
- Connectives: $\vee$ (*disjunction*), $\wedge$ (*conjunction*), $\rightarrow$ (*implication*)

Propositional logic: syntax

Language

- Countably many propositional variables:

$$\mathsf{Var}_p = \{p, q, r, \ldots\}$$

- Propositional constants: $\perp$ (*false*)
- Connectives: $\vee$ (*disjunction*), $\wedge$ (*conjunction*), $\rightarrow$ (*implication*)

Formulas (Form_p) $A, B, C, \ldots$ are inductively generated as follows:

Propositional logic: syntax

Language

- Countably many propositional variables:

$$\mathsf{Var}_p = \{p, q, r, \dots\}$$

- Propositional constants: $\bot$ (*false*)
- Connectives: $\vee$ (*disjunction*), $\wedge$ (*conjunction*), $\rightarrow$ (*implication*)

Formulas (Form_p) $A, B, C, \dots$ are inductively generated as follows:

- Propositional variables and constants are formulas

Propositional logic: syntax

Language

- Countably many propositional variables:

$$\mathrm{Var}_p = \{p, q, r, \dots\}$$

- Propositional constants: $\bot$ (*false*)
- Connectives: $\vee$ (*disjunction*), $\wedge$ (*conjunction*), $\rightarrow$ (*implication*)

Formulas (Form_p) $A, B, C, \dots$ are inductively generated as follows:

- Propositional variables and constants are formulas
- If A, B are formulas then $A \vee B$, $A \wedge B$, $A \rightarrow B$ are formulas.

$\top := p \vee \bar{p} \qquad \bot := p \wedge \bar{p}$

How do we interpret propositional formulas?

How do we interpret propositional formulas?

- Propositional assignment: assigns $\{0, 1\}$ to propositional variables

$$\alpha : \mathsf{Var}_p \rightarrow \{0, 1\}$$

How do we interpret propositional formulas?

- Propositional assignment: assigns $\{0, 1\}$ to propositional variables

$$\alpha : \mathsf{Var}_p \rightarrow \{0, 1\}$$

- Extend the assignment to formulas

How do we interpret propositional formulas?

- Propositional assignment: assigns $\{0, 1\}$ to propositional variables

$$\alpha : \mathsf{Var}_p \to \{0, 1\}$$

- Extend the assignment to formulas

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

A	B	$A \to B$
1	1	1
1	0	0
0	1	1
0	0	1

How do we interpret propositional formulas?

- Propositional assignment: assigns $\{0, 1\}$ to propositional variables

$$\alpha : \mathrm{Var}_p \rightarrow \{0, 1\}$$

- Extend the assignment to formulas

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

Equivalently: Define $\alpha \models A$ "α satisfies A"

$\alpha \not\models \bot$

$\alpha \models p$ *iff* $\alpha(p) = 1$

$\alpha \models A \wedge B$ *iff* $\alpha \models A$ and $\alpha \models B$

$\alpha \models A \vee B$ *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 $\mathcal{L}^=$ as follows:

Predicate logic: language

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

- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: $\mathrm{Var} = \{x, y, z, \ldots\}$

- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: Var $= \{x, y, z, \ldots\}$
- A set of function symbols: Fun $= \{f, g, h, \ldots\}$

- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: Var $= \{x, y, z, \ldots\}$
- A set of function symbols: Fun $= \{f, g, h, \ldots\}$

  Each function symbol has a fixed *arity* (n of arguments it takes)

- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: Var $= \{x, y, z, \ldots\}$
- A set of function symbols: Fun $= \{f, g, h, \ldots\}$

  Each function symbol has a fixed *arity* (n of arguments it takes)

  0-ary function symbols are called constants

- Propositional constants: $\perp$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: $\mathrm{Var} = \{x, y, z, \ldots\}$
- A set of function symbols: $\mathrm{Fun} = \{f, g, h, \ldots\}$

  Each function symbol has a fixed *arity* (n of arguments it takes)

  0-ary function symbols are called constants
- A set of predicate symbols: $\mathrm{Pred} = \{P, Q, R, \ldots\}$
- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: Var $= \{x, y, z, \ldots\}$
- A set of function symbols: 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: Pred $= \{P, Q, R, \ldots\}$

  Each predicate symbol has a fixed *arity* (n of arguments it takes)
- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: Var $= \{x, y, z, \ldots\}$
- A set of function symbols: 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: Pred $= \{P, Q, R, \ldots\}$

  Each predicate symbol has a fixed *arity* (n of arguments it takes)

  Propositional variables are 0-ary predicates
- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

We define a predicate language $\mathcal{L}^{=}$ 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-ary predicate)
- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.

Predicate logic: language

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

- Countably many variables: Var $= \{x, y, z, \ldots\}$
- A set of function symbols: 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: 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-ary predicate)
- Propositional constants: $\bot$
- Connectives $\vee, \wedge, \rightarrow$.
- Quantifiers: $\exists$ (*existential*) and $\forall$ (*universal*)

Predicate logic: terms

Predicate logic: terms

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

Predicate logic: terms

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

- Variables are terms

Predicate logic: terms

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

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

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

Predicate logic: terms

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

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

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

Any constant is a term.

Predicate logic: terms

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

- Variables are terms
- If $f \in \mathrm{Fun}$ 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

Predicate logic: formulas

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

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.

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:

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

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:

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

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

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
- $\perp$ is a formula
- If A, B are formulas then $A \vee B$, $A \wedge B$ and $A \rightarrow B$ are formulas

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 $A \vee B$, $A \wedge B$ 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

Proof systems, informally

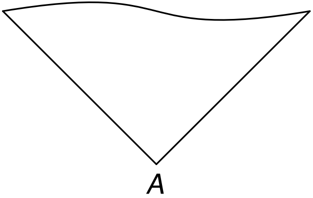
A proof system consists of:

- Set of axioms;
- Set of inference rules.

Proof systems, informally

A proof system consists of:

- Set of axioms;
- Set of inference rules.

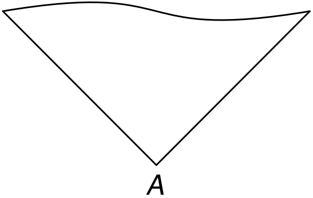


A proof of a formula A is constructed by chaining together axioms, inference rules, and objects generated from axioms and inference rules, until A is reached.

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

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: $\mathcal{H}_{cp}$

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

PL1. $A \to (B \to A)$

PL2. $\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

PL1. $A \to (B \to A)$

PL2. $\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$

PL3. $(A \wedge B) \to A$

PL4. $(A \wedge B) \to B$

PL5. $A \to \big(B \to (A \wedge B)\big)$

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

PL1.	$A \to (B \to A)$
PL2.	$\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$
PL3.	$(A \wedge B) \to A$
PL4.	$(A \wedge B) \to B$
PL5.	$A \to \big(B \to (A \wedge B)\big)$
PL6.	$A \to (A \vee B)$
PL7.	$B \to (A \vee B)$
PL8.	$(A \to C) \to \big((B \to C) \to \big((A \vee B) \to C\big)\big)$

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

PL1.	$A \to (B \to A)$
PL2.	$\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$
PL3.	$(A \wedge B) \to A$
PL4.	$(A \wedge B) \to B$
PL5.	$A \to \big(B \to (A \wedge B)\big)$
PL6.	$A \to (A \vee B)$
PL7.	$B \to (A \vee B)$
PL8.	$(A \to C) \to \big((B \to C) \to \big((A \vee B) \to C\big)\big)$
PL9.	$\bot \to A$

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

PL1. $A \to (B \to A)$

PL2. $\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$

PL3. $(A \wedge B) \to A$

PL4. $(A \wedge B) \to B$

PL5. $A \to \big(B \to (A \wedge B)\big)$

PL6. $A \to (A \vee B)$

PL7. $B \to (A \vee B)$

PL8. $(A \to C) \to \big((B \to C) \to \big((A \vee B) \to C\big)\big)$

PL9. $\bot \to A$

PL10. $A \vee (A \to \bot)$

An axiom system for classical propositional logic: $\mathcal{H}_{cp}$

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

PL1.	$A \to (B \to A)$
PL2.	$\big(A \to (B \to C)\big) \to \big((A \to B) \to (A \to C)\big)$
PL3.	$(A \wedge B) \to A$
PL4.	$(A \wedge B) \to B$
PL5.	$A \to \big(B \to (A \wedge B)\big)$
PL6.	$A \to (A \vee B)$
PL7.	$B \to (A \vee B)$
PL8.	$(A \to C) \to \big((B \to C) \to \big((A \vee B) \to C\big)\big)$
PL9.	$\bot \to A$
PL10.	$A \vee (A \to \bot)$

$$\text{mp}\ \frac{A \qquad\qquad A \to B}{B}$$

Examples

Prove the following:

- ☞ $\vdash_{\mathcal{H}_{cp}} A \to A$
- ☞ $\{A \to B, B \to C\} \vdash_{\mathcal{H}_{cp}} A \to C$

Examples

Prove the following:

- ☞ $\vdash_{\mathcal{H}_{cp}} A \rightarrow A$
- ☞ $\{A \rightarrow B, B \rightarrow C\} \vdash_{\mathcal{H}_{cp}} A \rightarrow C$

$$\begin{array}{ll} \text{PL1.} & A \rightarrow (B \rightarrow A) \\ \text{PL2.} & \big(A \rightarrow (B \rightarrow C)\big) \rightarrow \big((A \rightarrow B) \rightarrow (A \rightarrow C)\big) \end{array}$$

$$\text{mp}\ \frac{A \qquad\qquad A \rightarrow B}{B}$$

Proofs in $\mathcal{H}_{cp}$

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

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

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

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 Γ.

Proofs in $\mathcal{H}_{cp}$

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

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

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

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 Γ.

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.

Proofs in $\mathcal{H}_{cp}$

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

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

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

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 Γ.

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 $\{A \mid \vdash_{\mathcal{H}_{cp}} A\}$.

Deduction Theorem

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

$$\Gamma \vdash_{\mathcal{H}_{cp}} A \to B \quad \text{iff} \quad \Gamma \cup \{A\} \vdash_{\mathcal{H}_{cp}} B$$

Deduction Theorem

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

$$\Gamma \vdash_{\mathcal{H}_{cp}} A \to B \quad \text{iff} \quad \Gamma \cup \{A\} \vdash_{\mathcal{H}_{cp}} B$$

$$\begin{array}{c} \Gamma \\ \vdots \\ A \to B \end{array} \quad \Longleftrightarrow \quad \begin{array}{c} \Gamma \cup \{A\} \\ \vdots \\ B \end{array}$$

An axiom system for first-order logic: $\mathcal{H}_{fo}$

An axiom system for first-order logic: $\mathcal{H}_{fo}$

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

An axiom system for first-order logic: $\mathcal{H}_{fo}$

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

Axioms and inference rules of $\mathcal{H}_{cp}$, plus:

An axiom system for first-order logic: $\mathcal{H}_{fo}$

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

Axioms and inference rules of $\mathcal{H}_{cp}$, plus:

FO1. $\forall x(A(x)) \to A(t)$
FO2. $A(t) \to \exists x(A(x))$

An axiom system for first-order logic: $\mathcal{H}_{fo}$

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

Axioms and inference rules of $\mathcal{H}_{cp}$, plus:

FO1. $\forall x(A(x)) \rightarrow A(t)$

FO2. $A(t) \rightarrow \exists x(A(x))$

FO3. $\forall x(A \rightarrow B(x)) \rightarrow \big(A \rightarrow \forall x(B(x))\big)$ where $x \notin FV(A)$

FO4. $\forall x(A(x) \rightarrow B) \rightarrow \big(\exists x(A(x)) \rightarrow B\big)$ where $x \notin FV(B)$

An axiom system for first-order logic: $\mathcal{H}_{fo}$

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

Axioms and inference rules of $\mathcal{H}_{cp}$, plus:

FO1. $\forall x(A(x)) \rightarrow A(t)$

FO2. $A(t) \rightarrow \exists x(A(x))$

FO3. $\forall x(A \rightarrow B(x)) \rightarrow \big(A \rightarrow \forall x(B(x))\big)$ where $x \notin FV(A)$

FO4. $\forall x(A(x) \rightarrow B) \rightarrow \big(\exists x(A(x)) \rightarrow B\big)$ where $x \notin FV(B)$

FO4. $\forall x(x = x)$

FO5. $\forall x \forall y\big(x = x \rightarrow (A(x) \rightarrow A(y))\big)$

$$\text{gen}\ \frac{A}{\forall x(A)}$$

An axiom system for first-order logic: $\mathcal{H}_{fo}$

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

Axioms and inference rules of $\mathcal{H}_{cp}$, plus:

FO1.	$\forall x(A(x)) \to A(t)$	
FO2.	$A(t) \to \exists x(A(x))$	
FO3.	$\forall x(A \to B(x)) \to \big(A \to \forall x(B(x))\big)$	where $x \notin FV(A)$
FO4.	$\forall x(A(x) \to B) \to \big(\exists x(A(x)) \to B\big)$	where $x \notin FV(B)$
FO4.	$\forall x(x = x)$	
FO5.	$\forall x \forall y\big(x = x \to (A(x) \to A(y))\big)$	

$$\text{gen}\ \frac{A}{\forall x(A)}$$

Prove the following:

☞ $\{\forall x(A \to B), \forall x(A)\} \vdash_{\mathcal{H}_{fo}} \forall x(B)$

Proofs in $\mathcal{H}_{fo}$

A formula of $\mathcal{L}^{=}$, Γ set of formulas of $\mathcal{L}^{=}$:

A $\mathcal{H}_{fo}$ derivation of A from assumptions in Γ is a list of $\mathcal{L}^{=}$ formulas

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

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

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

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

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

First-order logic FOL is defined as $\{A \mid \vdash_{\mathcal{H}_{fo}} A\}$.

Peano Arithmetic (PA)

Peano Arithmetic (PA)

A first-order theory $\mathcal{T}$ consists of:

- A predicate language $\mathcal{L}^{=}_{\mathcal{T}}$;
- A set of formulas in the language $\mathcal{L}^{=}_{\mathcal{T}}$, the non-logical axioms of the theory.

Peano Arithmetic (PA)

A first-order theory $\mathcal{T}$ consists of:

- A predicate language $\mathcal{L}^{=}_{\mathcal{T}}$;
- A set of formulas in the language $\mathcal{L}^{=}_{\mathcal{T}}$, the non-logical axioms of the theory.

Peano Arithmetic (PA) is a first-order theory

$$\mathcal{L}^{=}_{PA} = \{0, s, +, \cdot\}$$

Peano Arithmetic (PA)

A first-order theory $\mathcal{T}$ consists of:

- A predicate language $\mathcal{L}^{=}_{\mathcal{T}}$;
- A set of formulas in the language $\mathcal{L}^{=}_{\mathcal{T}}$, the non-logical axioms of the theory.

Peano Arithmetic (PA) is a first-order theory

$$\mathcal{L}^{=}_{PA} = \{0, s, +, \cdot\}$$

PA1. $\forall x \neg(0 = s(x))$

PA2. $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$

PA3. $\forall x (x + 0 = x)$

PA4. $\forall x \forall y (x + s(y) = s(x + y))$

PA5. $\forall x (x \cdot 0 = 0)$

PA6. $\forall x \forall y (x \cdot s(y) = (x \cdot y) + x)$

PA7. $\big(A(0) \wedge \forall x (A(x) \rightarrow A(s(x)))\big) \rightarrow \forall x (A(x))$
where $A \in \mathcal{L}^{=}_{PA}, x \in FV(A)$

Discussion