Introduction to Proof Theory

Lecture 1 - First-order logic: syntax and semantics

Anupam Das

University of Birmingham

21st Midlands Graduate School
in Foundations of Computing Science

Sheffield University (virtual)
12 April 2021

These slides are available at `http://www.anupamdas.com/mgs21`.
Based on slides from ESSLLI'18, prepared with Thomas Powell.

Outline

Wherefore proof theory?

Proof theory is the study of mathematical proofs as formal objects.

Wherefore proof theory?

Proof theory is the study of mathematical proofs as formal objects.

Formally, a **proof system** defines what a proof may be. This allows not only the study of what is provable, but also of proofs themselves.

Wherefore proof theory?

Proof theory is the study of mathematical proofs as formal objects.

Formally, a **proof system** defines what a proof may be. This allows not only the study of what is provable, but also of proofs themselves.

But *why* do we study proofs?

Wherefore proof theory?

Proof theory is the study of mathematical proofs as formal objects.

Formally, a **proof system** defines what a proof may be. This allows not only the study of what is provable, but also of proofs themselves.

But *why* do we study proofs?

- *Foundational results*, which tell us something fundamental about reasoning itself. These lead to...
- *Applications*, which use insights and techniques from proof theory to accomplish something concrete in another discipline altogether.

Wherefore proof theory?

Proof theory is the study of mathematical proofs as formal objects.

Formally, a **proof system** defines what a proof may be. This allows not only the study of what is provable, but also of proofs themselves.

But *why* do we study proofs?

- *Foundational results*, which tell us something fundamental about reasoning itself. These lead to…
- *Applications*, which use insights and techniques from proof theory to accomplish something concrete in another discipline altogether.

This lecture course: An introduction to the field, focussing on classical results but hinting at powerful applications.

Foundational results from the early days

Gödel's incompleteness theorem (1931), informally

We cannot prove every true sentence in a given proof system.

This result fundamentally reoriented the direction of logic research in the 20th century.

Foundational results from the early days

Gödel's incompleteness theorem (1931), informally

We cannot prove every true sentence in a given proof system.

This result fundamentally reoriented the direction of logic research in the 20th century.

Gentzen's *Hauptsatz* (1934), informally

Every purely logical theorem can be proved analytically, i.e. without making 'guesses'.

The translation of proofs to analytic form, cut-elimination, has become one of the most powerful tools in all of logic.

Proof theory and computation

Proof theory plays an essential role in understanding computation:

- Extracting *computational content* via normal forms. (cut-elimination, normalisation,...)
- ...and by proof interpretations. (realisability, Dialectica, witnessing,...)

- Modelling non-deterministic complexity (proof complexity, proof search,...)
- Proof normalisation as computation. (Curry-Howard-Lambek,...)
- Proof search as computation. (focussing, logic programming,...)

References for this course

Proof theory is an incredibly big field (both deep and broad). The following are popular broad-spectrum references:

1. [Buss, 1998] *Handbook of Proof Theory* (a mathematical approach)
2. [Troelstra and Schwichtenberg, 1996] *Basic Proof Theory*. (a computational approach)
3. [Schoenfeld, 1967] *Mathematical Logic*. (a broader overview of logic)
4. *Stanford Encyclopedia of Philosophy*. (an excellent high-level general reference) `http://plato.stanford.edu/`

NB: While 1, 2 and 3 are a little dated in terms of topics, the material still remains standard initiation to proof theory.

Sprechen Sie deutsch?

I highly recommend taking a look at original texts:

- [Frege, 1879] *Begriffsschrift*. (the first serious attempt to formalise mathematics in logic, unfortunately erroneously)
- [Whitehead and Russell, 1927] *Principia Mathematica*. (the second serious attempt, notably error-free)
- [Gödel, 1931] *Über Formal Unentscheidbare Sätze der Principia Mathematica Und Verwandter Systeme I*. (the incompleteness theorems)
- [Szabo, 1972] *The collected papers of Gerhard Gentzen*. English translations. (the foundation of structural proof theory)

Outline

The basic syntax

A **language** is given by the following:

- A set Var of *variables*, $x, y, z, \ldots$ (we assume we have infinitely many)
- A set Fun of *function* symbols $f, g, h, \ldots$. Each function symbol f has a fixed arity $\alpha(f)$, which is just a fancy name for the number of arguments it takes.
- A set Rel of *relation* or *predicate* symbols $P, Q, R, S, \ldots$, where each relation symbol P also has a fixed arity $\alpha(P)$.

The basic syntax

A **language** is given by the following:

- A set Var of *variables*, $x, y, z, \ldots$ (we assume we have infinitely many)
- A set Fun of *function* symbols $f, g, h, \ldots$. Each function symbol f has a fixed arity $\alpha(f)$, which is just a fancy name for the number of arguments it takes.
- A set Rel of *relation* or *predicate* symbols $P, Q, R, S, \ldots$, where each relation symbol P also has a fixed arity $\alpha(P)$.

We often identify a set Cons $\subseteq$ Fun consisting of those function symbols of arity 0, written $a, b, c, \ldots$. We call these *constants*, as they don't take any arguments.

The basic syntax

A **language** is given by the following:

- A set Var of *variables*, $x, y, z, \ldots$ (we assume we have infinitely many)
- A set Fun of *function* symbols $f, g, h, \ldots$. Each function symbol f has a fixed arity $\alpha(f)$, which is just a fancy name for the number of arguments it takes.
- A set Rel of *relation* or *predicate* symbols $P, Q, R, S, \ldots$, where each relation symbol P also has a fixed arity $\alpha(P)$.

We often identify a set Cons $\subseteq$ Fun consisting of those function symbols of arity 0, written $a, b, c, \ldots$. We call these *constants*, as they don't take any arguments.

In **predicate logic**, we will always have:

- a special symbol $=$ for *equality*;
- the usual symbols of propositional logic $\bot, \vee, \wedge, \to, \neg$;
- existential $\exists$ and universal $\forall$ quantifiers.

Terms

Definition (Terms)

The set Ter of **terms**, written $s, t, u, \ldots$ is defined as follows:

- Any variable x is a term.
- Any constant symbol c is a term.
- If f is a function symbol of arity k and $t_1, \ldots, t_k$ are terms, then so is $f(t_1, \ldots, t_k)$.

Informally, terms are just 'things' (*individuals*).

Definition (Atomic formulas)

The set Atom of **atomic formulas** is defined as follows:

- If s, t are terms, then $s = t$ is a formula.
- If P is a relation symbol or arity k and $t_1, \ldots, t_k$ are terms, then $P(t_1, \ldots, t_k)$ is an atomic formula.

NB: a relation symbol of arity 0 is just a propositional variable.

Formulas

Definition (Atomic formulas)

The set Atom of **atomic formulas** is defined as follows:

- If s, t are terms, then $s = t$ is a formula.
- If P is a relation symbol or arity k and $t_1, \ldots, t_k$ are terms, then $P(t_1, \ldots, t_k)$ is an atomic formula.

NB: a relation symbol of arity 0 is just a propositional variable.

Definition (Formulas)

Finally, the set Form of **formulas**, written $A, B, C, \ldots$ is defined as follows:

- Any atomic formula is a formula.
- $\bot$ is a formula.
- If A, B are formulas then $\neg A$, $A \vee B$, $A \wedge B$ and $A \to B$ are formulas.
- If A is a formula then $\exists x A$ is a formula.
- If A is a formula then $\forall x A$ is a formula.

Formulas

Definition (Atomic formulas)

The set Atom of **atomic formulas** is defined as follows:

- If s, t are terms, then $s = t$ is a formula.
- If P is a relation symbol or arity k and $t_1, \ldots, t_k$ are terms, then $P(t_1, \ldots, t_k)$ is an atomic formula.

NB: a relation symbol of arity 0 is just a propositional variable.

Definition (Formulas)

Finally, the set Form of **formulas**, written $A, B, C, \ldots$ is defined as follows:

- Any atomic formula is a formula.
- $\bot$ is a formula.
- If A, B are formulas then $\neg A$, $A \vee B$, $A \wedge B$ and $A \to B$ are formulas.
- If A is a formula then $\exists x A$ is a formula.
- If A is a formula then $\forall x A$ is a formula.

NB: we can take $\bot$, $\to$ and $\forall$ as primitive propositional connectives, and define the others in terms of these.

(Blank slide)

Free and bound variables

Formally, we define $\mathsf{FV}(A)$ by induction over the structure of A:

$$\begin{aligned}
\mathsf{FV}(x) &:= \{x\} \\
\mathsf{FV}(f(t_1, \ldots, t_k)) = \mathsf{FV}(P(t_1, \ldots, t_k)) &:= \bigcup_{i=1}^{k} \mathsf{FV}(t_i) \\
\mathsf{FV}(s = t) &:= \mathsf{FV}(s) \cup \mathsf{FV}(t) \\
\mathsf{FV}(\neg A) &:= \mathsf{FV}(A) \\
\mathsf{FV}(A \star B) &:= \mathsf{FV}(A) \cup \mathsf{FV}(B) \quad \text{for } \star \in \{\vee, \wedge, \to\} \\
\mathsf{FV}(\forall x.A) = \mathsf{FV}(\exists x.A) &:= \mathsf{FV}(A) \backslash \{x\}
\end{aligned}$$

If $x \in \mathsf{FV}(A)$ then x is **free** in A. Otherwise it is **bound** in A.

Outline

Semantics recap

In propositional logic formulas are interpreted by an assignment:

$$\alpha : \mathsf{Form} \to \{0, 1\}.$$

We say that:

- α **satisfies** some formula A, which we write as $\alpha \vDash A$, if $\alpha(A) = 1$;
- Γ semantically **entails** A, which we write as $\Gamma \vDash A$, if for *every* assignment α, whenever $\alpha(A_i) = 1$ for all $A_i \in \Gamma$ then $\alpha(A)$.
- A is **valid** whenever $\vDash A$.

Semantics recap

In propositional logic formulas are interpreted by an assignment:

$$\alpha : \mathsf{Form} \to \{0, 1\}.$$

We say that:

- α **satisfies** some formula A, which we write as $\alpha \vDash A$, if $\alpha(A) = 1$;
- Γ semantically **entails** A, which we write as $\Gamma \vDash A$, if for *every* assignment α, whenever $\alpha(A_i) = 1$ for all $A_i \in \Gamma$ then $\alpha(A)$.
- A is **valid** whenever $\vDash A$.

In the setting of predicate logic, the role of α is played by a pair of objects:

- a **structure** $\mathcal{D}$, consisting of a domain and interpretations of symbols.
- a **variable assignment** $\sigma : \mathsf{Var} \to D$.

Together, these allow us to define a **valuation** map

$$[-]^{\sigma}_{\mathcal{D}} : \mathsf{Form} \to \{0, 1\}$$

from formulas to truth values.

Structures and variable assignments

Definition (Structure)

A **structure** $\mathcal{D}$ for a first order language consists of

- A non-empty set D called the **domain**
- An interpretation $f_D : D^k \to D$ whenever f is a function symbol of arity k
- An interpretation $P_D : D^k \to \{0, 1\}$ whenever P is a relation symbol of arity k.

Abuse: we often denote a structure simply by D, by which we implicitly mean that D is a domain equipped with interpretations for all function and relation symbols.

Structures and variable assignments

Definition (Structure)

A **structure** $\mathcal{D}$ for a first order language consists of

- A non-empty set D called the **domain**
- An interpretation $f_D : D^k \to D$ whenever f is a function symbol of arity k
- An interpretation $P_D : D^k \to \{0, 1\}$ whenever P is a relation symbol of arity k.

Abuse: we often denote a structure simply by D, by which we implicitly mean that D is a domain equipped with interpretations for all function and relation symbols.

Definition (Variable assignment)

A **variable assignment** for the structure $\mathcal{D}$ is a map $\sigma : \mathsf{Var} \to D$.

Structures and variable assignments

Definition (Structure)

A **structure** $\mathcal{D}$ for a first order language consists of

- A non-empty set D called the **domain**
- An interpretation $f_D : D^k \to D$ whenever f is a function symbol of arity k
- An interpretation $P_D : D^k \to \{0, 1\}$ whenever P is a relation symbol of arity k.

Abuse: we often denote a structure simply by D, by which we implicitly mean that D is a domain equipped with interpretations for all function and relation symbols.

Definition (Variable assignment)

A **variable assignment** for the structure $\mathcal{D}$ is a map $\sigma : \mathsf{Var} \to D$.

Definition (Valuation of terms)

The map $[-]^\sigma_D : \mathsf{Ter} \to D$ is defined as follows:

$$
\begin{aligned}
[x]^\sigma_D &:= \sigma(x) \\
[c]^\sigma_D &:= c_D \\
[f(t_1, \ldots, t_k)]^\sigma_D &:= f_D([t_1]^\sigma_D, \ldots, [t_k]^\sigma_D)
\end{aligned}
$$

Definition (Valuation of formulas)

We extend valuation to a map $[-]_D^\sigma : \mathsf{Form} \to \{0, 1\}$ as follows:

Definition (Valuation of formulas)

We extend valuation to a map $[-]^{\sigma}_{D} : \mathsf{Form} \to \{0, 1\}$ as follows:

$$[P(t_1, \ldots, t_k)]^{\sigma}_{D} := P_D([t_1]^{\sigma}_{D}, \ldots, [t_k]^{\sigma}_{D})$$

$$[s = t]^{\sigma}_{D} := \begin{cases} 1 & \text{if } [s]^{\sigma}_{D} =_D [t]^{\sigma}_{D} \\ 0 & \text{otherwise} \end{cases}$$

$$[\bot]^{\sigma}_{D} := 0$$

$$[A \star B]^{\sigma}_{D} := [A]^{\sigma}_{D} \star [B]^{\sigma}_{D} \quad \text{for } \star \in \{\vee, \wedge, \to\}$$

$$[\exists x.A]^{\sigma}_{D} := \begin{cases} 1 & \text{if there is some } d \in D \text{ such that } [A]^{\sigma[x \mapsto d]}_{D} = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$[\forall x.A]^{\sigma}_{D} := \begin{cases} 1 & \text{if for all } d \in D \text{ we have } [A]^{\sigma[x \mapsto d]}_{D} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Here, $\sigma[x \mapsto d]$ denotes the variable assignment which agrees with σ for all $y \neq x$, and maps x to d.

Satisfiability, semantic entailment, validity

Definition

- A structure $\mathcal{D}$ **satisfies** a formula A if $[A]_D^\sigma = 1$ for **every** variable assignment σ. We write $\mathcal{D} \vDash A$.
- Suppose that Γ is a set of sentences. We say that Γ **semantically entails** A and write $\Gamma \vDash A$ if, for any structure $\mathcal{D}$ such that $\mathcal{D} \vDash A_i$ for all $A_i \in \Gamma$, then $\mathcal{D} \vDash A$.
- A sentence A is **valid** if $\vDash A$.

(Blank slide)

Outline

What is deductive reasoning?

What is deductive reasoning?

We carry out deductive reasoning every day. E.g.

- The sun is out.
- If the sun is out then it is daytime.
- Therefore it is daytime.

What is deductive reasoning?

We carry out deductive reasoning every day. E.g.

- The sun is out.
- If the sun is out then it is daytime.
- Therefore it is daytime.

Deductive reasoning starts with some premisses, and uses a set of **rules** to reach a conclusion. The above deduction is an instance of the following pattern:

1.	p	(first premise)
2.	$p \to q$	(second premise)
3.	q	(conclusion, inferred from 1 and 2)

This is called ***modus ponens***, and is an example of an inference rule.

How does this relate to truth tables?

We can show that our deduction is *valid* by looking at the corresponding semantics:

The sun is out	It is daytime	If the sun is out then it is daytime
0	0	1
0	1	1
1	0	0
1	1	1

How does this relate to truth tables?

We can show that our deduction is *valid* by looking at the corresponding semantics:

The sun is out	It is daytime	If the sun is out then it is daytime
0	0	1
0	1	1
1	0	0
1	1	1

Desideratum 1: Any deductive process should be semantically valid.

Desideratum 2: Any semantically valid entailment should be formally obtainable through a deductive process.

How should a deductive system be designed?

There many different ways of setting up a formal system, depending on the goals at hand. One of the main dichotomies:

rich and expressive vs. **concise and minimal**

How should a deductive system be designed?

There many different ways of setting up a formal system, depending on the goals at hand. One of the main dichotomies:

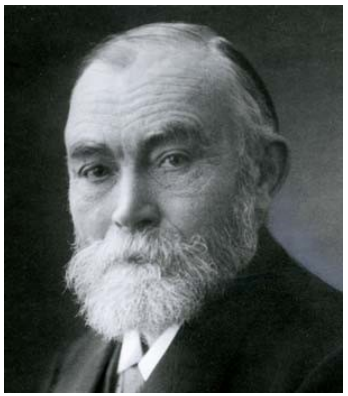
rich and expressive vs. **concise and minimal**

Generally speaking:

- in a **concise and minimal** system, it is easier to prove things about the system but harder to prove things within the system.
- in a **rich and expressive** system, it is easier to prove things within the system but harder to prove things about the system.

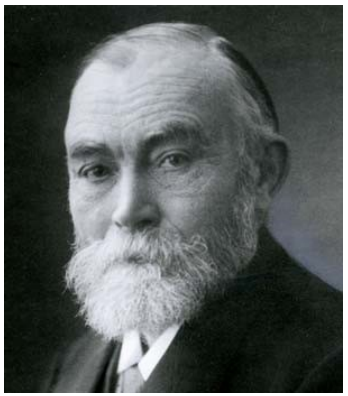
A minimalist system

Hilbert-Frege systems are named after David Hilbert (1862 – 1943) and Gottlob Frege (1848 – 1925).



A minimalist system

Hilbert-Frege systems are named after David Hilbert (1862 – 1943) and Gottlob Frege (1848 – 1925).



Both of them were concerned with proving things about deductive systems, and so they worked in a minimalist setting.

The deductive system

Our system contains the usual axioms and rules for propositional logic, together with some new ones which deal with quantifiers and equality.

The deductive system

Our system contains the usual axioms and rules for propositional logic, together with some new ones which deal with quantifiers and equality.

Definition (Axioms and rules of $\mathcal{F}$)

$\mathcal{F}$ has the following **axioms**:

- $A \to (B \to A)$
- $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
- $\neg\neg A \to A$

The deductive system

Our system contains the usual axioms and rules for propositional logic, together with some new ones which deal with quantifiers and equality.

Definition (Axioms and rules of $\mathcal{F}$)

$\mathcal{F}$ has the following **axioms**:

- $A \to (B \to A)$
- $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
- $\neg\neg A \to A$
- $\forall x\, A \to A[t/x]$
- $\forall x(A \to B) \to A \to \forall x\, B$ where $x \notin \mathsf{FV}(A)$
- $\forall x(x = x)$
- $\forall x, y(x = y \to A \to A[y/x])$

The deductive system

Our system contains the usual axioms and rules for propositional logic, together with some new ones which deal with quantifiers and equality.

Definition (Axioms and rules of $\mathcal{F}$)

$\mathcal{F}$ has the following **axioms**:

- $A \to (B \to A)$
- $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
- $\neg\neg A \to A$
- $\forall x\, A \to A[t/x]$
- $\forall x(A \to B) \to A \to \forall x\, B$ where $x \notin \mathsf{FV}(A)$
- $\forall x(x = x)$
- $\forall x, y(x = y \to A \to A[y/x])$

$\mathcal{F}$ also has two **inference rules**, namely:

$$mp\ \frac{A \quad A \to B}{B} \qquad\qquad gen\ \frac{A}{\forall x A}$$

Proofs and derivations

Let A be a formula and Γ be a set of sentences (called **hypotheses**).

Proofs and derivations

Let A be a formula and Γ be a set of sentences (called **hypotheses**).

Definition (Derivation)

A **derivation** of A from Γ is a list of formulae $(A_1, \ldots, A_n)$ with $A_n = A$ (called the **conclusion**) such that each member of the sequence is either:

- an axiom;
- a hypothesis i.e. an element of Γ;
- obtained from previous formulae by an inference rule.

We write $\Gamma \vdash A$ if there exists a derivation of A from Γ.

Proofs and derivations

Let A be a formula and Γ be a set of sentences (called **hypotheses**).

Definition (Derivation)

A **derivation** of A from Γ is a list of formulae $(A_1, \ldots, A_n)$ with $A_n = A$ (called the **conclusion**) such that each member of the sequence is either:

- an axiom;
- a hypothesis i.e. an element of Γ;
- obtained from previous formulae by an inference rule.

We write $\Gamma \vdash A$ if there exists a derivation of A from Γ.

Definition (Proof)

A **proof** is a derivation from an empty set of premises. We write $\vdash A$ if there is a proof of A ("A is a **theorem**").

Example: $\vdash A \to A$

It is obvious that $A \to A$ is valid for any formula A.

Example: $\vdash A \to A$

It is obvious that $A \to A$ is valid for any formula A.

The following is an $\mathcal{F}$ proof of $A \to A$:

1.	$A \to (B \to A) \to A$	*wk*
2.	$(A \to (B \to A) \to A) \to (A \to (B \to A)) \to A \to A$	*dist*
3.	$(A \to B \to A) \to A \to A$	$mp(1, 2)$
4.	$A \to B \to A$	*wk*
5.	$A \to A$	$mp(4, 3)$

Example: $\{A \to B, B \to C\} \vdash A \to C$

The following is a derivation of $A \to C$ from the hypotheses $\{A \to B, B \to C\}$:

1.	$B \to C$	*hyp*
2.	$(B \to C) \to A \to (B \to C)$	*wk*
3.	$A \to (B \to C)$	$mp(1, 2)$
4.	$(A \to (B \to C)) \to (A \to B) \to (A \to C)$	*dist*
5.	$(A \to B) \to (A \to C)$	$mp(3, 4)$
6.	$A \to B$	*hyp*
7.	$A \to C$	$mp(6, 5)$

Example: $A[t/x] \to \neg\forall x.\neg A$

A *sketch* of a formal derivation (with propositional *tautologies* instead of axioms):

$\forall x.\neg A \to \neg A[t/x]$	quantifier axiom
$(\forall x.\neg A \to \neg A[t/x]) \to (\neg\neg A[t/x] \to \neg\forall x.\neg A)$	$(p \to q) \to (\neg q \to \neg p)$
$\neg\neg A[t/x] \to \neg\forall x.\neg A$	modus ponens
$A[t/x] \to \neg\neg A[t/x]$	$p \to \neg\neg p$
$A[t/x] \to \neg\forall x.\neg A$	$(p \to q) \to (q \to r) \to (p \to r)$

Example: $A[t/x] \to \neg\forall x.\neg A$

A *sketch* of a formal derivation (with propositional *tautologies* instead of axioms):

$\forall x.\neg A \to \neg A[t/x]$	quantifier axiom
$(\forall x.\neg A \to \neg A[t/x]) \to (\neg\neg A[t/x] \to \neg\forall x.\neg A)$	$(p \to q) \to (\neg q \to \neg p)$
$\neg\neg A[t/x] \to \neg\forall x.\neg A$	modus ponens
$A[t/x] \to \neg\neg A[t/x]$	$p \to \neg\neg p$
$A[t/x] \to \neg\forall x.\neg A$	$(p \to q) \to (q \to r) \to (p \to r)$

Defining $\exists x.A :\equiv \neg\forall x.\neg A$ we have proven

$$A[t/x] \to \exists x.A$$

If $\exists$ is taken as primitive, we would add this as an axiom.

Outline

The relationship between $\vdash$ and $\to$

You may have noticed that $\vdash$ and $\to$ seem to have a similar meaning, although they correspond to two quite different things, namely:

$$A \vdash B : \textit{I can derive } B \textit{ if I am allowed to take } A \textit{ as a hypothesis}$$

$$\vdash A \to B : \textit{I can prove the implication } A \to B \textit{ without any hypotheses}$$

The relationship between $\vdash$ and $\to$

You may have noticed that $\vdash$ and $\to$ seem to have a similar meaning, although they correspond to two quite different things, namely:

$$A \vdash B : \textit{I can derive } B \textit{ if I am allowed to take } A \textit{ as a hypothesis}$$

$$\vdash A \to B : \textit{I can prove the implication } A \to B \textit{ without any hypotheses}$$

The **deduction theorem** confirms that these two notions are equivalent.

The relationship between $\vdash$ and $\to$

You may have noticed that $\vdash$ and $\to$ seem to have a similar meaning, although they correspond to two quite different things, namely:

$$A \vdash B : \textit{I can derive } B \textit{ if I am allowed to take } A \textit{ as a hypothesis}$$
$$\vdash A \to B : \textit{I can prove the implication } A \to B \textit{ without any hypotheses}$$

The **deduction theorem** confirms that these two notions are equivalent.

Theorem (Deduction theorem)

$\Gamma \vdash A \to B$ *if and only if* $\Gamma \cup \{A\} \vdash B$.

The relationship between $\vdash$ and $\to$

You may have noticed that $\vdash$ and $\to$ seem to have a similar meaning, although they correspond to two quite different things, namely:

$$A \vdash B : \textit{I can derive } B \textit{ if I am allowed to take } A \textit{ as a hypothesis}$$
$$\vdash A \to B : \textit{I can prove the implication } A \to B \textit{ without any hypotheses}$$

The **deduction theorem** confirms that these two notions are equivalent.

Theorem (Deduction theorem)

$\Gamma \vdash A \to B$ *if and only if* $\Gamma \cup \{A\} \vdash B$.

Warning: The deduction theorem (and in fact many of the theorems we will present) do *not* necessarily hold in an arbitrary deductive systems.

The relationship between $\vdash$ and $\to$

You may have noticed that $\vdash$ and $\to$ seem to have a similar meaning, although they correspond to two quite different things, namely:

$$A \vdash B : \textit{I can derive } B \textit{ if I am allowed to take } A \textit{ as a hypothesis}$$
$$\vdash A \to B : \textit{I can prove the implication } A \to B \textit{ without any hypotheses}$$

The **deduction theorem** confirms that these two notions are equivalent.

Theorem (Deduction theorem)

$\Gamma \vdash A \to B$ *if and only if* $\Gamma \cup \{A\} \vdash B$.

Warning: The deduction theorem (and in fact many of the theorems we will present) do *not* necessarily hold in an arbitrary deductive systems.

Proof (easy direction $\Rightarrow$).

Suppose that $(A_1, \ldots, A_n, A \to B)$ is a derivation of $A \to B$ from Γ.

The relationship between $\vdash$ and $\to$

You may have noticed that $\vdash$ and $\to$ seem to have a similar meaning, although they correspond to two quite different things, namely:

$$A \vdash B : \textit{I can derive } B \textit{ if I am allowed to take } A \textit{ as a hypothesis}$$
$$\vdash A \to B : \textit{I can prove the implication } A \to B \textit{ without any hypotheses}$$

The **deduction theorem** confirms that these two notions are equivalent.

Theorem (Deduction theorem)

$\Gamma \vdash A \to B$ *if and only if* $\Gamma \cup \{A\} \vdash B$.

Warning: The deduction theorem (and in fact many of the theorems we will present) do *not* necessarily hold in an arbitrary deductive systems.

Proof (easy direction $\Rightarrow$).

Suppose that $(A_1, \ldots, A_n, A \to B)$ is a derivation of $A \to B$ from Γ.

Then $(A_1, \ldots, A_n, A \to B, A, B)$ is a derivation of B from $\Gamma \cup \{A\}$. $\square$

Proof (hard direction $\Leftarrow$)

Suppose that $(A_1, \ldots, A_n = B)$ is a derivation of B from $\Gamma \cup \{A\}$. We show by structural induction on the derivation that $\Gamma \vdash A \to A_i$ for all $i = 1, \ldots, n$.

Proof (hard direction $\Leftarrow$)

Suppose that $(A_1, \ldots, A_n = B)$ is a derivation of B from $\Gamma \cup \{A\}$. We show by structural induction on the derivation that $\Gamma \vdash A \to A_i$ for all $i = 1, \ldots, n$.

If A_i is an axiom or element of Γ, then a derivation $\Gamma \vdash A \to A_i$ is given via

$$\frac{A_i \quad A_i \to (A \to A_i)}{A \to A_i}$$

Proof (hard direction $\Leftarrow$)

Suppose that $(A_1, \ldots, A_n = B)$ is a derivation of B from $\Gamma \cup \{A\}$. We show by structural induction on the derivation that $\Gamma \vdash A \to A_i$ for all $i = 1, \ldots, n$.

If A_i is an axiom or element of Γ, then a derivation $\Gamma \vdash A \to A_i$ is given via

$$\frac{A_i \quad A_i \to (A \to A_i)}{A \to A_i}$$

If A_i is A, then we have $\Gamma \vdash A \to A$ (we showed this earlier).

Proof (hard direction $\Leftarrow$)

Suppose that $(A_1, \ldots, A_n = B)$ is a derivation of B from $\Gamma \cup \{A\}$. We show by structural induction on the derivation that $\Gamma \vdash A \to A_i$ for all $i = 1, \ldots, n$.

If A_i is an axiom or element of Γ, then a derivation $\Gamma \vdash A \to A_i$ is given via

$$\frac{A_i \quad A_i \to (A \to A_i)}{A \to A_i}$$

If A_i is A, then we have $\Gamma \vdash A \to A$ (we showed this earlier).

Otherwise, A_i follows from A_j and $A_j \to A_i$ where these occur previously in the derivation. By the induction hypothesis we have $\Gamma \vdash A \to A_j$ and $\Gamma \vdash A \to (A_j \to A_i)$,

Proof (hard direction $\Leftarrow$)

Suppose that $(A_1, \ldots, A_n = B)$ is a derivation of B from $\Gamma \cup \{A\}$. We show by structural induction on the derivation that $\Gamma \vdash A \to A_i$ for all $i = 1, \ldots, n$.

If A_i is an axiom or element of Γ, then a derivation $\Gamma \vdash A \to A_i$ is given via

$$\frac{A_i \quad A_i \to (A \to A_i)}{A \to A_i}$$

If A_i is A, then we have $\Gamma \vdash A \to A$ (we showed this earlier).

Otherwise, A_i follows from A_j and $A_j \to A_i$ where these occur previously in the derivation. By the induction hypothesis we have $\Gamma \vdash A \to A_j$ and $\Gamma \vdash A \to (A_j \to A_i)$, and therefore

$$\dfrac{\dfrac{\vdots}{A \to A_j} \qquad \dfrac{\dfrac{\vdots}{A \to (A_j \to A_i)} \quad (A \to (A_j \to A_i)) \to (A \to A_j) \to (A \to A_i)}{(A \to A_j) \to A \to A_i}}{A \to A_i}$$

represents a derivation $\Gamma \vdash A \to A_i$.

Proof (hard direction $\Leftarrow$)

Suppose that $(A_1, \ldots, A_n = B)$ is a derivation of B from $\Gamma \cup \{A\}$. We show by structural induction on the derivation that $\Gamma \vdash A \to A_i$ for all $i = 1, \ldots, n$.

If A_i is an axiom or element of Γ, then a derivation $\Gamma \vdash A \to A_i$ is given via

$$\frac{A_i \quad A_i \to (A \to A_i)}{A \to A_i}$$

If A_i is A, then we have $\Gamma \vdash A \to A$ (we showed this earlier).

Otherwise, A_i follows from A_j and $A_j \to A_i$ where these occur previously in the derivation. By the induction hypothesis we have $\Gamma \vdash A \to A_j$ and $\Gamma \vdash A \to (A_j \to A_i)$, and therefore

$$\dfrac{\dfrac{\vdots}{A \to A_j} \qquad \dfrac{\dfrac{\vdots}{A \to (A_j \to A_i)} \quad (A \to (A_j \to A_i)) \to (A \to A_j) \to (A \to A_i)}{(A \to A_j) \to A \to A_i}}{A \to A_i}$$

represents a derivation $\Gamma \vdash A \to A_i$.

Thus by induction we have $\Gamma \vdash A \to A_i$ for all i, and in particular $\Gamma \vdash A \to B$. $\square$

Outline

Definition (First-order theories and their models)

A **first-order theory** Γ over some language is a set of sentences in that language.

Definition (First-order theories and their models)

A **first-order theory** Γ over some language is a set of sentences in that language.

A **model** of a theory Γ is a structure $\mathcal{D}$ with $\mathcal{D} \models A$ for all $A \in \Gamma$. We write $\mathcal{D} \models \Gamma$ in this case.

Definition (First-order theories and their models)

A **first-order theory** Γ over some language is a set of sentences in that language.

A **model** of a theory Γ is a structure $\mathcal{D}$ with $\mathcal{D} \models A$ for all $A \in \Gamma$. We write $\mathcal{D} \models \Gamma$ in this case.

Example

Consider the language with a single constant e, a unary function symbol i and a binary function symbol m. Let G be the theory given by the following *axioms*:

$$\forall x(m(e,x) = x = m(x,e))$$

$$\forall x(m(x,i(x)) = e = m(i(x),x))$$

$$\forall x,y,z(m(x,m(y,z)) = m(m(x,y),z))$$

Definition (First-order theories and their models)

A **first-order theory** Γ over some language is a set of sentences in that language.

A **model** of a theory Γ is a structure $\mathcal{D}$ with $\mathcal{D} \vDash A$ for all $A \in \Gamma$. We write $\mathcal{D} \vDash \Gamma$ in this case.

Example

Consider the language with a single constant e, a unary function symbol i and a binary function symbol m. Let G be the theory given by the following *axioms*:

$$\forall x(m(e,x) = x = m(x,e))$$

$$\forall x(m(x,i(x)) = e = m(i(x),x))$$

$$\forall x,y,z(m(x,m(y,z)) = m(m(x,y),z))$$

Then G is a model of G precisely when G is a group!

Definition (First-order theories and their models)

A **first-order theory** Γ over some language is a set of sentences in that language.

A **model** of a theory Γ is a structure $\mathcal{D}$ with $\mathcal{D} \vDash A$ for all $A \in \Gamma$. We write $\mathcal{D} \vDash \Gamma$ in this case.

Example

Consider the language with a single constant e, a unary function symbol i and a binary function symbol m. Let G be the theory given by the following *axioms*:

$$\forall x(m(e,x) = x = m(x,e))$$

$$\forall x(m(x,i(x)) = e = m(i(x),x))$$

$$\forall x,y,z(m(x,m(y,z)) = m(m(x,y),z))$$

Then G is a model of G precisely when G is a group!

What mathematical entities can be captured as a first order theory?

There are lots of examples:

- equivalence relations
- orders
- various types of groups
- rings, fields
- . . .

What mathematical entities can be captured as a first order theory?

There are lots of examples:

- equivalence relations
- orders
- various types of groups
- rings, fields
- . . .

Roughly speaking, the more concrete the entity we are trying to capture, the more axioms we need.

What mathematical entities can be captured as a first order theory?

There are lots of examples:

- equivalence relations
- orders
- various types of groups
- rings, fields
- . . .

Roughly speaking, the more concrete the entity we are trying to capture, the more axioms we need.

Groups are very abstract and extremely easy to axiomatize. But what about numbers?

Axiomatising the natural numbers

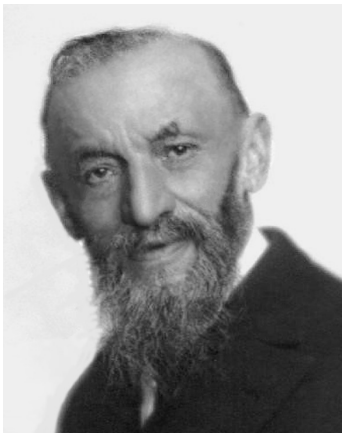
Peano Arithmetic PA is a first-order theory which captures the natural numbers together with simple arithmetical operations on them.

It is named after Giuseppe Peano (1858-1932), who gave one of the first axiomatisations of arithmetic.

Axiomatising the natural numbers

Peano Arithmetic PA is a first-order theory which captures the natural numbers together with simple arithmetical operations on them.

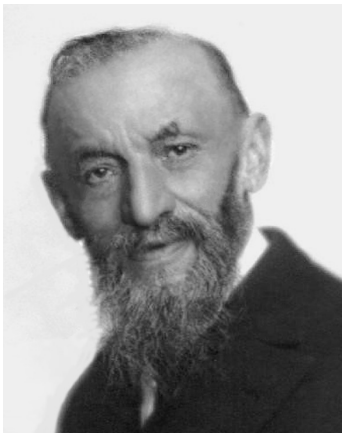
It is named after Giuseppe Peano (1858-1932), who gave one of the first axiomatisations of arithmetic.



Axiomatising the natural numbers

Peano Arithmetic PA is a first-order theory which captures the natural numbers together with simple arithmetical operations on them.

It is named after Giuseppe Peano (1858-1932), who gave one of the first axiomatisations of arithmetic.



The crucial axioms in PA are the axioms of **induction**.

Remark. Induction is not just a tool to prove things about numbers, but an axiom schema which characterises them.

The theory of Peano arithmetic

Definition (Peano arithmetic)

Peano Arithmetic is a first-order theory over the language with a single constant 0, a unary symbol s ('successor') and two binary constants $+$ and $\cdot$, which we usually write infix (so $x + y$ instead of $+(x, y)$).

Definition (Peano arithmetic)

Peano Arithmetic is a first-order theory over the language with a single constant 0, a unary symbol s ('successor') and two binary constants $+$ and $\cdot$, which we usually write infix (so $x + y$ instead of $+(x, y)$).

PA includes the following axioms:

- $\forall x \neg(0 = s(x))$
- $\forall x, y(s(x) = s(y) \rightarrow x = y)$
- $\forall x(x + 0 = x)$
- $\forall x, y(x + s(y) = s(x + y))$
- $\forall x(x \cdot 0 = 0)$
- $\forall x, y(x \cdot s(y) = x \cdot y + x)$

Definition (Peano arithmetic)

Peano Arithmetic is a first-order theory over the language with a single constant 0, a unary symbol s ('successor') and two binary constants $+$ and $\cdot$, which we usually write infix (so $x + y$ instead of $+(x, y)$).

PA includes the following axioms:

- $\forall x \neg(0 = s(x))$
- $\forall x, y(s(x) = s(y) \to x = y)$
- $\forall x(x + 0 = x)$
- $\forall x, y(x + s(y) = s(x + y))$
- $\forall x(x \cdot 0 = 0)$
- $\forall x, y(x \cdot s(y) = x \cdot y + x)$

In addition, we need an *axiom schema* of **induction**. Namely for each formula A of our language with $\mathsf{FV}(A) = \{x, y_1, \ldots, y_k\}$ we include the axiom:

$$\forall y_1, \ldots, y_k \, (A[0/x] \wedge \forall x(A \to A[s(x)/x]) \to \forall x.A)$$

Example: $\forall x(0 + x = x)$

Example: $\forall x(0 + x = x)$

First note that we have $0 + 0 = 0$. Now, using equality and axioms together with those for $+$, we get the following derivation (sketched)

Example: $\forall x(0 + x = x)$

First note that we have $0 + 0 = 0$. Now, using equality and axioms together with those for $+$, we get the following derivation (sketched)

1. $0 + x = x \to s(0 + x) = s(x)$	equality axioms
2. $0 + s(x) = s(0 + x) \to s(0 + x) = s(x) \to 0 + s(x) = s(x)$	equality axioms
3. $0 + s(x) = s(0 + x)$	axioms for $+$
4. $s(0 + x) = s(x) \to 0 + s(x) = s(x)$	modus ponens (3,2)
5. $0 + x = x \to 0 + s(x) = s(x)$	propositional taut. (1,4)
6. $\forall x(0 + x = x \to 0 + s(x) = s(x))$	$\forall$ rule

Example: $\forall x(0 + x = x)$

First note that we have $0 + 0 = 0$. Now, using equality and axioms together with those for $+$, we get the following derivation (sketched)

1. $0 + x = x \to s(0 + x) = s(x)$	equality axioms
2. $0 + s(x) = s(0 + x) \to s(0 + x) = s(x) \to 0 + s(x) = s(x)$	equality axioms
3. $0 + s(x) = s(0 + x)$	axioms for $+$
4. $s(0 + x) = s(x) \to 0 + s(x) = s(x)$	modus ponens (3,2)
5. $0 + x = x \to 0 + s(x) = s(x)$	propositional taut. (1,4)
6. $\forall x(0 + x = x \to 0 + s(x) = s(x))$	$\forall$ rule

Now for $A :\equiv 0 + x = x$ the corresponding instance of induction is

$$\underbrace{0 + 0 = 0}_{A[0/x]} \wedge \forall x(\underbrace{0 + x = x}_{A} \to \underbrace{0 + s(x) = s(x)}_{A[s(x)/x]}) \to \forall x(\underbrace{0 + x = x}_{A})$$

and so combining this with the above we have

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

Models of PA

Models of PA

The obvious model of PA we have in mind is the structure $\mathcal{N}$ of natural numbers, with domain $\mathbb{N}$ and $0_{\mathcal{N}}$, $s_{\mathcal{N}}$, $+_{\mathcal{N}}$ and $\cdot_{\mathcal{N}}$ given by by the obvious interpretations.

Models of PA

The obvious model of PA we have in mind is the structure $\mathcal{N}$ of natural numbers, with domain $\mathbb{N}$ and $0_{\mathcal{N}}$, $s_{\mathcal{N}}$, $+_{\mathcal{N}}$ and $\cdot_{\mathcal{N}}$ given by by the obvious interpretations.

We clearly have

$$\mathcal{N} \models \mathrm{PA}$$

Models of PA

The obvious model of PA we have in mind is the structure $\mathcal{N}$ of natural numbers, with domain $\mathbb{N}$ and $0_{\mathcal{N}}$, $s_{\mathcal{N}}$, $+_{\mathcal{N}}$ and $\cdot_{\mathcal{N}}$ given by by the obvious interpretations.

We clearly have

$$\mathcal{N} \models \text{PA}$$

Ideally we would hope for a converse too, namely that whenever

$$\mathcal{D} \models \text{PA}$$

then the structure $\mathcal{D}$ is in some sense isomorphic to $\mathcal{N}$.

Models of PA

The obvious model of PA we have in mind is the structure $\mathcal{N}$ of natural numbers, with domain $\mathbb{N}$ and $0_{\mathcal{N}}$, $s_{\mathcal{N}}$, $+_{\mathcal{N}}$ and $\cdot_{\mathcal{N}}$ given by by the obvious interpretations.

We clearly have

$$\mathcal{N} \models \mathrm{PA}$$

Ideally we would hope for a converse too, namely that whenever

$$\mathcal{D} \models \mathrm{PA}$$

then the structure $\mathcal{D}$ is in some sense isomorphic to $\mathcal{N}$.

But it turns out that this isn't possible! I will say more about this in the next lecture…

Outline

1. Show $\{A, B, A \to (B \to C)\} \vdash C$ and hence $\vdash A \to (B \to C) \to B \to A \to C$.
2. We can extend $\mathcal{F}$ to include conjunction $\wedge$ by adding the following axioms:

$$\begin{array}{rl} (pair) & p \to q \to (p \wedge q) \\ (pl) & (p \wedge q) \to p \\ (pr) & (p \wedge q) \to q \end{array}$$

   Show that this extended system proves: $(A \to (B \to C)) \leftrightarrow ((A \wedge B) \to C)$
3. We haven't yet used the axiom (neg)! Show that $\mathcal{F}$ proves:

$$\begin{array}{l} \neg A \to (A \to B) \\ (\neg B \to \neg A) \to (A \to B) \\ ((A \to B) \to A) \to A \end{array}$$

4. (**Hard**). Write $\mathcal{F}(\to)$ for the fragment of $\mathcal{F}$ without the neg axiom.
   - Notice that $\mathcal{F}(\to)$ still satisfies the Deduction theorem!
   - Show $\mathcal{F}(\to)$ is not equivalent to $\mathcal{F}$ over $\to$-only formulae. (Non-conservativity)
   - On the other hand, show that $\mathcal{F}(\to) + ((p \to q) \to p) \to p$ is equivalent to $\mathcal{F}$ over $\to$-only formulae.

1. Suppose we have a language with at least one constant c and a unary predicate symbol P. Investigate the so-called drinkers paradox:

$$\mathrm{DP} := \exists x(P(x) \to \forall y.P(y))$$

In natural language, this is popularly captured by the statement:

In any pub there is a person such that if they are drinking, then everyone is drinking.

Is DP valid? If so, can you sketch a derivation?

2. Show that $\{A \to B\} \vdash \neg\forall x.\neg A \to B$ when $x \notin \mathsf{FV}(B)$. This corresponds to the $\exists$-rule

$$\frac{A \to B}{\exists x.A \to B}$$

3. Outline a first-order theory whose models are the partial orders. Adapt this theory to characterise
   - total orders e.g. $\mathbb{Z}$ with $\leq$
   - total orders with a minimum element e.g. $\mathcal{N}$ with $\leq, 0$
   - partial orders with least upper bounds e.g. $\mathcal{P}(\mathcal{N})$ with $\subseteq$ and $\cup$

Outline

References I

Buss, S. R., editor (1998).
Handbook of Proof Theory, volume 137 of *Studies in Logic and the Foundations of Mathematics*.
Elsevier.

Frege, G. (1879).
Begriffsschrift: Eine Der Arithmetische Nachgebildete Formelsprache des Reinen Denkens.
L. Nebert.

Gödel, K. (1931).
über formal unentscheidbare sätze der principia mathematica und verwandter systeme i.
Monatshefte für Mathematik, 38(1):173–198.

Schoenfeld, J. S. (1967).
Mathematical Logic.
Addison-Wesley, Reading.

Smullyan, R. M. (1968).
First-Order Logic.
Springer-Verlag.

References II

Smullyan, R. M. (1980).
What is the name of this book? the riddle of dracula and other logical puzzles.
Philosophical Review, 89(3):467–470.

Szabo, M. E. (1972).
The Collected Papers of Gerhard Gentzen.
Philosophy of Science, 39(1):91–91.

Troelstra, A. S. and Schwichtenberg, H. (1996).
Basic Proof Theory.
Cambridge University Press, New York, NY, USA.

Whitehead, A. N. and Russell, B. (1925–1927).
Principia Mathematica.
Cambridge University Press.