Mathematical Logic

Lecture 1

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Bureaucracy:

- Introduction
- Program
- Examination
- Timing
- Questions

A brief history logic.

Introduction

Mathematical logic is the field of Mathematics which studies the deduction process and the foundations of the whole discipline.

This course will introduce mathematical logic from the very beginning, requiring as a prerequisite a minimal knowledge of elementary mathematics at the university level.

The material of the course is more or less standard, and most introductory textbooks will cover it. For the purposes of this course the slides are made available to students.

The course is in English.

Program

The course takes 78 hours, and its content will be an introduction to classical logic with a glimpse to other logical systems.

The detailed program is

- *Propositional logic*: language, deduction system, semantics, soundness, completeness;
- *First-order logic*: syntax, semantics, soundness, completeness, compactness, extensions of models;
- *Set theory*: fundamental axioms, ordinals, cardinals, transfinite induction, axiom of choice, continuum hypothesis;
- *Computability*: computable functions, λ-calculi, simple theory of types;
- *Constructive mathematics*: intuitionistic logic, expressive power, semantics, propositions as types;
- *Limiting results*: Peano arithmetic, Gödel's incompleteness theorems, natural incompleteness results.

Texts

All the slides are available at the course website:

`http://marcobenini.me/lectures/mathematical-logic`

Also, at the end of each lesson, references to articles, texts, and other resources which may be of interest to those interested in learning more, will be provided. While the content of the slides is *mandatory*, looking at the references is *optional*. Some non-official online video lessons are available on the website.

Although there is no standard textbook, I will mainly use the classical Bell, Machover *A Course in Mathematical Logic*.

Examination

The examination will be oral. It will require to perform simple exercises, like proving a theorem using a formal deductive system, and to state, discuss, and prove the results explained during the course.

The examination will be, at the student's choice, either in Italian or in English.

Informally, a student may take the examination by fixing an appointment: this can be done at every time after the end of the course.
Formally, examinations can be registered only during the scheduled dates: students **must** subscribe the date to be able to register their marks. Students are strongly encouraged to plan when to take examinations, and to fix an appointment in advance. Then, they can register the result whenever they prefer, within 18 months from the end of the course.

As usual, independently from the results, repeating an examination cancels the (unregistered) previous marks.

Examination

Although it is not mandatory, there will be four intermediate assignments during the course. Each assignment is preceded by a lesson which examines selected exercises from past assignments.

Assignments will take place during lesson time, and they will cover

1. propositional logic
2. first-order logic
3. set theory and computability
4. constructive mathematics and limiting results

Students willing to take them can avoid the examination: each assignment will get a mark, and the average will be the final mark. Rules for registration are the same as for regular examinations.

The assignments of previous years are available at the course website.

Timing

The schedule of lessons is fixed, and it cannot be easily changed. In general, a lesson will start 10 minutes after the official time, and it will finish 10 minutes before the official time, with no breaks.

The intermediate assignments will take place during lesson time. You students decide when they will take place, choosing a date after the end of the corresponding section of the course.

Questions

Questions are welcome. Please, do not hesitate to ask questions when you do not understand something during a lesson.

Questions could be asked also before the start of a lesson, or after the end.

Another possibility is to ask questions by email: in case write at the address

`marco.benini@uninsubria.it`

specifying your name, the course, and the question. Please, use your *official* email from `uninsubria`.

There are no office hours in this course: students have to fix an appointment. Please, do so only if you really think there is no other way to solve your problem: although I am usually available to receive students during the course time, when I am not teaching, it is often the case that I am not around in University, so use this opportunity as your last resource.
Online appointments are always possible and encouraged.

Your teacher

I am a researcher in Mathematical Logic. This means that my main job is to think, and hopefully to find novel results in this field of Mathematics.

Teaching is part of my academic duties, but is not my first occupation.

As a logician, my interests lie in the interplay between truth and computability. Indeed, I investigate mainly constructive logical systems, which have nice computational properties, and my current playground, the 'universe' I work within, is Homotopy Type Theory.

For more, please visit my web page:

`http://marcobenini.me`

Mathematical logic

Mathematical logic studies the mathematical deduction process and the notion of truth, at large.

Logic is an ancient part of Mathematics: its origins go back to Aristotle, while its mathematical foundations can be traced in the work of Boole, Frege, Cantor, Russell, Hilbert, Gödel, ...

Since Gödel's Incompleteness results, the discipline underwent a huge development, and today it is a very active part of contemporary Mathematics, with application in Computer Science and Philosophy.

Since this is a first course in mathematical logic, we will stop after proving the incompleteness results. Here and there, hints about future developments will be given, but the course sticks on the classical track.

Greek mathematics

©Marco Benini, Pytaghoras in Samos

Proof
Theorem

Greek mathematics

Logos

Johannes Moreelse,
Heraclite,
Centraal Museum, Utrecht,
1630

Proof by contradiction
Paradoxes on infinity

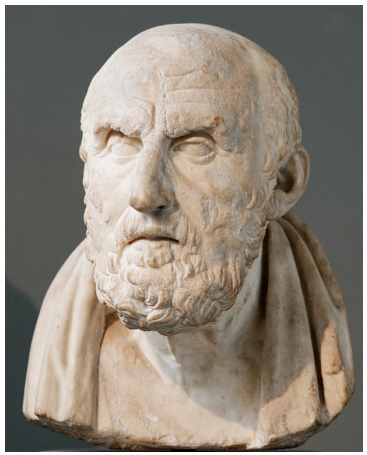
Marcus Meibomius,
engraving of Zeno of Elea in Diogenis Laertii De Vitis,
1698

Greek mathematics

Formal reasoning
Excluded middle
Formal system

Organon,
Aristotle

Greek mathematics



Chrysippos of Soli,
Marble, Roman copy of the late 3rd century BC

Conditionals, implication
Relation between meaning and truth, semantics

Islam

Avicenna
Portrait on Silver Vase
Museum at BuAli Sina (Avicenna) Mausoleum
Hamadan, Western Iran
©Adam Jones photographer, 2012

Precursors of ideal objects
Algebra
Algorithm

Medieval Europe

Museum of Galileo, Florence,
©Marco Benini, 2015

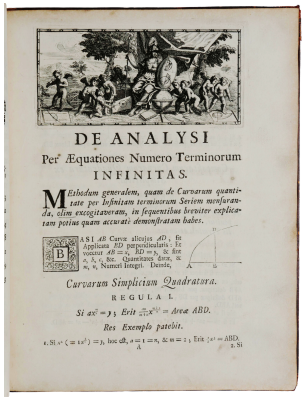
Descartes

Portrait of René Descartes by Frans Hals

Analytic geometry
Bridge between geometry and algebra
Space

Newton

Mathematical analysis
Foundations in Euclidean geometry



DE ANALYSI
Per Æquationes Numero Terminorum
INFINITAS.

Curvarum Simplicium Quadratura.

REGULA I.

Res Exemplo patebit.

De analysi per aequationes numero terminorum infinitas,
Isaac Newton, 1711

Liebniz

Bildnis des Philosophen Gottfried Wilhelm Freiherr von Leibniz,
Christoph Bernhard Francke, 1695

Mathematical analysis
Characteristica Universalis

The crisis in analysis

Carl Friedrich Gauß,
Christian Albrecht Jensen, 1840

Augustin Louis Cauchy,
photo by Charles Reutlinger

The crisis in analysis

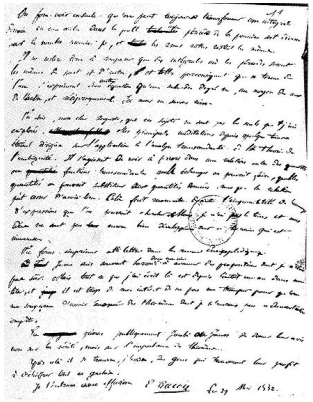
Georg Friedrich Bernhard Riemann, 1863

Julius Wilhelm Richard Dedekind,
photo by Johannes Ganz, 1866

Karl Weierstraß,
Conrad Fehr, 1895

Revolution

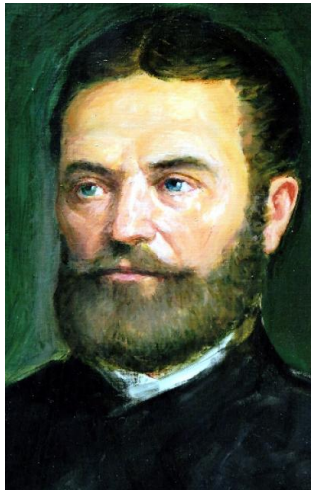
Algebra



–
Last page of the letter from Évariste Galois to Auguste Chevalier,
29th March 1832

Niels Henrik Abel

Non-Euclidean geometry

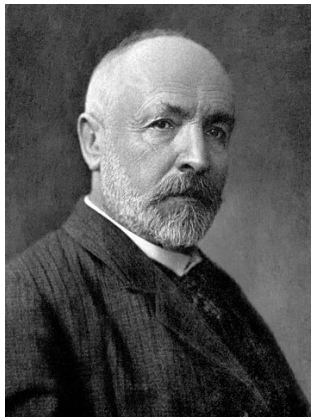


János Bolyai,
painting by Márkos Ferenc, 2012

Nikolai Ivanovich Lobachevsky,
portrait by Lev Kryukov, 1843

Foundations of Mathematics

Cantor

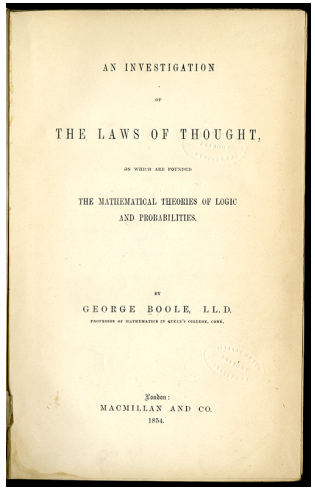


Georg Ferdinand Ludwig Philipp Cantor,
1910

Set theory
Infinities
Cardinality

Boole

The Laws of Thought
Mathematical logic


AN INVESTIGATION
OF
THE LAWS OF THOUGHT,
ON WHICH ARE FOUNDED
THE MATHEMATICAL THEORIES OF LOGIC
AND PROBABILITIES.
BY
GEORGE BOOLE, LL.D.
London:
MACMILLAN AND CO.
1854.

The Laws of Thought,
1854

Frege

Friedrich Ludwig Gottlob Frege,
1879

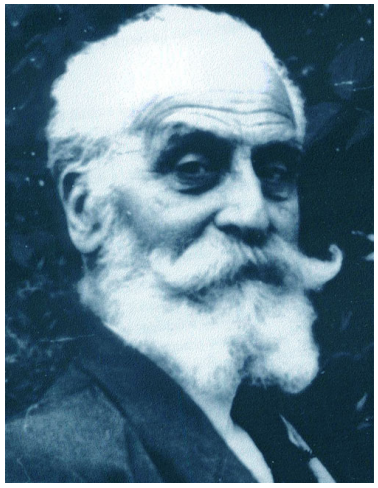
Variables
Quantifiers

Russell's paradox

Let $R = \{x : x \notin x\}$.

Then $R \in R$ if and only if $R \notin R$.

Peano



Giuseppe Peano

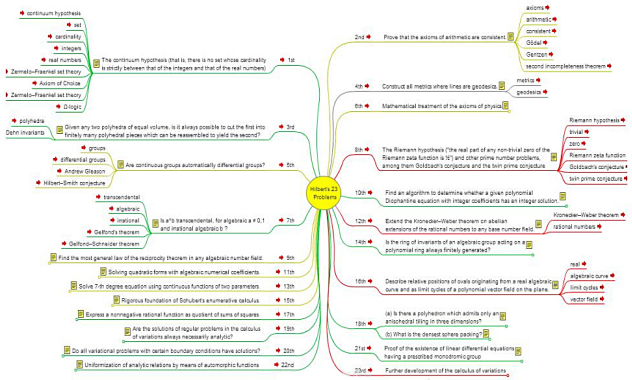
Formal arithmetic
Induction

Hilbert

David Hilbert,
1912

Formal geometry

Hilbert



© Sharjeel Khan

Hilbert

Formalisation: all mathematical statements have to be written, at least in principle, in a precise formal language and manipulated according to a fixed, precise and formal set of rules.

Consistency: the whole corpus of mathematics has to be proved to be contradiction free by means of a formal proof inside mathematics itself.

Finitistic: the language, the rules of inference, and the proofs have to be finite and effective. In particular, the consistency proofs have to be finitistic.

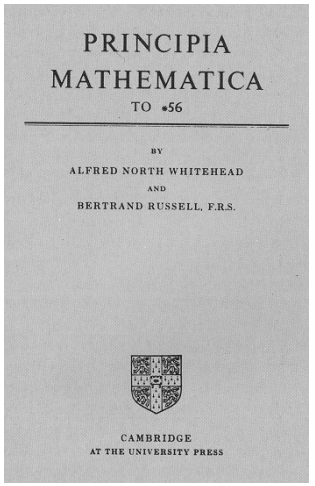
Zermelo and Frænkel

Ernst Zermelo,
1900

Abraham Halevi Frænkel,
1939–49

Russell

Bertrand Russell



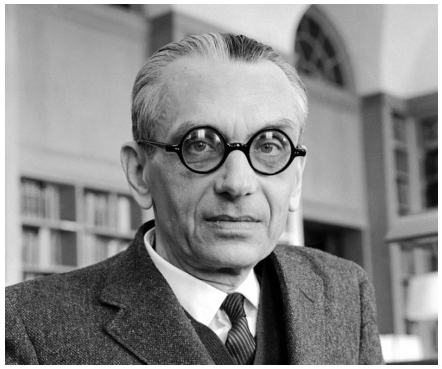
Principia Mathematica

Löwenheim and Skolem

Leopold Löwenheim

Thoralf Skolem,
1930

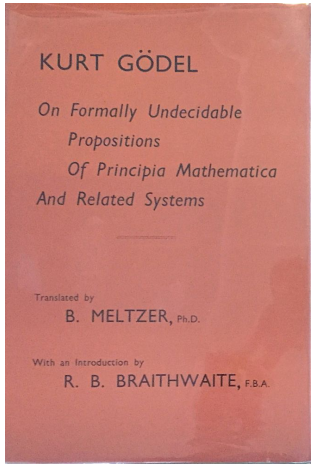
Gödel



Kurt Gödel

Completeness of first order logic

Gödel



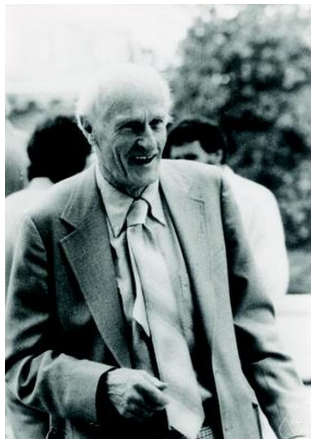
Incompleteness theorems

Gentzen

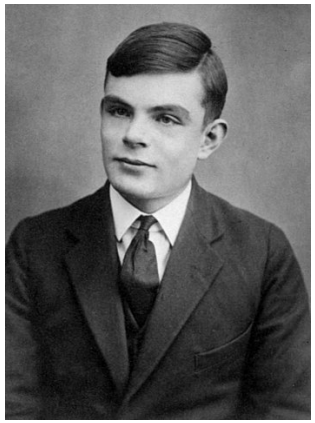
Gerhard Gentzen,
photo by Eckart Menzler-Trott, Prague, 1945

Consistency of arithmetic
Cut elimination
Proof theory

Computability theory



Stephen Cole Kleene,
photo by Konrad Jacobs, Erlangen, 1978



Alan Mathison Turing

Computability theory

Halting problem
Church-Turing thesis

Alonzo Church

Intuitionism

Luitzen Egbertus Jan Brouwer

Constructive mathematics

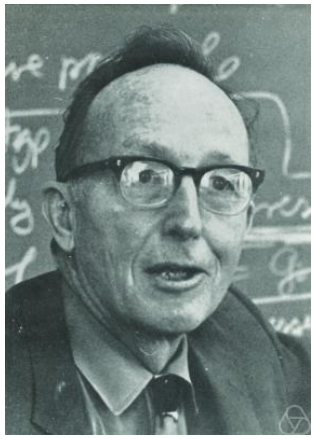
Afterwards

Mathematical logic (Jon Barwise, 1977):

- Set theory
- Proof theory
- Model theory
- Recursion theory

Nowadays also: Category theory, Topos theory, Type Theory, ...

Afterwards

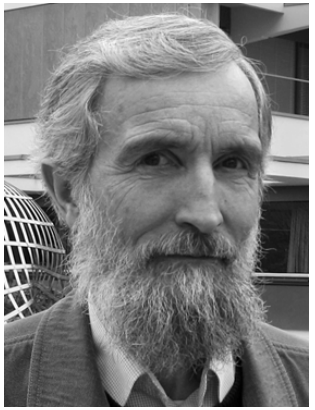


Saunders Mac Lane,
photo by Konrad Jacobs, 1972

Category theory

Afterwards

Type theory



Per Martin-Löf

Afterwards

The Art of Ordinal Analysis

Michael Rathjen

Abstract. Ordinal analysis of theories is a core area of proof theory whose origins can be traced back to Hilbert's programme - the aim of which was to lay to rest all worries about the foundations of mathematics once and for all by securing mathematics via an absolute proof of consistency. Ordinal-theoretic proof theory came into existence in 1936, springing forth from Gentzen's head in the course of his consistency proof of arithmetic. The central theme of ordinal analysis is the classification of theories by means of transfinite ordinals that measure their 'consistency strength' and 'computational power'. The so-called *proof-theoretic ordinal* of a theory also serves to characterize its provably recursive functions and can yield both conservation and combinatorial independence results.

This paper intends to survey the development of "ordinally informative" proof theory from the work of Gentzen up to more recent advances in determining the proof-theoretic ordinals of strong subsystems of second order arithmetic.

Mathematics Subject Classification (2000). Primary 03F15, 03F05, 03F35; Secondary 03F03, 03-03.

Keywords. Proof theory, ordinal analysis, ordinal representation systems, proof-theoretic strength.

References

Two classical books on the history of Mathematics are: *Carl B. Boyer*, A History of Mathematics, John Wiley & Sons (1968), and *Morris Kline*, Mathematical Thought from Ancient to Modern Times, Oxford University Press (1972). Also, very good references to authors and ideas can be found in the Stanford Encyclopedia of Philosophy: `https://plato.stanford.edu/`

For those interested in the recent history of logic, a nice and short book is *Piergiorgio Odifreddi*, La matematica del Novecento—Dagli insiemi alla complessità, Piccola Biblioteca Einaudi, Einaudi, (2000).

There are many introductory textbooks of mathematical logic and a few important reference texts. I would like to mention the comprehensive guide *Jon Barwise*, Handbook of Mathematical Logic, Studies in Logic and the Foundations of Mathematics 90, North-Holland, (1977).

Although not required, most of the course is based on *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North Holland (1977).

Mathematical Logic

Lecture 2

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Propositional logic:

- Induction
- Language
- Intended interpretation
- Deduction system

Proving by induction

Intuitively, to show that a property $P(x)$ holds for every possible value of x in a domain D, one could substitute x with each $v \in D$ and prove $P(v)$. This is, generally, impractical, and impossible when D is infinite.

However, if D can be generated by a process which has a finite description, or in jargon, if D is *finitely generated*, then one may use the generation process to prove $P(x)$.

This idea is called *induction*.

Proving by induction

Let $\mathbb{N}$ be the set of natural numbers $\{0,1,2,\ldots\}$.

Observe how $\mathbb{N}$ can be finitely generated:

1. $0 \in \mathbb{N}$;
2. if $m \in \mathbb{N}$ then $S(m) \in \mathbb{N}$, with S the *successor* function, $S(x) = x+1$.

Hence, if P is a property, we have an *induction principle*:

1. if we have a proof of $P(0)$,
2. if we are able to prove $P(S(m))$ from the hypothesis $P(m)$,

then $P(x)$ holds for every $x \in \mathbb{N}$.

Proving by induction

The induction principle is intuitively justified:

- we have a proof π_0 of $P(0)$ by step 1;
- composing π_0 with the proof of step 2, we obtain a proof π_1 of $P(1)$;
- composing π_1 with the proof of step 2, we obtain a proof π_2 of $P(2)$;
- and so on ...

So, whatever value $v \in \mathbb{N}$, we will find a proof for $P(v)$ in the list above.

Example

Proposition 2.1

$\sum_{n=0}^{k} n = \frac{k(k+1)}{2}$.

Proof.

By induction on $k \in \mathbb{N}$:

1. when $k = 0$, $\sum_{n=0}^{0} n = 0 = \frac{0(0+1)}{2}$;
2. assume $\sum_{n=0}^{k} n = \frac{k(k+1)}{2}$. Then,

$$\sum_{n=0}^{k+1} n = k+1+\sum_{n=0}^{k} n = k+1+\frac{k(k+1)}{2} = \frac{k^2+3k+2}{2} = \frac{(k+1)((k+1)+1)}{2} .$$

□

Definition by induction

Induction can be used to define new concepts and new objects.

Let τ be a map from $\mathbb{N}$. Then, the image of τ is a new concept, and its elements are new objects.

For example, posing $\tau(x) = 2x$, we define a new concept: *even numbers*.

Considering the collection of all the maps $\rho : \mathbb{N} \to \{0,\ldots,9\}$ to the set of digits, we obtain new objects, one for each ρ, denoting the real number $0.\rho(0)\rho(1)\cdots$ whose digit at the decimal position n is $\rho(n)$, and a new concept, the *unit interval* $[0,1]$.

In general

The idea of induction is far more general: whenever we can finitely generate a domain, we have an induction principle which can be used to reason and to define properties and concepts.

In Mathematical Logic, induction is one of the fundamental and most powerful tools.

Propositional logic

In this lesson, we want to introduce classical propositional logic.

We will start from its syntax, and its intended meaning.

The idea is that a proposition stands for a *truth value*, either *true* or *false*. Composite propositions will derive their truth value from their components, while basic propositions will have a truth value which depends on the world they are interpreted in.

For example, the sentence 'Socrates is a man' may be true or false, as Socrates may be the ancient Greek philosopher, or a cat. On the other side, 'If Socrates is a man then Socrates is mortal' is true when Socrates is both a man and mortal, but also when Socrates is not a man, and it is false when Socrates is an immortal man.

Language

Definition 2.2 (Formula)

Let V be an infinite (countable) set of symbols, called *variables*, not containing '(', ')', '$\top$', '$\bot$', '$\wedge$', '$\vee$', '$\supset$', '$\neg$'.
Then, a *formula* is inductively defined as

1. a variable $x \in V$ is a formula;
2. $\top$, spelt *true*, and $\bot$, *false*, are formulæ;
3. if A is a formula, so is $(\neg A)$, *not*, *negation*;
4. if A and B are formulæ, so are $(A \wedge B)$, *and*, *conjunction*; $(A \vee B)$, *or*, *disjunction*; and $(A \supset B)$, *implication*.

Note how A and B above are not part of the language, but are variables in the metalanguage—we will be mostly informal about the metalanguage, i.e., the language we use to describe the logical language.

Language

To simplify the notation, we use a number of abbreviations:

- outermost parentheses are not written: $x \wedge y$ instead of $(x \wedge y)$;
- conjunction and disjunction have a higher precedence over implication: $x \wedge y \supset z \vee w$ instead of $((x \wedge y) \supset (z \vee w))$;
- negation has a higher precedence over conjunction, disjunction, and implication: $\neg x \wedge \neg y$ instead of $((\neg x) \wedge (\neg y))$;
- lowercase letters, unless specified otherwise, stand for variables.
- uppercase letters, unless stated otherwise, stand for objects in the metalanguage.

An important point to remark is that the definition of formula is by induction.

Language

As an example of inductive definition, let's introduce the notion of *subformula*:

Definition 2.3 (Subformula)

Given a formula A on the set V of variables, B is a *subformula* of A if and only if B belongs to the set $S(A)$ inductively defined as

1. if $A \in V$, $A \equiv \top$, or $A \equiv \bot$ then $S(A) = \{A\}$;
2. if $A \equiv B \wedge C$, $A \equiv B \vee C$, or $A \equiv B \supset C$ then $S(A) = \{A\} \cup S(B) \cup S(C)$;
3. if $A \equiv \neg B$ then $S(A) = \{A\} \cup S(B)$.

We may equivalently say that B *occurs* in A, meaning that B is a subformula of A.

In general, the symbol $\equiv$ in the meta-language means 'literally equal', i.e., written in exactly the same way.

Intended interpretation

Informally, a *truth value* is either true or false.

- A variable x stands for some truth value.
- $\top$ denotes true.
- $\bot$ denotes false.
- $A \wedge B$ is true when both A and B are true; and it is false otherwise.
- $A \vee B$ is true when A is true, or B is true, or both are true; and it is false when both A and B are false.
- $A \supset B$ is true if, when A is true, so is B, and it is true also when A is false. It is false when A is true but B is false.
- $\neg A$ is true exactly when A is false.

In general, the truth value of a formula depends on the values of its variables. Sometimes, it happens that a formula is true independently from the value of its variables, e.g., $x \supset x$ is true whatever truth value x may assume.

Logic is mainly concerned in the study of *tautologies*, those formulæ which are true independently from the values of their variables.

Natural deduction

An obvious way to discover whether a formula is true, is to try all the possible values for the variables occurring in it.

But there are three main drawbacks in this strategy:

- the strategy is exponential: if there are n distinct variables in a formula, we have to try 2^n possible assignments.
- the strategy does not scale to other logical systems. For example, take arithmetic: it is unfeasible to show the truth of a formula trying all the possible values for its variables, as each of them stands for a natural!
- the strategy does not provide any insight: we have no idea why the formula holds, except that it exhaustively satisfies all the possible assignments. In particular, we do not know which axioms in our theory are required to make the property true.

What we want is a notion of *proof*: a way to reason that, starting from some basic accepted facts, and adopting a series of accepted rules, allows us to conclude that the formula is true.

Natural deduction

Definition 2.4 (Theory)

Fixed a language, a *theory* T is a set of formulæ, each one usually referred to as an *axiom*.
When $T = \emptyset$, we will speak of the theory as *pure logic*.

Definition 2.5 (Proof)

Fixed a language and a theory T in it, a *proof* or *deduction* of the formula A, the *conclusion*, from a set Γ of formulæ, the *hypotheses* or *assumptions*, is inductively defined by a set of inference rules summarised in the next slides.
A formula A which is the conclusion of a proof with no assumptions, is called a *theorem* in the theory T.

Natural deduction

The inference rules governing conjunctions are:

$$\frac{A \wedge B}{A}\,\wedge\mathrm{E}_1 \qquad \frac{A \wedge B}{B}\,\wedge\mathrm{E}_2 \qquad \frac{A \quad B}{A \wedge B}\,\wedge\mathrm{I}$$

we have two elimination rules, and one introduction rule.

Those governing disjunctions are:

$$\frac{A}{A \vee B}\,\vee\mathrm{I}_1 \qquad \frac{B}{A \vee B}\,\vee\mathrm{I}_2 \qquad \frac{A \vee B \quad \begin{array}{c}[A]\\ \vdots \\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ C\end{array}}{C}\,\vee\mathrm{E}$$

Natural deduction

Implication and negation are subject to the following rules:

$$\dfrac{\begin{array}{c}[A]\\ \vdots\\ B\end{array}}{A \supset B}\supset\mathsf{I} \qquad \dfrac{A \supset B \quad A}{B}\supset\mathsf{E}$$

$$\dfrac{\begin{array}{c}[A]\\ \vdots\\ \bot\end{array}}{\neg A}\neg\mathsf{I} \qquad \dfrac{\neg A \quad A}{\bot}\neg\mathsf{E}$$

They are very similar, since, as we will see in the next lesson, negation can be defined from implication.

Natural deduction

True and false are governed by the following rules:

$$\dfrac{}{\top}\top\mathsf{I} \qquad \dfrac{\bot}{A}\bot\mathsf{E}$$

If A is an axiom of the theory T, i.e., if $A \in T$, we are allowed to deduce it:

$$\dfrac{}{A}\,\mathsf{ax}$$

If A is an assumption, i.e., if $A \in \Gamma$, we can deduce it

$$A$$

Natural deduction

Finally, for every formula A either A is true or it is false. This is expressed by the Law of Excluded Middle:

$$\dfrac{}{A \vee \neg A}\,\mathsf{lem}$$

As we will say later in the course, the Law of Excluded Middle is *delicate*, and it has a special status.

In general, whenever possible, we will try to avoid its use in a proof.

As matter of fact, the same deduction system **without** the Law of Excluded Middle, identifies another logic, *intuitionistic logic*, we will introduce later in the course.

Natural deduction

A couple of comments:

- except for the Law of Excluded Middle, the rules come in pairs: any connective is associated to one or more introduction rule, and one or more elimination rule.
- assumptions may be *free* or *discharged*. Free assumptions are real, in the sense that the proof depends on them; discharged assumptions are used to get rid of a local assumption, which does not affect the whole proof. This is best understood looking at the 'implication introduction' rule: to prove $A \supset B$, we locally assume A, and we try to prove B, but the final result does not depend on A anymore.

When we want to name but not to detail the proof, we write $\pi : \Gamma \vdash_T A$, meaning that π is a proof of A from the assumptions Γ in the theory T. When the proof is not relevant, we omit the π; when the theory is understood or empty, we omit the T; when the set of assumptions is empty, we omit the Γ.

Summary

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathsf{I} \qquad \frac{A \wedge B}{A}\,\wedge\mathsf{E}_1 \qquad \frac{A \wedge B}{B}\,\wedge\mathsf{E}_2 \qquad \frac{\bot}{A}\,\bot\mathsf{E}$$

$$\frac{A}{A \vee B}\,\vee\mathsf{I}_1 \qquad \frac{B}{A \vee B}\,\vee\mathsf{I}_2 \qquad \frac{A \vee B \quad \begin{array}{c}[A]\\ \vdots\\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots\\ C\end{array}}{C}\,\vee\mathsf{E} \qquad \frac{}{\top}\,\top\mathsf{I}$$

$$\frac{\begin{array}{c}[A]\\ \vdots\\ B\end{array}}{A \supset B}\,\supset\mathsf{I} \qquad \frac{A \supset B \quad A}{B}\,\supset\mathsf{E} \qquad \boxed{\frac{}{A \vee \neg A}\,\mathsf{lem}} \qquad \frac{\begin{array}{c}[A]\\ \vdots\\ \bot\end{array}}{\neg A}\,\neg\mathsf{I} \qquad \frac{\neg A \quad A}{\bot}\,\neg\mathsf{E}$$

Summary

This lesson is fundamental. You have to memorise the inference rules of the previous slide and use them at will.

Although the intended meaning seems obvious, be sure to really understand the way we interpret implication.

Take some time to note the symmetries among the inference rules:

- except for the Law of Excluded Middle, there are introduction and elimination rules for every connective;
- you cannot introduce falsity;
- you cannot eliminate truth;
- implication and negation are similar;
- conjunction and disjunction are similar.

References

Natural deduction in its current format has been presented in the classical text *Dag Prawitz*, Natural Deduction, Almqvist & Wiksell, Stockholm, (1965). Recently, this text has been reprinted by Dover.

We will use mainly *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), in this course as a general reference. Although it is an old book, it is still a classical reference, and it contains a complete, formal development of all the notions.

For a comprehensive and deep treatment of natural deduction, see *Anne Sjerp Troelstra* and *Helmut Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43, Cambridge University Press, (1996). This book extends far over the content of our course.

Mathematical Logic

Lecture 3

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Propositional logic:

- Examples
- Proving techniques

Examples

To prove a formula we need to think backwards: so introduction rules eliminate the main connective from a formula.

The first basic technique is to reduce the formula to prove by applying the only introduction rule which could generate it.

Example 3.1

Prove $\vdash A \supset (B \supset A)$

$$\dfrac{\dfrac{[A]^1}{B \supset A}\supset \mathrm{I}}{A \supset (B \supset A)}\supset \mathrm{I}^1$$

Examples

A useful way to help proving a formula is to keep track of the assumptions we generate in the deduction process.

In the last example we started from

$$A \supset (B \supset A)$$

We tried to simplify the goal to prove by the implication introduction rule

$$\dfrac{B \supset A}{A \supset (B \supset A)}\supset \mathrm{I}$$

and in the meanwhile our set of assumptions, which was initially empty, has become $\{A\}$.

We tried to simplify the current goal $B \supset A$, obtaining

$$\dfrac{\dfrac{A}{B \supset A}\supset \mathrm{I}}{A \supset (B \supset A)}\supset \mathrm{I}$$

and in the meanwhile our set of assumptions has become $\{A, B\}$.

Examples

Now we see that the current goal is in the set of available assumptions, so we can close the proof by discharging.

$$\dfrac{\dfrac{[A]^1}{B \supset A}\supset \mathrm{I}}{A \supset (B \supset A)}\supset \mathrm{I}^1$$

It is worth noting that

- we should remember which rule introduced which assumption, so that discharging could be correctly performed;
- we may have unused assumptions, like B in the example.

Examples

When an assumption is a complex formula, it is worth dismounting it by means of an elimination rule.

Example 3.2
Prove $\vdash (A \supset C) \supset ((B \supset C) \supset (A \vee B \supset C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \vee B]^1 \quad \dfrac{[A]^2 \quad [A \supset C]^3}{C}\supset E \quad \dfrac{[B]^2 \quad [B \supset C]^4}{C}\supset E}{C}\vee E^2}{A \vee B \supset C}\supset I^1}{(B \supset C) \supset (A \vee B \supset C)}\supset I^4}{(A \supset C) \supset ((B \supset C) \supset (A \vee B \supset C))}\supset I^3$$

Note how assumptions are local to a subproof. Try to redo this exercise and to understand how assumptions are managed.

Examples

Example 3.3
Prove $\vdash A \vee B = B \vee A$ ($\vee$ is commutative).

We introduced something new: $A = B$ abbreviates $(A \supset B) \wedge (B \supset A)$. To prove such a formula it suffices to prove $A \supset B$ and $B \supset A$.

We note that the property is auto-dual, so it is enough to show

$$\dfrac{\dfrac{[A \vee B]^1 \quad \dfrac{[A]^2}{B \vee A}\vee I_2 \quad \dfrac{[B]^2}{B \vee A}\vee I_1}{B \vee A}\vee E^2}{A \vee B \supset B \vee A}\supset I^1$$

Examples

Example 3.4
Prove $\vdash A \wedge B = B \wedge A$ ($\wedge$ is commutative).

$$\dfrac{\dfrac{\dfrac{[A \wedge B]^1}{B}\wedge E_2 \quad \dfrac{[A \wedge B]^1}{A}\wedge E_1}{B \wedge A}\wedge I}{A \wedge B \supset B \wedge A}\supset I^1$$

Examples

There could be more than one way to prove a result.

Example 3.5
$\vdash A \vee A = A$ ($\vee$ is idempotent).

$$\dfrac{\dfrac{[A \vee A]^1 \quad [A]^2 \quad [A]^2}{A}\vee E^2}{A \vee A \supset A}\supset I^1 \qquad \dfrac{\dfrac{[A]^1}{A \vee A}\vee I_1}{A \supset A \vee A}\supset I^1 \qquad \dfrac{\dfrac{[A]^1}{A \vee A}\vee I_2}{A \supset A \vee A}\supset I^1$$

Examples

Example 3.6
$\vdash A \wedge A = A$ ($\wedge$ is idempotent).

$$\dfrac{\dfrac{[A \wedge A]^1}{A}\,\wedge E_1}{A \wedge A \supset A}\,\supset I^1 \qquad \dfrac{\dfrac{[A \wedge A]^1}{A}\,\wedge E_2}{A \wedge A \supset A}\,\supset I^1 \qquad \dfrac{\dfrac{[A]^1 \quad [A]^1}{A \wedge A}\,\wedge I}{A \supset A \wedge A}\,\supset I^1$$

Examples

Example 3.7
$\vdash A \vee (A \wedge B) = A$ (absorption law).

$$\dfrac{\dfrac{[A \vee (A \wedge B)]^1 \quad [A]^2 \quad \dfrac{[A \wedge B]^2}{A}\,\wedge E_1}{A}\,\vee E^2}{A \vee (A \wedge B) \supset A}\,\supset I^1 \qquad \dfrac{\dfrac{[A]^1}{A \vee (A \wedge B)}\,\vee I_1}{A \supset A \vee (A \wedge B)}\,\supset I^1$$

Examples

Example 3.8
$\vdash A \wedge (A \vee B) = A$ (absorption law).

$$\dfrac{\dfrac{[A \wedge (A \vee B)]^1}{A}\,\wedge E_1}{A \wedge (A \vee B) \supset A}\,\supset I^1 \qquad \dfrac{\dfrac{[A]^1 \quad \dfrac{[A]^1}{A \vee B}\,\vee I_1}{A \wedge (A \vee B)}\,\wedge I}{A \supset A \wedge (A \vee B)}\,\supset I^1$$

Examples

Example 3.9
$\vdash (A \wedge B) \wedge C = A \wedge (B \wedge C)$ ($\wedge$ is associative).

$$\dfrac{\dfrac{\dfrac{[(A \wedge B) \wedge C]^1}{A \wedge B}\,\wedge E_1}{A}\,\wedge E_1 \quad \dfrac{\dfrac{[(A \wedge B) \wedge C]^1}{A \wedge B}\,\wedge E_1}{B}\,\wedge E_2 \quad \dfrac{[(A \wedge B) \wedge C]^1}{C}\,\wedge E_2}{\dfrac{B \wedge C}{A \wedge (B \wedge C)}\,\wedge I}\,\wedge I}{(A \wedge B) \wedge C \supset A \wedge (B \wedge C)}\,\supset I^1$$

$$\dfrac{\dfrac{\dfrac{[A \wedge (B \wedge C)]^1}{A}\,\wedge E_1 \quad \dfrac{\dfrac{[A \wedge (B \wedge C)]^1}{B \wedge C}\,\wedge E_2}{B}\,\wedge E_1}{A \wedge B}\,\wedge I \quad \dfrac{[A \wedge (B \wedge C)]^1}{\dfrac{B \wedge C}{C}\,\wedge E_2}\,\wedge E_2}{(A \wedge B) \wedge C}\,\wedge I}{A \wedge (B \wedge C) \supset (A \wedge B) \wedge C}\,\supset I^1$$

Examples

Falsity elimination allows to deduce any formula one needs. But falsity always comes from a contradiction.

Example 3.10
$\vdash \neg A \supset (A \supset B)$.

$$\dfrac{\dfrac{\dfrac{[\neg A]^1 \quad [A]^2}{\bot}\neg E}{B}\bot E}{A \supset B}\supset I^2}{\neg A \supset (A \supset B)}\supset I^1$$

Examples

Thinking backwards, not introduction allows to assume the conclusion deprived of the negation. It is a form of reasoning by contradiction.

Example 3.11
$\vdash A \wedge B \supset \neg(A \supset \neg B)$.

$$\dfrac{\dfrac{\dfrac{[A \supset \neg B]^1 \quad \dfrac{[A \wedge B]^2}{A}\wedge E_1}{\neg B}\supset E \quad \dfrac{[A \wedge B]^2}{B}\wedge E_2}{\bot}\neg E}{\neg(A \supset \neg B)}\neg I^1}{A \wedge B \supset \neg(A \supset \neg B)}\supset I^2$$

Examples

Example 3.12
$\vdash \neg(A \vee B) = \neg A \wedge \neg B$ (De Morgan's law).

$$\dfrac{\dfrac{[\neg(A \vee B)]^1 \quad \dfrac{[A]^2}{A \vee B}\vee I_1}{\bot}\neg E}{\neg A}\neg I^2 \qquad \dfrac{\dfrac{[\neg(A \vee B)]^1 \quad \dfrac{[B]^3}{A \vee B}\vee I_2}{\bot}\neg E}{\neg B}\neg I^3}{\neg A \wedge \neg B}\wedge I}{\neg(A \vee B) \supset \neg A \wedge \neg B}\supset I^1$$

$$\dfrac{\dfrac{[A \vee B]^1 \quad \dfrac{\dfrac{[A]^2}{\bot}\neg E \quad \dfrac{\dfrac{[\neg A \wedge \neg B]^3}{\neg A}\wedge E_1}{\bot}\neg E}{\bot}\vee E^2 \qquad \dfrac{\dfrac{[B]^2}{\bot}\neg E \quad \dfrac{[\neg A \wedge \neg B]^3}{\neg B}\wedge E_2}{\bot}\neg E}{\bot}}{\neg(A \vee B)}\neg I^1}{\neg A \wedge \neg B \supset \neg(A \vee B)}\supset I^3$$

Examples

Example 3.13
$\vdash \neg\neg(A \wedge B) \supset \neg\neg A \wedge \neg\neg B$.

$$\dfrac{\dfrac{\dfrac{[\neg\neg(A \wedge B)]^1 \quad \dfrac{\dfrac{[\neg A]^2 \quad \dfrac{[A \wedge B]^3}{A}\wedge E_1}{\bot}\neg E}{\neg(A \wedge B)}\neg I^3}{\bot}\neg E}{\neg\neg A}\neg I^2 \qquad \dfrac{\dfrac{[\neg\neg(A \wedge B)]^1 \quad \dfrac{\dfrac{[\neg B]^4 \quad \dfrac{[A \wedge B]^5}{B}\wedge E_2}{\bot}\neg E}{\neg(A \wedge B)}\neg I^5}{\bot}\neg E}{\neg\neg B}\neg I^4}{\neg\neg A \wedge \neg\neg B}\wedge I}{\neg\neg(A \wedge B) \supset \neg\neg A \wedge \neg\neg B}\supset I^1$$

Examples

Example 3.14
$\vdash \neg\neg A \wedge \neg\neg B \supset \neg\neg(A \wedge B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]^1 \quad [B]^2}{A \wedge B}\wedge I \qquad [\neg(A \wedge B)]^3}{\bot}\neg E}{\neg A}\neg I^1 \qquad \dfrac{[\neg\neg A \wedge \neg\neg B]^4}{\neg\neg A}\wedge E_1}{\bot}\neg E \qquad \dfrac{[\neg\neg A \wedge \neg\neg B]^4}{\neg\neg B}\wedge E_2$$

$$\dfrac{\dfrac{\bot}{\neg B}\neg I^2 \qquad \dfrac{[\neg\neg A \wedge \neg\neg B]^4}{\neg\neg B}\wedge E_2}{\bot}\neg E$$

$$\dfrac{\dfrac{\bot}{\neg\neg(A \wedge B)}\neg I^3}{\neg\neg A \wedge \neg\neg B \supset \neg\neg(A \wedge B)}\supset I^4$$

Examples

Example 3.15
$\vdash \neg\neg(A \supset B) \supset (\neg\neg A \supset \neg\neg B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^1 \quad [A \supset B]^2}{B}\supset E \qquad [\neg B]^3}{\bot}\neg E}{\neg(A \supset B)}\neg I^2 \qquad [\neg\neg(A \supset B)]^4}{\bot}\neg E}{\neg A}\neg I^1 \qquad [\neg\neg A]^5}{\bot}\neg E$$

$$\dfrac{\dfrac{\bot}{\neg\neg B}\neg I^3}{\neg\neg A \supset \neg\neg B}\supset I^5$$

$$\dfrac{}{\neg\neg(A \supset B) \supset (\neg\neg A \supset \neg\neg B)}\supset I^4$$

Examples

Example 3.16
$\vdash (\neg\neg A \supset \neg\neg B) \supset \neg\neg(A \supset B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^3 \quad [\neg A]^4}{\bot}\neg E}{\neg\neg A}\neg I^4 \qquad [\neg\neg A \supset \neg\neg B]^2}{\neg\neg B}\supset E \qquad \dfrac{\dfrac{[\neg(A \supset B)]^1 \quad \dfrac{[B]^5}{A \supset B}\supset I}{\bot}\neg E}{\neg B}\neg I^5}{\bot}\neg E}{B}\bot E}{A \supset B}\supset I^3$$

$$\dfrac{\dfrac{[\neg(A \supset B)]^1 \qquad A \supset B}{\bot}\neg E}{\neg\neg(A \supset B)}\neg I^1$$

$$\dfrac{}{(\neg\neg A \supset \neg\neg B) \supset \neg\neg(A \supset B)}\supset I^2$$

Examples

Proofs involving the Law of Excluded Middle are more difficult. The fundamental strategy is that an application of the principle is required when no other strategy could be applied.

Example 3.17
$\vdash A = \neg\neg A$ (double negation law).

$$\dfrac{\dfrac{\dfrac{}{A \vee \neg A}\text{lem} \qquad \dfrac{[A]^1}{\dfrac{\dfrac{[\neg A]^1 \quad [\neg\neg A]^2}{\bot}\neg E}{A}\bot E}\vee E^1}{A}\supset I^2}{\neg\neg A \supset A}$$

$$\dfrac{\dfrac{\dfrac{[\neg A]^1 \quad [A]^2}{\bot}\neg E}{\neg\neg A}\neg I^1}{A \supset \neg\neg A}\supset I^2$$

Examples

Do not rely on the shape of the theorem! Small variations could be provable **without** the Law of Excluded Middle!

Example 3.18
$\vdash \neg A = \neg\neg\neg A$.

$$\dfrac{\dfrac{\dfrac{[\neg A]^2 \quad [A]^3}{\bot}\neg E}{\neg\neg A}\neg I^2 \quad [\neg\neg\neg A]^1}{\dfrac{\bot}{\neg A}\neg I^3}\neg E \qquad \dfrac{\dfrac{[\neg\neg A]^1 \quad [\neg A]^2}{\bot}\neg E}{\neg\neg\neg A}\neg I^1$$
$$\dfrac{}{\neg\neg\neg A \supset \neg A}\supset I^1 \qquad \dfrac{}{\neg A \supset \neg\neg\neg A}\supset I^2$$

Examples

You may think the Law of Excluded Middle is about negation. This is false: there are elementary facts in which negation does not appear that **require** the Law of Excluded Middle to be proved.

Example 3.19
$\vdash (A \supset B) \vee (B \supset A)$.

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{\dfrac{[A]^1}{B \supset A}\supset I}{(A \supset B) \vee (B \supset A)}\vee I_2 \quad \dfrac{\dfrac{\dfrac{[A]^2 \quad [\neg A]^1}{\bot}\neg E}{B}\bot E}{A \supset B}\supset I^2}{(A \supset B) \vee (B \supset A)}\vee I_1 \vee E^1$$

Examples

Example 3.20
$\vdash ((A \supset B) \supset A) \supset A$ (Pierce's law).

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{[A]^1}{A} \quad \dfrac{\dfrac{[(A \supset B) \supset A]^2 \quad \dfrac{\dfrac{\dfrac{[\neg A]^1 \quad [A]^3}{\bot}\neg E}{B}\bot E}{A \supset B}\supset I^3}{A}\supset E}{A}\vee E^1}{((A \supset B) \supset A) \supset A}\supset I^2$$

Examples

Example 3.21
$\vdash A \supset B = \neg B \supset \neg A$ (contraposition).

$$\dfrac{\dfrac{\dfrac{[A \supset B]^1 \quad [A]^2}{B}\supset E \quad [\neg B]^3}{\bot}\neg E}{\neg A}\neg I^2$$
$$\dfrac{}{\neg B \supset \neg A}\supset I^3$$
$$\dfrac{}{(A \supset B) \supset (\neg B \supset \neg A)}\supset I^1$$

$$\dfrac{\dfrac{}{B \vee \neg B}\,\text{lem} \quad \dfrac{\dfrac{\dfrac{[\neg B \supset \neg A]^2 \quad [\neg B]^1}{\neg A}\supset E \quad [A]^3}{\bot}\neg E}{B}\bot E}{B}\vee E^1$$
$$\dfrac{}{A \supset B}\supset I^3$$
$$\dfrac{}{(\neg B \supset \neg A) \supset (A \supset B)}\supset I^2$$

Examples

Example 3.22
$\vdash A \supset B = \neg(A \wedge \neg B)$.

$$\dfrac{\dfrac{\dfrac{[A \supset B]^1 \quad \dfrac{[A \wedge \neg B]^2}{A}\wedge E_1}{B}\supset E \quad \dfrac{[A \wedge \neg B]^2}{\neg B}\wedge E_2}{\bot}\neg E}{\neg(A \wedge \neg B)}\neg I^2}{(A \supset B) \supset \neg(A \wedge \neg B)}\supset I^1$$

$$\dfrac{\dfrac{\dfrac{\overline{B \vee \neg B}\,\text{lem} \quad [B]^1}{B}\vee E^1}{A \supset B}\supset I^2}{\neg(A \wedge \neg B) \supset (A \supset B)}\supset I^3 \qquad \dfrac{\dfrac{[A]^2 \quad [\neg B]^1}{A \wedge \neg B}\wedge I \quad [\neg(A \wedge \neg B)]^3}{\bot}\neg E \quad \dfrac{\bot}{B}\bot E$$

Examples

Example 3.23
$\vdash A \vee B = \neg A \supset B$.

$$\dfrac{\dfrac{\dfrac{[A \vee B]^1 \quad \dfrac{\dfrac{[A]^2 \quad [\neg A]^3}{\bot}\neg E}{B}\bot E \quad [B]^2}{B}\vee E^2}{\neg A \supset B}\supset I^3}{A \vee B \supset (\neg A \supset B)}\supset I^1 \qquad \dfrac{\dfrac{\overline{A \vee \neg A}\,\text{lem} \quad \dfrac{[A]^1}{A \vee B}\vee I_1 \quad \dfrac{[\neg A]^1 \quad [\neg A \supset B]^2}{B}\supset E}{A \vee B}\vee I_2}{(\neg A \supset B) \supset A \vee B}\vee E^1}{}\supset I^2$$

References

Exercises could be found in any standard textbook, see, e.g., Chapter 1 of *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977).

Some further exercises are available on the course web site.

Proving techniques come from the completeness and the normalisation proofs.

Mathematical Logic

Lecture 4

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Propositional logic:

- Semantics: truth tables
- Examples
- Applications
- Soundness

Semantics

The intended meaning of propositional logic can be formalised.
In this way we will get a first, very simple semantics for the syntax introduced in the previous lessons.

Definition 4.1 (Truth-tables semantics)

Fixed a map $v\colon V \to \{0,1\}$ from the set of variables V to the truth values, denoted by 0 and 1, the *meaning* $[\![A]\!]$ of a formula A is inductively defined as:

- if $A \in V$ is a variable then $[\![A]\!] = v(A)$;
- $[\![\top]\!] = 1$;
- $[\![\bot]\!] = 0$;

↪

Semantics

↪ (Truth-tables semantics)

- if $A \equiv B \wedge C$ then $[\![A]\!]$ is calculated according to

$[\![B]\!]$	$[\![C]\!]$	$[\![B \wedge C]\!]$
0	0	0
0	1	0
1	0	0
1	1	1

- if $A \equiv B \vee C$ then $[\![A]\!]$ is calculated according to

$[\![B]\!]$	$[\![C]\!]$	$[\![B \vee C]\!]$
0	0	0
0	1	1
1	0	1
1	1	1

↪

Semantics

↪ (Truth-tables semantics)

- if $A \equiv \neg B$ then $[\![A]\!]$ is calculated according to

$[\![B]\!]$	$[\![\neg B]\!]$
0	1
1	0

- if $A \equiv B \supset C$ then $[\![A]\!]$ is calculated according to

$[\![B]\!]$	$[\![C]\!]$	$[\![B \supset C]\!]$
0	0	1
0	1	1
1	0	0
1	1	1

Semantics

Example 4.2

We can show that the formula $x \wedge y \supset x \vee y$ is true whatever values we may assign to x and y;

$[\![x]\!]$	$[\![y]\!]$	$[\![x \wedge y]\!]$	$[\![x \vee y]\!]$	$[\![x \wedge y \supset x \vee y]\!]$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

The corresponding proofs in natural deduction are:

$$\dfrac{\dfrac{\dfrac{[x \wedge y]^*}{x}\wedge E_1}{x \vee y}\vee I_1}{x \wedge y \supset x \vee y}\supset I^* \qquad \dfrac{\dfrac{\dfrac{[x \wedge y]^*}{y}\wedge E_2}{x \vee y}\vee I_2}{x \wedge y \supset x \vee y}\supset I^*$$

Example

Example 4.3

Truth-tables allow to derive semantic properties, too.

$[\![x]\!]$	$[\![y]\!]$	$[\![x \wedge y]\!]$	$[\![x \vee y]\!]$	$[\![x \wedge y]\!] \leq [\![x \vee y]\!]$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

Applications

Truth tables are widely used in the synthesis of (logical) circuits, and many techniques to minimise the number of electronic gates, each one implementing a logical connective, have been implemented.

In logic truth tables are not an effective way to check whether a formula is true for any assignment of its variables: the number of assignment to try is 2^n, with n the number of variables, so it grows exponentially with respect to the number of variables.

Anyway in pure logic truth tables are a very effective way to construct a minimal set of connectives. Indeed, the collection of connectives is redundant as they can be mutually defined.

Interdependence of connectives

Proposition 4.4

Negation can be defined using implication and falsity.

Proof.

Checking the truth tables, $\neg A$ is equivalent to $A \supset \bot$. □

Proposition 4.5

The set of connectives $\wedge$, $\vee$, and $\neg$ suffice to define all the others.

Proof.

Just checking the truth tables, one can see that

- $\top$ can be defined as $\neg X \vee X$, for any choice of X;
- $\bot$ can be defined as $\neg \top$;
- $A \supset B$ can be defined as $\neg A \vee B$. □

Interdependence of connectives

Proposition 4.6

Conjunction can be defined from disjunction and negation. Moreover, disjunction can be defined from conjunction and negation.

Proof.

Writing down the proof tables it is immediate to see that

- $A \wedge B$ is the same as $\neg(\neg A \vee \neg B)$;
- $A \vee B$ is the same as $\neg(\neg A \wedge \neg B)$. □

Usually $\neg(A \wedge B) = \neg A \vee \neg B$ and $\neg(A \vee B) = \neg A \wedge \neg B$ are referred to as De Morgan's Laws.
Here $A = B$ between two formulæ A and B means that both $A \supset B$ and $B \supset A$ hold, i.e., A and B are equivalent.

Interdependence of connectives

So, the following set of connectives are sufficient to define all the others:

- $\{\bot, \supset\}$;
- $\{\neg, \wedge\}$;
- $\{\neg, \vee\}$;
- $\{\neg, \supset\}$.

But, in principle, one can reduce to a single connective although this is impractical. Define $A \mid B \equiv \neg(A \wedge B)$, which is known as *Sheffer's stroke*. Then using truth tables it is easy to prove

- $\neg A = A \mid A$;
- $A \supset B = A \mid (B \mid B)$.

Soundness

We want to show that every conclusion we may derive in the proof system is true whenever all the assumptions it depends upon are true.

Before stating the theorem and proving it we should make one important remark. The collection of proofs is inductively generated by the inference rules. So we can reason about a provable statement by saying: if A is provable, let π be a proof of A. If a property holds for every proof then it holds for π, too.

To prove that a property holds for every proof, we can prove that each inference rule *preserves* the property, which means that assuming the property to hold for the proofs in the premises of the rule, we have to show that the proof whose last rule is the inference rule under examination, has the property too. In the case of the Soundness Theorem the property of interest is 'the conclusion is true'.

Soundness

Theorem 4.7 (Soundness)

If Γ is a set of formulæ, and we have a proof $\pi : \Gamma \vdash A$ in the natural deduction system then whenever each formula in Γ is true, so is A.

Proof. (i)

The main hypothesis is that, for every $G \in \Gamma$, $[\![G]\!] = 1$.
We proceed by induction on the definition of the proof π, showing that if all the antecedents of an inference rules satisfy the property in the statement, so does the conclusion:

- if π is an instance of the assumption rule then $A \in \Gamma$, so $[\![A]\!] = 1$ by hypothesis.
- if π is an instance of the $\top$I rule then $A \equiv \top$, so $[\![A]\!] = 1$. ↪

Soundness

↪ Proof. (ii)

- if π is an instance of the $\bot$E rule then $[\![\bot]\!] = 1$ by induction hypothesis, but we know by definition that $[\![\bot]\!] = 0$, thus $0 = 1$. Then it follows that $[\![A]\!] = 1$ since $[\![A]\!] \in \{0,1\}$, which is indeed a singleton.
- if π is an instance of the Law of Excluded Middle, $A \equiv B \vee \neg B$. But $[\![B \vee \neg B]\!] = 1$ as it is immediate to check using truth tables.
- if π is an instance of $\neg$I then by the induction hypothesis applied to $\pi' : \Gamma \cup \{A\} \vdash \bot$, we have that $[\![A]\!] = 1$ implies $[\![\bot]\!] = 1$. Then, the contrapositive form of the implication says that $[\![\bot]\!] \neq 1$ implies $[\![A]\!] \neq 1$, which means $[\![\bot]\!] = 0$ implies $[\![A]\!] = 0$. But we know that $[\![\bot]\!] = 0$, so $[\![A]\!] = 0$, that is $[\![\neg A]\!] = 1$.
- if π is an instance of $\neg$E then by the induction hypothesis applied to both antecedents, we get that $[\![\neg A]\!] = 1$ and $[\![A]\!] = 1$. Thus, $0 = [\![A]\!] = 1$. Then $[\![\bot]\!] = 0 = 1$. ↪

Soundness

↪ Proof. (iii)

- if π is an instance of $\wedge$I then $A \equiv B \wedge C$ and by the induction hypothesis applied to both antecedents, $[\![B]\!] = 1$ and $[\![C]\!] = 1$. So by the truth table of conjunction, $[\![B \wedge C]\!] = 1$.
- if π is an instance of $\wedge\mathrm{E}_1$ then the antecedent is a proof of $A \wedge B$ from Γ. Applying the induction hypothesis, we get that $[\![A \wedge B]\!] = 1$, so by the truth table of conjunction we derive that $[\![A]\!] = 1$.
- if π is an instance of $\wedge\mathrm{E}_2$ then the antecedent is a proof of $B \wedge A$ from Γ. Applying the induction hypothesis, we get that $[\![B \wedge A]\!] = 1$, so by the truth table of conjunction we derive that $[\![A]\!] = 1$. ↪

Soundness

↪ Proof. (iv)

- if π is an instance of $\vee\mathrm{I}_1$ then $A \equiv B \vee C$, and the antecedent is a proof of B from Γ. By the induction hypothesis $[\![B]\!] = 1$, so by the truth table of disjunction $[\![B \vee C]\!] = 1$.
- if π is an instance of $\vee\mathrm{I}_2$ then $A \equiv B \vee C$, and the antecedent is a proof of C from Γ. By the induction hypothesis $[\![C]\!] = 1$, so by the truth table of disjunction $[\![B \vee C]\!] = 1$.
- if π is an instance of $\vee$E then applying the induction hypothesis to the first antecedent, we get that $[\![B \vee C]\!] = 1$ for appropriate B and C. Thus by the truth table of disjunction $[\![B]\!] = 1$ or $[\![C]\!] = 1$. In the former case, applying the induction hypothesis to the second antecedent, we get that $[\![A]\!] = 1$. In the latter case, applying the induction hypothesis to the third antecedent, we get that $[\![A]\!] = 1$. ↪

Soundness

↪ Proof. (v)

- if π is an instance of $\supset$I then $A \equiv B \supset C$. If $[\![B]\!] = 0$ then by the truth table of implication, $[\![B \supset C]\!] = 1$. Otherwise $[\![B]\!] = 1$, and we can apply the induction hypothesis to the antecedent of the inference rule obtaining $[\![C]\!] = 1$. Thus by the truth table of implication $[\![B \supset C]\!] = 1$.
- if π is an instance of $\supset$E then applying the induction hypothesis to both antecedents, we get $[\![B \supset A]\!] = 1$ and $[\![B]\!] = 1$. Thus by the truth table of implication it follows that $[\![A]\!] = 1$ too. □

References

The soundness theorem is folklore. Indeed, we will see soon a more interesting and powerful version of it, which uses a more refined semantics.

The interest of the soundness theorem lies in the structure of its proof: most soundness theorems are proved by induction on the structure of proofs, checking that each inference rule preserves the truth of antecedents into the consequence. It is important to become acquainted with this technique.

Mathematical Logic

Lecture 5

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Propositional logic:

- Orders
- Lattices
- Boolean algebras
- Semantics
- Soundness

Orders

A rather more interesting semantics for propositional logic comes from the algebra of orders. In the following, we will develop what is needed to introduce it.

Definition 5.1 (Order)

An *order* $\mathscr{O} = \langle S; \leq \rangle$ is a set S equipped with a binary relation $\leq$ which is

- *reflexive*, i.e., for all $x \in S$, $x \leq x$;
- *anti-symmetric*, i.e., for all $x, y \in S$, when $x \leq y$ and $y \leq x$ then $x = y$;
- *transitive*, i.e., for all $x, y, z \in S$, if $x \leq y$ and $y \leq z$ then $x \leq z$.

Noting that if $\mathscr{O} = \langle S; \leq \rangle$ is an order, so is $\mathscr{O}^{\mathrm{op}} = \langle S; \geq \rangle$ we have a *duality principle*: when a property holds for all orders, its instance on the opposite order generates a *dual* property, which holds for all orders, too.

Orders

Definition 5.2 (Least upper bound)

Fixed an order $\mathcal{O} = \langle S; \leq \rangle$ and a $U \subseteq S$, we call the element $m \in S$, if it exists, the *least upper bound* (lub), or *supremum*, or *join* of U whenever

- for every $x \in U$, $x \leq m$;
- for each $w \in S$ such that $x \leq w$ for every $x \in U$, $m \leq w$.

Definition 5.3 (Greatest lower bound)

Fixed an order $\mathcal{O} = \langle S; \leq \rangle$ and a $U \subseteq S$, we call the element $m \in S$, if it exists, the *greatest lower bound* (glb), or *infimum*, or *meet* of U whenever

- for every $x \in U$, $m \leq x$;
- for each $w \in S$ such that $w \leq x$ for every $x \in U$, $w \leq m$.

Observe how the two notions are dual.

Lattices

Definition 5.4 (Lattice)

An order $\mathcal{O} = \langle S; \leq \rangle$ is called a *lattice* when, for every pair $x, y \in S$ there exists the join of $\{x, y\}$, denoted by $x \vee y$, and there exists the meet of $\{x, y\}$, denoted by $x \wedge y$.
Moreover, a lattice is said to be *bounded* when, for every finite $U \subseteq S$, there is $\bigvee U$, the join of U, and $\bigwedge U$, the meet of U. Conventionally, $\bigvee \emptyset$ is denoted by $\bot$, *bottom*, and $\bigwedge \emptyset$ is denoted by $\top$, *top*.

Observe how lattices preserve duality.

Lattices

Proposition 5.5

In a bounded lattice $\langle S; \leq \rangle$ *every element is greater that* $\bot$ *and less than* $\top$.

Proof.

By duality, it suffices to prove just one part of the statement.

Since $\top = \bigwedge \emptyset$, by definition of meet for any $y \in S$ such that $y \leq x$ for all $x \in \emptyset$, it holds $y \leq \top$. But there are no elements in $\emptyset$, so $y \leq \top$ for any $y \in S$. □

Observe how these properties *uniquely characterise* $\top$ and $\bot$.

Proposition 5.6

In a bounded lattice $\langle S; \leq \rangle$ $\bigvee S = \top$ *and* $\bigwedge S = \bot$.

Proof.

By definition of join for every $x \in S$, $x \leq \bigvee S$, and by Proposition 5.5 $\top$ is such that for all $x \in S$, $x \leq \top$. So, $\top \leq \bigvee S$ and $\bigvee S \leq \top$. By anti-symmetry $\bigvee S = \top$. The other part follows by duality. □

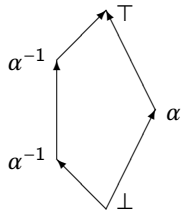
Lattices

Definition 5.7 (Complemented lattice)

A bounded lattice $\mathcal{O} = \langle S; \leq \rangle$ is said to be *complemented* when for each element $x \in S$, there is an element $y \in S$ such that

- $x \wedge y = \bot$;
- $x \vee y = \top$.

The element y is not necessarily unique. For example



Lattices

Definition 5.8 (Distributive lattice)

A lattice $\mathcal{O} = \langle S; \leq \rangle$ is said to be *distributive* when for every $x, y, z \in S$,

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \ .$$

Proposition 5.9

In every lattice, $x \wedge y = y \wedge x$ and $x \vee y = y \vee x$.

Proof.

Immediate by definition of meet and join. □

Lattices

Proposition 5.10

For each x in a bounded lattice $x = x \wedge \top$ and $x = x \vee \bot$.

Proof.

Immediate by definition of meet and join, and Proposition 5.5. □

Proposition 5.11 (Absorption)

For each x and y in a lattice $x \vee (x \wedge y) = x$ and $x \wedge (x \vee y) = x$.

Proof.

By definition of join $x \leq x \vee (x \wedge y)$, so it suffices to show $x \vee (x \wedge y) \leq x$.

But $x \leq x$ by reflexivity, and $x \wedge y \leq x$ by definition of meet, so $x \vee (x \wedge y) \leq x$ by definition of join. The other part follows by duality. □

Lattices

Proposition 5.12

In any distributive lattice for all x, y, and z, $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.

Proof.

$$\begin{aligned}
& (x \vee y) \wedge (x \vee z) && \\
&= ((x \vee y) \wedge x) \vee ((x \vee y) \wedge z) && \text{distributivity} \\
&= (x \wedge x) \vee (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) && \text{distributivity twice} \\
&= x \vee (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) && \text{idempotence} \\
&= x \vee (x \wedge z) \vee (y \wedge z) && \text{absorption} \\
&= x \vee (y \wedge z) && \text{absorption}
\end{aligned}$$

□

Associativity and commutativity of $\wedge$ and $\vee$ are silently applied.

Lattices

Proposition 5.13

In any bounded distributive complemented lattice each element x has a unique complement, denoted by $\neg x$.

Proof.

Suppose the element x has two complements y and z. Then by definition of complement: $x \wedge y = \bot = x \wedge z$ and $x \vee y = \top = x \vee z$. Thus

$$\begin{aligned}
& y \\
&= y \wedge \top \\
&= y \wedge (x \vee z) \\
&= (y \wedge x) \vee (y \wedge z) \\
&= (z \wedge x) \vee (z \wedge y) \\
&= z \wedge (x \vee y) \\
&= z \wedge \top \\
&= z \ .
\end{aligned}$$

□

Boolean algebras

Definition 5.14 (Boolean algebra)

A *Boolean algebra* is a bounded distributive complemented lattice.

Example 5.15

The set $\{0,1\}$ with the ordering $0 \leq 1$ is a Boolean algebra with $\top = 1$ and $\bot = 0$. This is the structure supporting the truth-table semantics.

Example 5.16

Fixed a set U, the powerset $\wp(U) = \{S : S \subseteq U\}$ ordered by inclusion is a Boolean algebra. The complement of S is the difference $U \setminus S$, while $\wedge$ is the intersection, and $\vee$ is the union.

Example 5.17

Let $n \in \mathbb{N}$ be such that it cannot be divided by the square of any other number, e.g., $105 = 3 \cdot 5 \cdot 7$. Then the divisors of n form a Boolean algebra with the operations of greatest common divisor, least common multiple, and the complement of x being n/x.

Semantics

We introduced Boolean algebra for a precise purpose: interpreting propositional logic.

Definition 5.18 (Semantics)

Fixed a Boolean algebra $\mathscr{O} = \langle O; \leq \rangle$ and $v : V \to O$ mapping each variable into an element of the algebra, the interpretation $[\![A]\!]$ of a formula A is inductively defined as:

- if A is a variable, $[\![A]\!] = v(A)$;
- if $A \equiv \top$, $[\![A]\!] = \top$, the maximum element of $\mathscr{O}$;
- if $A \equiv \bot$, $[\![A]\!] = \bot$, the minimum element of $\mathscr{O}$;
- if $A \equiv B \wedge C$, $[\![A]\!] = [\![B]\!] \wedge [\![C]\!]$, the meet of the interpretations of conjuncts;
- if $A \equiv B \vee C$, $[\![A]\!] = [\![B]\!] \vee [\![C]\!]$, the join of the interpretations of disjuncts;
- if $A \equiv B \supset C$, $[\![A]\!] = \neg[\![B]\!] \vee [\![C]\!]$, that is $[\![A]\!] = [\![\neg B \vee C]\!]$ interpreting implication as a *relative complement*;
- if $A \equiv \neg B$, $[\![A]\!] = \neg[\![B]\!]$, the complement of the interpretation of B.

Soundness

Definition 5.19 (Validity)

A formula A is *valid* or *true* in a Boolean algebra $\mathscr{O} = \langle O; \leq \rangle$ together with an interpretation $v : V \to O$ of variables when $[\![A]\!] = \top$.
A set of formulæ is *valid* or *true* when each formula in the set is valid. The pair $(\mathscr{O}, v)$ is called a *model* for a theory T when it makes true all the formulæ in T.

Theorem 5.20 (Soundness)

In any model $(\mathscr{O} = \langle O; \leq \rangle, v : V \to O)$ *for the theory* T *and the assumptions in the set* Δ*, if* $\pi : \Delta \vdash_T A$ *then* A *is valid.*

Soundness

Proof. (i)

The proof is by induction on the structure of π: we show that the interpretation of the conclusion A is greater than $\bigwedge_{G \in \Gamma} [\![G]\!]$, with Γ the finite set of assumptions occurring in the proof of A:

- if π is a proof by assumption then $A \in \Gamma$ and by definition of $\wedge$, $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!]$.
- if π is a proof by axiom, then $A \in T$ and by hypothesis, $[\![A]\!] = \top$, so $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!]$ by definition of $\top$.
- if π is an instance of the Law of Excluded Middle then $A \equiv B \vee \neg B$, and $[\![A]\!] = [\![B \vee \neg B]\!] = [\![B]\!] \vee \neg[\![B]\!] = \top$ by definition of complement in a Boolean algebra. Thus $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!] = \top$ by definition of $\top$.
- if π is an instance of $\top$-introduction then $A \equiv \top$, so $[\![A]\!] = [\![\top]\!] = \top$. Thus $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!] = \top$ by definition of $\top$. ↪

Soundness

↪ Proof. (ii)

- if π is an instance of $\bot$-elimination then by induction hypothesis $\bot \leq \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![\bot]\!] = \bot$. Thus by anti-symmetry, $\bigwedge_{G\in\Gamma}[\![G]\!] = \bot$. So, by definition of $\bot$, $\bot = \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$.
- if π is an instance of $\wedge$-introduction then $A \equiv B \wedge C$, and by induction hypothesis twice, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!]$ and $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![C]\!]$. Thus by definition of $\wedge$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!] \wedge [\![C]\!] = [\![B \wedge C]\!] = [\![A]\!]$.
- if π is an instance of $\wedge_1$-elimination then by induction hypothesis for some formula B, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A \wedge B]\!] = [\![A]\!] \wedge [\![B]\!]$. Thus by definition of $\wedge$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$.
- if π is an instance of $\wedge_2$-elimination then by induction hypothesis for some formula B, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B \wedge A]\!] = [\![B]\!] \wedge [\![A]\!]$. Thus by definition of $\wedge$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$. ↪

Soundness

↪ Proof. (iii)

- if π is an instance of $\vee_1$-introduction then $A \equiv B \vee C$ and by induction hypothesis $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!]$. So by definition of $\vee$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!] \leq [\![B]\!] \vee [\![C]\!] = [\![B \vee C]\!] = [\![A]\!]$.
- if π is an instance of $\vee_2$-introduction then $A \equiv B \vee C$ and by induction hypothesis $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![C]\!]$. So by definition of $\vee$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![C]\!] \leq [\![B]\!] \vee [\![C]\!] = [\![B \vee C]\!] = [\![A]\!]$.
- if π is an instance of $\vee$-elimination then by induction hypothesis for some formulæ B and C, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B \vee C]\!] = [\![B]\!] \vee [\![C]\!]$, $[\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$, and $[\![C]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$.
  It follows that by definition of $\vee$ and distributing, $([\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!]) \vee ([\![C]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!]) = ([\![B]\!] \vee [\![C]\!]) \wedge \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$.
  But since $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!] \vee [\![C]\!]$, $([\![B]\!] \vee [\![C]\!]) \wedge \bigwedge_{G\in\Gamma}[\![G]\!] = \bigwedge_{G\in\Gamma}[\![G]\!]$ by definition of $\wedge$, so $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$. ↪

Soundness

↪ Proof. (iv)

- if π is an instance of $\supset$-introduction then $A \equiv B \supset C$ for some formulæ B and C. By induction hypothesis, $[\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![C]\!]$.
  So by definition of $\vee$, $[\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!] \leq \neg[\![B]\!] \vee [\![C]\!]$. Evidently $\neg[\![B]\!] \leq \neg[\![B]\!] \vee [\![C]\!]$. Thus by definition of $\vee$, $[\![A]\!] = [\![B \supset C]\!] = \neg[\![B]\!] \vee [\![C]\!] \geq \neg[\![B]\!] \vee ([\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!])$. Distributing and by definition of complement, $[\![A]\!] \geq (\neg[\![B]\!] \vee [\![B]\!]) \wedge (\neg[\![B]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!]) = \top \wedge (\neg[\![B]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!]) = \neg[\![B]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!]$. By definition of $\vee$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq \neg[\![B]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$.
- if π is an instance of $\supset$-elimination then for some formula B, by induction hypothesis twice, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B \supset A]\!]$ and $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!]$.
  By definition of $\wedge$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B \supset A]\!] \wedge [\![B]\!]$. But $[\![B \supset A]\!] = \neg[\![B]\!] \vee [\![A]\!]$.
  So $\bigwedge_{G\in\Gamma}[\![G]\!] \leq (\neg[\![B]\!] \vee [\![A]\!]) \wedge [\![B]\!]$. Distributing and by definition of $\neg$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq (\neg[\![B]\!] \wedge [\![B]\!]) \vee ([\![A]\!] \wedge [\![B]\!]) = \bot \vee ([\![A]\!] \wedge [\![B]\!]) = [\![A]\!] \wedge [\![B]\!] \leq [\![A]\!]$. ↪

Soundness

↪ Proof. (v)

- if π is an instance of $\neg$-introduction then $A \equiv \neg B$ for some formula B. So by induction hypothesis $[\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![\bot]\!] = \bot$. Thus by definition of $\bot$ and anti-symmetry, $[\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!] = \bot$. Then $[\![A]\!] = [\![\neg B]\!] = \neg[\![B]\!] = \neg[\![B]\!] \vee \bot = \neg[\![B]\!] \vee ([\![B]\!] \wedge \bigwedge_{G\in\Gamma}[\![G]\!])$, and distributing, $[\![A]\!] = (\neg[\![B]\!] \vee [\![B]\!]) \wedge (\neg[\![B]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!]) = \top \wedge (\neg[\![B]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!]) = [\![A]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!]$. Thus by definition of $\vee$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!] \vee \bigwedge_{G\in\Gamma}[\![G]\!] = [\![A]\!]$.
- if π is an instance of $\neg$-elimination then $A \equiv \bot$ and by induction hypothesis twice, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![\neg B]\!]$ and $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![B]\!]$. But $[\![\neg B]\!] = \neg[\![B]\!]$. So by definition of $\wedge$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq \neg[\![B]\!] \wedge [\![B]\!]$. By definition of complement, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq \neg[\![B]\!] \wedge [\![B]\!] = \bot = [\![A]\!]$.

Hence, for every formula A being the conclusion of a proof in the theory T from Γ, the finite set of assumptions really occurring in the proof, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$. But by hypothesis for every $G \in \Delta$, $[\![G]\!] = \top$, and $\Gamma \subseteq \Delta$ so $\bigwedge_{G\in\Gamma}[\![G]\!] = \top$, thus by definition of $\top$, $\top \leq [\![A]\!] \leq \top$, that is, by anti-symmetry, $[\![A]\!] = \top$. □

References

Boolean algebras, in the form of the powerset of a set, have been introduced for the first time in *George Boole*, An Investigation of the Laws of Thought, Prometheus Books, (2003), reprint from the original edition (1854).

Two excellent references for orders, lattices, and Boolean algebras are *Brian A. Davey* and *Hilary Ann Priestley*, Introduction to Lattices and Order, Cambridge University Press, (2002), and *George Grätzer*, General Lattice Theory, second edition, Birkhäuser, (1996).

The idea of the proof of the Soundness Theorem is folklore: indeed, the proof itself is adapted from a more general result which uses the internal logic of a Boolean topos. This is an advanced topic, which will not be covered in the course, and the interested student can give a glimpse to *Peter Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002).

Syllabus

Propositional logic:

- Completeness

Mathematical Logic

Lecture 6

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Completeness

We will show that, fixed a theory T, any formula A which is valid in every Boolean algebra making T true, is provable, i.e., there is a natural deduction derivation with no assumptions that has A as its conclusion.

Indeed, we will prove a stronger result: in a theory T for any finite set Γ of formulæ and for any formula A, if $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$ in every Boolean algebra which makes the theory T true, there is a natural deduction proof $\pi\colon \Gamma \vdash_T A$.

As a corollary, noting that when $\Gamma = \emptyset$, $\bigwedge_{G\in\Gamma}[\![G]\!] = \top$, the previous result follows by anti-symmetry.

Preliminaries

The proof is complex and subtle.

In the first place, it is worth noting that if $\pi\colon \Gamma \vdash_T A$ then there is a finite $\Delta \subseteq \Gamma$ such that $\pi\colon \Delta \vdash_T A$. Indeed, since any proof is a finite object and any inference rule has a finite number of antecedents, only a finite number of assumptions may be used in a proof.

In this sense the limit of having a finite Γ in the statement of the Completeness Theorem is not committing.

Strategy

Of course, the difficult aspect of the theorem lies in considering the totality of Boolean algebras.

The strategy behind the proof is

- construct a *canonical* Boolean algebra $\mathbb{B}$ which makes the axioms of T true and which is 'easy' to manage;
- prove that, for any finite set Γ of formulæ and for any formula A, if $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$ in $\mathbb{B}$ then there exists $\pi\colon \Gamma \vdash_T A$;
- since the antecedent holds in every Boolean algebra, we deduce completeness.

This strategy is general: many completeness results for most logical systems follow this pattern. But there are exceptions...

Canonical model

The idea is to define a *canonical Boolean algebra* in which truth and provability are the same notion.

Definition 6.1 (Canonical Boolean algebra)

Let T be a theory. Then the *canonical Boolean algebra* $\mathbb{B}(T)$ on T is the pair $\langle \{A\colon A \text{ is a formula in the language of } T\}/{\sim}; \leq_{\mathbb{B}(T)}\rangle$, where

- $A \sim B$ if and only if $A \vdash_T B$ and $B \vdash_T A$,
- $[A]_\sim \leq_{\mathbb{B}(T)} [B]_\sim$ exactly when $A \vdash_T B$.

For the sake of simplicity, when it is clear from the context, we omit the subscripts. Also, observe how $[A]_\sim \leq_{\mathbb{B}(T)} [B]_\sim$ and $[B]_\sim \leq_{\mathbb{B}(T)} [A]_\sim$ implies $[A]_\sim = [B]_\sim$.

But we have to show first that $\mathbb{B}(T)$ is a Boolean algebra.

An auxiliary result

Lemma 6.2

If $\pi\colon \Gamma \cup \{A\} \vdash_T B$ and $\theta\colon \Gamma \vdash_T A$ then there is a proof $\nu\colon \Gamma \vdash_T B$.

Proof. (i)

By induction on the structure of the proof π.

- if π is an instance of the assumption rule either $B \in \Gamma$, so ν coincides with π which does not depend on A, or $B \equiv A$ thus $\nu = \theta$.
- if π is an instance of the axiom rule, $B \in T$, so $\nu = \pi$ which does not depend on A.
- if π is an instance of $\top$-introduction, $B \equiv \top$, so $\nu = \pi$ which does not depend on A.
- if π is an instance of $\bot$-elimination, by induction hypothesis there is $\xi\colon \Gamma \vdash_T \bot$, so applying the $\bot$-elimination rule to ξ gives the required ν.↪

An auxiliary result

↪ Proof. (ii)

- if π is an instance of the Law of Excluded Middle, $B \equiv C \vee \neg C$, so $\nu = \pi$ which does not depend on A.
- if π is an instance of $\wedge$-introduction, $B \equiv C \wedge D$, and by induction hypothesis there are $\xi\colon \Gamma \vdash_T C$ and $\mu\colon \Gamma \vdash_T D$, so the required ν is obtained by applying $\wedge$-introduction to ξ and μ.
- if π is an instance of $\wedge_1$-elimination, by induction hypothesis there is $\xi\colon \Gamma \vdash_T B \wedge C$, so ν is obtained by applying $\wedge_1$-elimination to ξ.
- if π is an instance of $\wedge_2$-elimination, by induction hypothesis there is $\xi\colon \Gamma \vdash_T C \wedge B$, so ν is obtained by applying $\wedge_2$-elimination to ξ. ↪

An auxiliary result

↪ Proof. (iii)

- if π is an instance of $\vee_1$-introduction then $B \equiv C \vee D$, and by induction hypothesis there is $\xi\colon \Gamma \vdash_T C$, so ν is obtained by applying $\vee_1$-introduction to ξ.
- if π is an instance of $\vee_2$-introduction, then $B \equiv C \vee D$, and by induction hypothesis there is $\xi\colon \Gamma \vdash_T D$, so ν is obtained by applying $\vee_2$-introduction to ξ.
- if π is an instance of $\vee$-elimination, by induction hypothesis there are $\xi\colon \Gamma \vdash_T C \vee D$, $\mu_C\colon \Gamma \cup \{C\} \vdash_T B$ and $\mu_D\colon \Gamma \cup \{D\} \vdash_T B$, so applying $\vee$-elimination to ξ, μ_C, and μ_D the required ν is constructed. ↪

An auxiliary result

↪ Proof. (iv)

- if π is an instance of $\supset$-introduction then $B \equiv C \supset D$, and by induction hypothesis there is $\xi\colon \Gamma \cup \{C\} \vdash_T D$, so ν is obtained by applying $\supset$-introduction to ξ.
- if π is an instance of $\supset$-elimination, by induction hypothesis there are $\xi\colon \Gamma \vdash_T C \supset B$ and $\mu\colon \Gamma \vdash_T C$, so ν is constructed applying $\supset$-elimination to ξ and μ.
- if π is an instance of $\neg$-introduction, $B \equiv \neg C$, and by induction hypothesis there is $\xi\colon \Gamma \cup \{C\} \vdash_T \bot$, thus ν is obtained applying $\neg$-introduction to ξ.
- if π is an instance of $\neg$-elimination, by induction hypothesis there are $\xi\colon \Gamma \vdash_T \neg C$ and $\mu\colon \Gamma \vdash_T C$, so ν is constructed applying $\neg$-elimination to ξ and μ. □

Properties of the canonical model

Proposition 6.3

The relation $\sim$ is an equivalence relation.

Proof.

- By the assumption inference rule $A \vdash_T A$, so $A \sim A$ for any formula A, i.e., $\sim$ is reflexive.
- If $A \sim B$ then $A \vdash_T B$ and $B \vdash_T A$, so $B \sim A$ too. That is, $\sim$ is symmetric.
- If $A \sim B$ and $B \sim C$ then there are $\pi_B\colon A \vdash_T B$ and $\pi_A\colon B \vdash_T A$, and $\theta_C\colon B \vdash_T C$ and $\theta_B\colon C \vdash_T B$. By Lemma 6.2 there are $\pi\colon A \vdash_T C$ and $\theta\colon C \vdash_T A$, that is, $A \sim C$, which means $\sim$ is transitive. □

Properties of the canonical model

Proposition 6.4

The relation $\leq_{\mathbb{B}(T)}$ is an ordering.

Proof.

- The relation $[A]_\sim \leq [B]_\sim$ does not depend on the choices of the representatives in the equivalence classes on $\sim$, indeed if $[A] = [A']$ and $[B] = [B']$ then $A \sim A'$ and $B \sim B'$. So by definition of $\sim$, $A' \vdash_T A$ and $B \vdash_T B'$. But by definition of $\leq$, $A \vdash_T B$, thus by Lemma 6.2 twice, $A' \vdash_T B'$, that is $[A'] \leq [B']$.
- By the assumption rule $A \vdash_T A$, so $[A] \leq [A]$, i.e., $\leq$ is reflexive.
- If $[A] \leq [B]$ and $[B] \leq [C]$ then $A \vdash_T B$ and $B \vdash_T C$, so by Lemma 6.2, $A \vdash_T C$, that is $[A] \leq [C]$, i.e., $\leq$ is transitive.
- If $[A] \leq [B]$ and $[B] \leq [A]$ then $A \vdash_T B$ and $B \vdash_T A$, so by definition of $\sim$, $A \sim B$, that is, $[A] = [B]$, i.e., $\leq$ is anti-symmetric. □

Properties of the canonical model

Proposition 6.5

$\mathbb{B}(T)$ *is a lattice.*

Proof.

- Consider $[A \wedge B]$: $[A \wedge B] \leq [A]$ since $A \wedge B \vdash_T A$ by $\wedge_1$-elimination; also, $[A \wedge B] \leq [B]$ since $A \wedge B \vdash_T B$ by $\wedge_2$-elimination.
  If $[C] \leq [A]$ and $[C] \leq [B]$ then $C \vdash_T A$ and $C \vdash_T B$, so $C \vdash_T A \wedge B$ by $\wedge$-introduction, thus $[C] \leq [A \wedge B]$. So by definition of $\wedge$ in an order, $[A] \wedge [B] = [A \wedge B]$.
- Consider $[A \vee B]$: $[A] \leq [A \vee B]$ since $A \vdash_T A \vee B$ by $\vee_1$-introduction; also, $[B] \leq [A \vee B]$ since $B \vdash_T A \vee B$ by $\vee_2$-introduction.
  If $[A] \leq [C]$ and $[B] \leq [C]$ then $A \vdash_T C$ and $B \vdash_T C$, so $A \vee B \vdash_T C$ by $\vee$-elimination, thus $[A \vee B] \leq [C]$. So by definition of $\vee$ in an order, $[A] \vee [B] = [A \vee B]$. □

Properties of the canonical model

Proposition 6.6

$\mathbb{B}(T)$ *is a bounded lattice.*

Proof.

- For each formula A, $A \vdash_T \top$ by $\top$-introduction, so $[A] \leq [\top]$. Thus by definition of $\top$ in a lattice, $\top = [\top]$.
- For each formula A, $\bot \vdash_T A$ by $\bot$-elimination, so $[\bot] \leq [A]$. Thus by definition of $\bot$ in a lattice, $\bot = [\bot]$. □

Properties of the canonical model

Proposition 6.7

$\mathbb{B}(T)$ *is a distributive lattice.*

Proof. (i)

For any A, B, and C, $[A] \vee ([B] \wedge [C]) = [A] \vee [B \wedge C] = [A \vee (B \wedge C)]$ and $([A] \vee [B]) \wedge ([A] \vee [C]) = [A \vee B] \wedge [A \vee C] = [(A \vee B) \wedge (A \vee C)]$.

But $A \vee (B \wedge C) \vdash_T (A \vee B) \wedge (A \vee C)$ since

$$\dfrac{A \vee (B \wedge C) \quad \dfrac{\dfrac{[A]^*}{A \vee B}\vee I_1 \quad \dfrac{[A]^*}{A \vee C}\vee I_1}{(A \vee B) \wedge (A \vee C)}\wedge I \quad \dfrac{\dfrac{\dfrac{[B \wedge C]^*}{B}\wedge E_1}{A \vee B}\vee I_2 \quad \dfrac{\dfrac{[B \wedge C]^*}{C}\wedge E_2}{A \vee C}\vee I_2}{(A \vee B) \wedge (A \vee C)}\wedge I}{(A \vee B) \wedge (A \vee C)}\vee E^*$$

↪

Properties of the canonical model

↪ Proof. (ii)

Also $(A \vee B) \wedge (A \vee C) \vdash_T A \vee (B \wedge C)$ since

$$\dfrac{\dfrac{(A \vee B) \wedge (A \vee C)}{A \vee B}\,{\wedge}\mathrm{E}_1 \quad \dfrac{\dfrac{[A]^*}{A \vee (B \wedge C)}\,{\vee}\mathrm{I}_1 \quad \begin{array}{c}[B]^*\\ \vdots\\ A \vee (B \wedge C)\end{array}}{A \vee (B \wedge C)}\,{\vee}\mathrm{E}^*$$

where the third antecedent is

$$\dfrac{\dfrac{(A \vee B) \wedge (A \vee C)}{A \vee C}\,{\wedge}\mathrm{E}_2 \quad \dfrac{\dfrac{[A]^\dagger}{A \vee (B \wedge C)}\,{\vee}\mathrm{I}_1 \quad \dfrac{\dfrac{B \quad [C]^\dagger}{B \wedge C}\,{\wedge}\mathrm{I}}{A \vee (B \wedge C)}\,{\vee}\mathrm{I}_2}{A \vee (B \wedge C)}\,{\vee}\mathrm{E}^\dagger$$

Thus $(A \vee B) \wedge (A \vee C) \sim A \vee (B \wedge C)$ and the conclusion follows. □

Properties of the canonical model

Proposition 6.8

$\mathbb{B}(T)$ is a complemented lattice.

Proof.

Consider $[\neg A]$ for any formula A: $[A] \wedge [\neg A] = [A \wedge \neg A] = [\bot] = \bot$ since $\bot \vdash_T A \wedge \neg A$ by $\bot$-elimination, and

$$\dfrac{\dfrac{A \wedge \neg A}{A}\,{\wedge}\mathrm{E}_1 \quad \dfrac{A \wedge \neg A}{\neg A}\,{\wedge}\mathrm{E}_2}{\bot}\,\neg\mathrm{E}$$

Also, $[A] \vee [\neg A] = [A \vee \neg A] = [\top] = \top$ since $A \vee \neg A \vdash_T \top$ by $\top$-introduction and $\top \vdash_T A \vee \neg A$ by the Law of Excluded Middle. □

Corollary 6.9

$\mathbb{B}(T)$ is a Boolean algebra.

Classifying models

Proposition 6.10

Fixed a theory T let $(\mathbb{O}, v)$ be a model of T. If $[B]_\sim \leq_{\mathbb{B}(T)} [C]_\sim$ then $[\![B]\!]_{\mathbb{O}} \leq_{\mathbb{O}} [\![C]\!]_{\mathbb{O}}$.

Proof.

If $[B]_\sim \leq_{\mathbb{B}(T)} [C]_\sim$ then there is $\pi\colon B \vdash_T C$ by definition of $\leq_{\mathbb{B}(T)}$.
Thus by the proof of the Soundness Theorem 5.20 applied in the $\mathbb{O}$ Boolean algebra with the v assignment, $[\![B]\!]_{\mathbb{O}} \leq_{\mathbb{O}} [\![C]\!]_{\mathbb{O}}$. □

Classifying models

Definition 6.11 (Canonical map)

Fixed a theory T let $(\mathbb{O}, v)$ be a model of T. Then the map $\xi_{\mathbb{O}}\colon \mathbb{B} \to \mathbb{O}$ defined by $[B]_\sim \mapsto [\![B]\!]_{\mathbb{O}}$ is the *canonical map* to $\mathbb{O}$.

This definition does not depend on the choice of the representatives in $\mathbb{B}$. Indeed if $[A] = [A']$ then, $[A] \leq [A']$ and $[A'] \leq [A]$, so by Proposition 6.10, $[\![A]\!] \leq [\![A']\!]$ and $[\![A']\!] \leq [\![A]\!]$ in $\mathbb{O}$, thus by anti-symmetry, $[\![A]\!] = [\![A']\!]$.

Moreover the canonical map preserves the ordering of $\mathbb{B}$.

Completeness

Theorem 6.12 (Completeness)

Fixed a theory T, for any finite set Γ of formulæ and for any formula A if $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$ in any model of T then there is a natural deduction proof $\pi\colon \Gamma \vdash_T A$.

Proof.

If $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$, then $[\![\bigwedge_{G\in\Gamma} G]\!] \leq [\![A]\!]$ being Γ finite.
Since this fact holds in any Boolean algebra, it holds also in $\mathbb{B}(T)$, the canonical Boolean algebra on T. And because of the way interpretation is defined in $\mathbb{B}(T)$, $[\bigwedge_{G\in\Gamma} G] \leq [A]$.
So by definition of $\leq$ in $\mathbb{B}(T)$ there is $\pi\colon \bigwedge_{G\in\Gamma} G \vdash_T A$. Noting that $\Gamma \vdash_T \bigwedge_{G\in\Gamma} G$ by iterating the $\wedge$-introduction rule, by Proposition 6.2 it follows $\Gamma \vdash_T A$. □

Completeness

Corollary 6.13

If $[\![A]\!] = \top$ in every model of T then there is a proof $\pi\colon \vdash_T A$.

Proof.

If $[\![A]\!] = \top$ then $\top = [\![\top]\!] \leq [\![A]\!]$, being $\leq$ reflexive. By the Completeness Theorem 6.12 the result follows immediately. □

Classifying models

In fact we have another result for free: any *model* for a theory T, i.e., any Boolean algebra $\mathbb{O}$ together with an assignment of variables is described by its canonical map $\xi_{\mathbb{O}}$.

In a sense, all the models of a theory T can be synthesised from the canonical model applying a canonical map. It is tempting to identify the models with the class of canonical maps. . .

. . . but this is another story which leads very far. And we will not pursue it during this course. We just observe that, when there is a classifying model, then we can limit the study to the classifying model to analyse properties, like completeness, that hold in every model.

References

The proof has been adapted from the one in topos theory, which is illustrated in Section D of *Peter Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, Oxford Logic Guides 43, Oxford University Press, (2003).

The notion of classifying model is central in the topos-theoretic approach and in some way it goes back to Grothendieck's work. Again, Johnstone's book is a good starting point.

Mathematical Logic

Lecture 7

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

First order logic:

- Language
- Substitution
- Natural deduction

First-order logic

Propositional logic is a toy system. A very useful one indeed, but still it has not enough expressive power to allow us to describe any useful mathematical theory, e.g., arithmetic or set theory.

Although propositional theories are very well-behaved, as we have seen, we want to use logic as a tool to do real mathematics. And to achieve this objective we need to speak about objects.

The main novelty in first-order logic is that the language is able to identify objects and to write formulæ on them. As already said, we allow quantification to freely range over objects, but not over sets of objects or other collections/structures of objects.

Although outside the scope of the present course higher-order logics, which allow extended quantification, cannot be complete. And first-order logic is in a way at the borderline for completeness, as we will illustrate in due time.

Language

Definition 7.1 (Signature)

A *signature* $\Sigma = \langle S; F; R \rangle$ is composed by

- a finite set S of symbols for *sorts*.
- a set F of symbols for *functions*. Each symbol $f \in F$ is uniquely associated with a *type* $s_1 \times \cdots \times s_n \to s_0$, with $s_i \in S$ for each $0 \leq i \leq n$. When $n = 0$ we say that f is a *constant* of type s_0.
- a set R of symbols for *relations*. Each symbol $r \in R$ is uniquely associated with a *type* $s_1 \times \cdots \times s_n$, with $s_i \in S$ for each $1 \leq i \leq n$. When $n = 0$ we say that r is a *propositional constant*.

The notations $f : s_1 \times \cdots \times s_n \to s_0 \in F$ and $r : s_1 \times \cdots \times s_n \in R$ mean that f is a function symbol whose type is $s_1 \times \cdots \times s_n \to s_0$, and r is a relation symbol whose type is $s_1 \times \cdots \times s_n$, respectively. Also, we require that S, F, and R do not contain the logical connectives and quantifiers.

A signature describes a first-order alphabet: sorts stand for collections of elements, functions are used to denote elements, while relations are used to form basic formulæ.

Language

Example 7.2
The signature

$$\mathcal{N} = \langle \{\mathbb{N}\}; \{0\colon \mathbb{N}, S\colon \mathbb{N} \to \mathbb{N}; +\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}, \cdot\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}\}; \{=\colon \mathbb{N}\times\mathbb{N}\}\rangle$$

specifies the basic language for arithmetic. There is one sort, which stands for the collection of natural numbers in the intended interpretation. There is a constant, 0, denoting the zero natural number, there is a function S, which stands for 'successor', denoting the next natural number, so that $S(5) = 6$ in the intended interpretation, while the functions $+$ and $\cdot$ denote addition and multiplication.
There is only one relation symbol denoting equality.

Of course, the theory of arithmetic should be devised in such a way that as far as possible the formal behaviour, that is, what we can prove, conforms to the intended interpretation.

Language

Example 7.3
The signature $\mathcal{G} = \langle \{G\}; \{1\colon G, \cdot\colon G\times G \to G, \_^{-1}\colon G \to G\}; \{=\colon G\times G\}\rangle$ describes the language of an algebraic group.

Example 7.4
The signature $\mathcal{O} = \langle \{O\}; \emptyset; \{\leq\colon O\times O\}\rangle$ describes the language of an order.

Example 7.5
The signature $\mathcal{L} = \langle \{E, L\}; \{\mathrm{nil}\colon L, \mathrm{cons}\colon E\times L \to L\}; \{=_E\colon E\times E, =_L\colon L\times L\}\rangle$ defines the language of the theory of lists. A computer scientist would say it defines the *data type* of lists.

Terms

The first-order language has two purposes: to provide a syntax to denote elements in the universe, i.e., in the collections denoted by the sorts, and to provide a syntax to denote properties of those elements.

The first issue is addressed by *terms*.

Definition 7.6 (Term)
Let $\Sigma = \langle S; F; R\rangle$ be a signature and let V be an infinite set of symbols, called *variables*, such that $V \cap (S \cup F \cup R) = \emptyset$. Also assume that each variable $x \in V$ has a uniquely associated type $s \in S$ denoted by $x\colon s$. We require that there is an infinite (countable) amount of variables for each type $s \in S$.
A *term* along with the set of its *free variables* is inductively defined as:

- if $x\colon s \in V$ then x is a term of type s, and $\mathrm{FV}(x) = \{x\}$;
- if $f\colon s_1 \times \cdots \times s_n \to s_0 \in F$ and $t_1, \ldots, t_n$ are terms of type $s_1, \ldots, s_n$ respectively, then $f(t_1, \ldots, t_n)$ is a term of type s_0, and $\mathrm{FV}(f(t_1, \ldots, t_n)) = \bigcup_{i=1}^{n} \mathrm{FV}(t_i)$.

We use the notation $t\colon s$ to say that the term t has type s.

Terms

Example 7.7
Using the signature $\mathcal{N}$ of arithmetic, 0, $S(0)$, $S(S(0))$, ... are terms of type $\mathbb{N}$. Also $+(x, 0)$ and $\cdot(x, +(S(0), S(S(0))))$ are terms of type $\mathbb{N}$.
Note how $x + 0$ and $x(1 + 2)$ are **not** terms.

To cope with the need of expressing the standard notation of mathematics within the rigid syntax of terms we will formally introduce definitions later.

Formulæ

As terms are used to denote elements, formulæ are used to denote properties of elements. The syntax is similar to propositional logic with two important differences: we have atomic formulæ instead of propositional variables and we have quantifiers.

Definition 7.8 (Formula)

Fixed a signature $\Sigma = \langle S; F; R \rangle$ and a set of variables as for terms, a *formula* along with the set of its *free variables* is inductively defined as

- $\top$ and $\bot$ are formulæ, and $\mathrm{FV}(\top) = \mathrm{FV}(\bot) = \emptyset$.
- if $r : s_1 \times \cdots \times s_n \in R$ is a relation symbol and $t_1 : s_1, \ldots, t_n : s_n$ are terms then $r(t_1, \ldots, t_n)$ is an *atomic* formula, and $\mathrm{FV}(r(t_1, \ldots, t_n)) = \bigcup_{i=1}^{n} \mathrm{FV}(t_i)$.
- if A and B are formulæ, so are $\neg A$, $A \wedge B$, $A \vee B$, and $A \supset B$, and $\mathrm{FV}(\neg A) = \mathrm{FV}(A)$, $\mathrm{FV}(A \wedge B) = \mathrm{FV}(A \vee B) = \mathrm{FV}(A \supset B) = \mathrm{FV}(A) \cup \mathrm{FV}(B)$.
- if $x : s$ is a variable and A is a formula, so are $\forall x : s. A$ and $\exists x : s. A$, and $\mathrm{FV}(\forall x : s. A) = \mathrm{FV}(\exists x : s. A) = \mathrm{FV}(A) \setminus \{x\}$.

Formulæ

There are two main differences between propositional and first-order formulæ:

- instead of propositional variables we have atomic formulæ, which link the formulæ with terms by means of a relation;
- there are quantified formulæ where the variable is **not** free. We say that quantified variables are *bounded*.

The notion of bounded variable is not new: for example, the expression $\int_a^b f(x)\mathrm{d}x$ does not really depend on the variable x. Indeed, the x is a placeholder to give some name to the argument of the f function. A bounded variable does not denote a value, but rather it acts as a placeholder which allows to write a formula or a term.
Its meaning is controlled by the quantifier and not by the way variables are interpreted, as in the integral x does not denote a real or complex number but rather what is allowed to vary in the function.

Substitution

Variables are subject to a fundamental operation: substitution. Indeed, from a formula A where the variable x appears free, we may obtain another formula, $A[t/x]$, where the term t is substituted for x. For example, in the language of arithmetic x can be substituted in $x + 0 = x$ to obtain $2 + 0 = 2$.

Substitution is fundamental in describing the inference rules governing quantifiers. Bounded variables make substitution not immediately intuitive.

There are many equivalent ways to describe the substitution operation: we will use a method which is not the most immediate but it will become very handy later in the course.

Substitution

Definition 7.9 (Substitution on terms)

Fixed a signature and a term t on it the *substitution* of the variable $x : s$ with the term $r : s$, yielding $t[r/x]$, is defined by induction on the structure of the term t:

- if $t \equiv x$ then $t[r/x] \equiv r$;
- if t is a variable, but $t \not\equiv x$, $t[r/x] \equiv t$;
- if $t \equiv f(t_1, \ldots, t_n)$ then $t[r/x] \equiv f(t_1[r/x], \ldots, t_n[r/x])$.

Note that the substitution operation is defined only when t and x share the same type. Also, observe that substitution is a purely syntactical operation.

Substitution

Definition 7.10 (Substitution on formulæ)

Fixed a signature and a formula A on it the *substitution* of the variable $x : s$ with the term $t : s$, yielding $A[t/x]$, is defined by induction on the structure of the formula A:

- if $A \equiv \top$ or $A \equiv \bot$ then $A[t/x] \equiv A$;
- if $A \equiv r(t_1, \ldots, t_n)$ then $A[t/x] \equiv r(t_1[t/x], \ldots, t_n[t/x])$;
- if $A \equiv \neg B$ then $A[t/x] \equiv \neg B[t/x]$;
- if $A \equiv B \wedge C$, $A \equiv B \vee C$, or $A \equiv B \supset C$ then $A[t/x] \equiv B[t/x] \wedge C[t/x]$, $A[t/x] \equiv B[t/x] \vee C[t/x]$, or $A[t/x] \equiv B[t/x] \supset C[t/x]$, respectively;
- if $A \equiv \forall y : r.B$, or $A \equiv \exists y : r.B$ and $y : r \equiv x : s$ then $A[t/x] \equiv A$;
- if $A \equiv \forall y : r.B$, or $A \equiv \exists y : r.B$, and $y : r \not\equiv x : s$ then $A[t/x] \equiv \forall z : r.(B[z/y])[t/x]$, or $A[t/x] \equiv \exists z : r.(B[z/y])[t/x]$ respectively, where $z : r \notin \mathrm{FV}(B) \cup \mathrm{FV}(t)$.

Substitution

The first clauses in the definition are obvious: we substitute the variable x with the term t where it appears.

The last but one clause means that a bounded variable cannot be substituted: this is simple to understand as it does not make sense to substitute x with 5 in the formula $\exists x : \mathbb{N}.x^2 = x^3$. Indeed, the formula is true because $1^2 = 1 = 1^3$, but evidently it happens just for **some** values of x that the existential quantifier is meant to single out.

The last clause is a bit cryptic. It says that before performing the substitution of x with t on the quantified formula B, we should rename the quantified variable y with a **new** variable z, which does not appear in B and t.

An example may clarify why this must be done: let $A \equiv \exists x : \mathbb{N}.x = 2(y+1)$ and let $t \equiv 2x$. If we do not rename variables $A[t/y]$ would give $\exists x : \mathbb{N}.x = 2(2x+1)$, that is, $\exists x : \mathbb{N}.3x + 2 = 0$. We note the A holds whatever value y may take, while $A[t/y]$ is always false. The problem is that the x in t and the one in A should be kept distinct—and we obtain this by renaming before performing the substitution.

Definitions

The language of first-order logic is cumbersome. Despite we already use a simplified notation, avoiding unneeded parentheses and hiding what can be immediately inferred from the context, the formal nature of the language is far from the reality of the mathematical practice.

On the contrary, the formal nature of the language is what allows it to be analysed: we constantly use induction on the structure of the language (terms, formulæ, proofs) as our main proving instrument.

There is a way in between: we can construct a reasonable language by taking a basic formal language and enriching it with *syntactical sugar*. This does not change the formal nature of the language, but allows to make the language much closer to the standard practice.

This practise takes place by allowing syntactical construction which are not part of the formal language, but still can be directly translated into the formal language. This construction is called *definition* and it has to follow a few precise rules.

Definitions

Definition 7.11 (Function definition)

Fixed a first-order language with equality, let f be a new symbol. Whenever it holds that

$$\forall x_1 : s_1 \ldots \forall x_n : s_n.\exists y : s_0.F \wedge \forall z : s_0.F[z/y] \supset z = y \ ,$$

with $\mathrm{FV}(F) \subseteq \{x_1, \ldots, x_n, y\}$ then $f : s_1 \times \cdots \times s_n \to s_0$ can be used as an additional function symbol since it can be removed from the language by

$$A[f(t_1, \ldots, t_n)/z] = \exists z : s_0.A \wedge (F[z/y])[t_1/x_1, \ldots, t_n/x_n] \wedge \\ \wedge \forall w : s_0.(F[w/y])[t_1/x_1, \ldots, t_n/x_n] \supset z = w$$

for any formula A. As far as a different syntax is non-ambiguous we allow it in place of the standard functional notation.

The idea is that F specifies a functional property and f provides a name for it: we can always remove f from the language using F instead.

Definitions

Definition 7.12 (Relation definition)

Fixed a first-order language, let r be a new symbol. Then $r\colon s_1 \times \cdots \times s_n$ can be used as an additional relation symbol standing for the formula R whenever $\mathrm{FV}(R) = \{x_1, \ldots, x_n\}$ as it can be removed by substituting $R[t_1/x_1, \ldots, t_n/x_n]$ wherever $r(t_1, \ldots, t_n)$ occurs in any formula A. Again, as far as the syntax is non-ambiguous we allow fancy syntactical constructions.

Definitions

Consider any first-order language with equality. Then we add a new form of quantification, which is read as 'uniquely exists': $\exists!x\colon s.\, A$ with $x\colon s$ a variable and A a formula, which stands for $\exists x\colon s.\, A \wedge \forall z\colon s.\, A[z/x] \supset z = x$ with $z\colon s \notin \mathrm{FV}(A)$.

Natural deduction

Fixed any first-order language, the definition of *theory* follows the one already given in the propositional case.

The same holds for the definition of *proof* and the other related terms except that the collection of inference rules contains four new rules to deal with quantifiers. They are illustrated in the next slides.

When the language contains equality we require the presence of other inference rules detailed in the next slides.

The modular composition of inference rules in natural deduction explains why we chose this deduction system instead of one of the many others in literature: all the deduction systems in this course are obtained by adding or deleting a few rules from the propositional or the first-order case.

Natural deduction

Following the previous notation, the rules for universal quantification are

$$\frac{A}{\forall x\colon s.\, A}\,\forall\mathrm{I} \qquad \frac{\forall x\colon s.\, A}{A[t/x]}\,\forall\mathrm{E}$$

provided that

- in $\forall$E, t is a term of type s;
- in $\forall$I, the variable $x\colon s$ does not *occur free in the proof* of the antecedent, which means that for every assumption G, $x\colon s \notin \mathrm{FV}(G)$. This condition is sometimes referred to by saying that $x\colon s$ is an *eigenvariable*.

Note the similarity between the rules for $\forall$ and for $\wedge$.

Natural deduction

Similarly, the rules for existential quantification are

$$\frac{A[t/x]}{\exists x\colon s.A}\,\exists\mathrm{I} \qquad \frac{\exists x\colon s.B \quad \begin{array}{c}[B]\\ \vdots \\ A\end{array}}{A}\,\exists\mathrm{E}$$

provided that

- in $\exists$I, t is a term of type s;
- in $\exists$E, the variable $x\colon s$ does not occur free in the proof of the second antecedent, that is, for every assumption G in the second subproof except for B, $x\colon s \notin \mathrm{FV}(G)$ **and** $x \notin \mathrm{FV}(A)$. Again, $x\colon s$ is said to be an eigenvariable. Note how this inference rule discharges the assumption B.

Observe the similarity between the rules for $\exists$ and for $\vee$.

Natural deduction

Equality is a special relation and this is captured in a series of ad-hoc inference rules. When the language has an equality relation for some sort s it is subject to the following rules:

$$\frac{}{\forall x\colon s.x = x}\,\mathrm{refl} \qquad \frac{}{\forall x\colon s.\forall y\colon s.x = y \supset y = x}\,\mathrm{sym}$$

$$\frac{}{\forall x\colon s.\forall y\colon s.\forall z\colon s.x = y \wedge y = z \supset x = z}\,\mathrm{trans}$$

$$\frac{A[t/x] \quad t = r}{A[r/x]}\,\mathrm{subst}$$

$$\frac{}{\forall x_1\colon s_1 \ldots \forall x_n\colon s_n.\exists! z\colon s_0.z = f(x_1,\ldots,x_n)}\,\mathrm{fun}$$

where t and r are terms of type s and $f\colon s_1 \times \cdots \times s_n \to s_0$ is a function symbol of the language.

References

Usually, first-order logic is presented in a simplified way, by avoiding the multi-sorted language and by using a reduced number of connectives. Although this approach simplifies the initial presentation it makes difficult to move to other logical system, e.g., intuitionistic logic and to deal with real mathematical theories where multiple sorts are often present.

A good text which introduces the first-order language in a formal way is *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), which covers our treatment of definitions, too.

Natural deduction is described in many textbooks. This lesson follows *Anne Sjerp Troelstra* and *Helmut Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43,Cambridge University Press, (1996). The counterexamples have been taken from that text.

Mathematical Logic

Lecture 8

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

First order logic:

- Examples

Examples

Example 8.1

$$\dfrac{\dfrac{\dfrac{[P]^1}{\exists x\colon s.\,P}\exists\mathrm{I} \qquad [\neg\exists x\colon s.\,P]^2}{\bot}\neg\mathrm{E}}{\dfrac{\neg P}{\forall x\colon s.\,\neg P}\forall\mathrm{I}}\neg\mathrm{I}^1}{(\neg\exists x\colon s.\,P)\supset\forall x\colon s.\,\neg P}\supset\mathrm{I}^2$$

By applying the double-negation law ($A=\neg\neg A$) and taking $P\equiv\neg A$ we get that $(\neg\exists x\colon s.\,\neg A)\supset\forall x\colon s.\,A$.

Examples

Example 8.2

$$\dfrac{\dfrac{\dfrac{[\exists x\colon s.\,P]^1 \qquad \dfrac{[P]^2 \qquad \dfrac{[\forall x\colon s.\,\neg P]^3}{\neg P}\forall\mathrm{E}}{\bot}\neg\mathrm{E}}{\bot}\exists\mathrm{E}^2}{\neg\exists x\colon s.\,P}\neg\mathrm{I}^1}{(\forall x\colon s.\,\neg P)\supset\neg\exists x\colon s.\,P}\supset\mathrm{I}^3$$

Putting $P\equiv\neg A$ and applying the double negation law one gets that $\forall x\colon s.\,A=\neg\exists x\colon s.\,\neg A$, i.e., the $\forall$ quantifier is redundant and it can be expressed using negation and the existential quantifier.

Examples

Example 8.3

$$\dfrac{\dfrac{\dfrac{[\exists x\colon s.\,\neg P]^1 \qquad \dfrac{\dfrac{[\forall x\colon s.\,P]^2}{P}\forall\mathrm{E} \qquad [\neg P]^3}{\bot}\neg\mathrm{E}}{\bot}\exists\mathrm{E}^3}{\neg\forall x\colon s.\,P}\neg\mathrm{I}^2}{(\exists x\colon s.\,\neg P)\supset\neg\forall x\colon s.\,P}\supset\mathrm{I}^1$$

Examples

Example 8.4

To show that the restrictions on variables in the introduction rule of the universal quantifier is essential consider the following counterexample.
Let $x : s \in \mathrm{FV}(P)$.

$$\dfrac{\dfrac{\dfrac{[P]^1}{\forall x : s.\, P}\,\forall\mathrm{I}}{P \supset \forall x : s.\, P}\,\supset\mathrm{I}^1}{\forall x : s.\, (P \supset \forall x : s.\, P)}\,\forall\mathrm{I}$$

The instance of the $\forall$I rule on the top is invalid since $x : s$ appear in the assumptions which are undischarged in that moment of the proof.
In arithmetic if P stands for 'x is even' the conclusion allows to prove by $\forall$E that, since $P[0/x]$ is true, every natural number is even!

Examples

Example 8.5

Another counterexample, showing why the restriction on variables is essential in the elimination rule for the existential quantifier is the following.
Again, let $x : s \in \mathrm{FV}(P)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{[\exists x : s.\, P]^1 \qquad \dfrac{[P \supset Q]^2 \quad [P]^3}{Q}\,\supset\mathrm{E}}{Q}\,\exists\mathrm{E}^3}{(\exists x : s.\, P) \supset Q}\,\supset\mathrm{I}^1}{(P \supset Q) \supset ((\exists x : s.\, P) \supset Q)}\,\supset\mathrm{I}^2}{\forall x : s.\, ((P \supset Q) \supset ((\exists x : s.\, P) \supset Q))}\,\forall\mathrm{I}$$

Inside arithmetic, let $Q \equiv \bot$ so the conclusion reduces to $\forall x : s.\, (\neg P \supset \neg \exists x : s.\, P)$. If P stands for 'x is even', since $P[1/x]$ is false the conclusion allows to deduce by $\forall$E that there is no even natural number!

Examples

Example 8.6

The last counterexample shows why the restriction that the quantified variable must not occur in the conclusion of the exist elimination rule.
Let $x \in \mathrm{FV}(A)$:

$$\dfrac{\dfrac{\dfrac{[\exists x : s.\, A]^1 \quad [A]^2}{A}\,\exists\mathrm{E}^2}{\forall x : s.\, A}\,\forall\mathrm{I}}{(\exists x : s.\, A) \supset (\forall x : s.\, A)}\,\supset\mathrm{I}^1$$

Inside arithmetic, let A be the formula stating that its argument is even. Since there is at least an even number, 2 for example, it follows that every number is even.

Examples

Example 8.7

$\vdash \forall x.\, B \supset A = B \supset \forall x.\, A$ with $x \notin \mathrm{FV}(B)$

$$\dfrac{\dfrac{\dfrac{[B]^1 \qquad \dfrac{[\forall x.\, B \supset A]^2}{B \supset A}\,\forall\mathrm{E}}{A}\,\supset\mathrm{E}}{\forall x.\, A}\,\forall\mathrm{I}}{B \supset \forall x.\, A}\,\supset\mathrm{I}^1}{(\forall x.\, B \supset A) \supset (B \supset \forall x.\, A)}\,\supset\mathrm{I}^2$$

$$\dfrac{\dfrac{\dfrac{[B \supset \forall x.\, A]^1 \quad [B]^2}{\forall x.\, A}\,\supset\mathrm{E}}{A}\,\forall\mathrm{E}}{B \supset A}\,\supset\mathrm{I}^2}{(B \supset \forall x.\, A) \supset (\forall x.\, B \supset A)}\,\supset\mathrm{I}^1$$

Examples

Example 8.8
$\vdash \forall x.A \supset B = (\exists x.A) \supset B$ with $x \notin \mathrm{FV}(B)$

$$\dfrac{\dfrac{[\exists x.A]^1 \quad \dfrac{[A]^2 \quad \dfrac{[\forall x.A \supset B]^3}{A \supset B}\forall E}{B}\supset E}{B}\exists E^2}{(\exists x.A) \supset B}\supset I^1}{(\forall x.A \supset B) \supset ((\exists x.A) \supset B)}\supset I^3$$

$$\dfrac{\dfrac{\dfrac{[A]^1}{\exists x.A}\exists I \quad [(\exists x.A) \supset B]^2}{B}\supset E}{A \supset B}\supset I^1}{\forall x.A \supset B}\forall I}{((\exists x.A) \supset B) \supset (\forall x.A \supset B)}\supset I^2$$

Examples

Example 8.9
$\vdash \neg\neg\forall x.A \supset \forall x.\neg\neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\forall x.A]^1}{A}\forall E \quad [\neg A]^2}{\bot}\neg E}{\neg\forall x.A}\neg I^1 \quad [\neg\neg\forall x.A]^3}{\bot}\neg E}{\neg\neg A}\neg I^2}{\forall x.\neg\neg A}\forall I}{\neg\neg(\forall x.A) \supset \forall x.\neg\neg A}\supset I^3$$

Examples

Example 8.10
$\vdash A \wedge (\exists x.B) \supset \exists x.A \wedge B$ with $x \notin \mathrm{FV}(A)$

$$\dfrac{\dfrac{[A \wedge \exists x.B]^1}{\exists x.B}\wedge E_2 \quad \dfrac{\dfrac{[A \wedge \exists x.B]^1}{A}\wedge E_1 \quad [B]^2}{A \wedge B}\wedge I}{\exists x.A \wedge B}\exists I}{A \wedge (\exists x.B) \supset \exists x.A \wedge B}\exists E^2 \supset I^1$$

Examples

Example 8.11
$\vdash \exists x.A \wedge B \supset A \wedge \exists x.B$ with $x \notin \mathrm{FV}(A)$

$$\dfrac{\dfrac{[\exists x.A \wedge B]^1 \quad \dfrac{\dfrac{[A \wedge B]^2}{A}\wedge E_1 \quad \dfrac{[A \wedge B]^2}{B}\wedge E_2}{A \wedge \exists x.B}\exists I}{A \wedge \exists x.B}\wedge I}{(\exists x.A \wedge B) \supset A \wedge \exists x.B}\exists E^2 \supset I^1$$

Examples

Example 8.12
$\vdash A \wedge \forall x.B = \forall x.A \wedge B$ with $x \notin \mathrm{FV}(A)$

$$\dfrac{\dfrac{\dfrac{[A \wedge \forall x.B]^1}{A}\wedge E_1 \quad \dfrac{\dfrac{[A \wedge \forall x.B]^1}{\forall x.B}\wedge E_2}{B}\forall E}{A \wedge B}\wedge I}{\dfrac{A \wedge B}{\forall x.A \wedge B}\forall I}\supset I^1}{A \wedge (\forall x.B) \supset \forall x.A \wedge B}$$

$$\dfrac{\dfrac{\dfrac{[\forall x.A \wedge B]^1}{A \wedge B}\forall E}{A}\wedge E_1 \quad \dfrac{\dfrac{[\forall x.A \wedge B]^1}{A \wedge B}\forall E}{B}\wedge E_2}{A \wedge \forall x.B}\wedge I}{(\forall x.A \wedge B) \supset A \wedge \forall x.B}\supset I^1$$

Examples

Example 8.13
$\vdash (\forall x.A \wedge B) \supset (\forall x.A) \wedge (\forall x.B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\forall x.A \wedge B]^1}{A \wedge B}\forall E}{A}\wedge E_1}{\forall x.A}\forall I \quad \dfrac{\dfrac{\dfrac{[\forall x.A \wedge B]^1}{A \wedge B}\forall E}{B}\wedge E_2}{\forall x.B}\forall I}{(\forall x.A) \wedge (\forall x.B)}\wedge I}{(\forall x.A \wedge B) \supset (\forall x.A) \wedge (\forall x.B)}\supset I^1$$

Examples

Example 8.14
$\vdash (\forall x.A) \wedge (\forall x.B) \supset \forall x.A \wedge B$

$$\dfrac{\dfrac{\dfrac{\dfrac{[(\forall x.A) \wedge (\forall x.B)]^1}{\forall x.A}\wedge E_1}{A}\forall E \quad \dfrac{\dfrac{[(\forall x.A) \wedge (\forall x.B)]^1}{\forall x.B}\wedge E_2}{B}\forall E}{A \wedge B}\wedge I}{\forall x.A \wedge B}\forall I}{(\forall x.A) \wedge (\forall x.B) \supset \forall x.A \wedge B}\supset I^1$$

Examples

Example 8.15
$\vdash (\exists x.A \wedge B) \supset (\exists x.A) \wedge (\exists x.B)$

$$\dfrac{\dfrac{[\exists x.A \wedge B]^1 \quad \dfrac{\dfrac{\dfrac{[A \wedge B]^2}{A}\wedge E_1}{\exists x.A}\exists I \quad \dfrac{\dfrac{[A \wedge B]^2}{B}\wedge E_2}{\exists x.B}\exists I}{(\exists x.A) \wedge (\exists x.B)}\wedge I}{(\exists x.A) \wedge (\exists x.B)}\exists E^2}{(\exists x.A \wedge B) \supset (\exists x.A) \wedge (\exists x.B)}\supset I^1$$

Examples

Example 8.16
$\vdash (\exists x.\forall y.A) \supset \forall y.\exists x.A$

$$\dfrac{\dfrac{\dfrac{[\exists x.\forall y.A]^1 \quad \dfrac{\dfrac{\dfrac{[\forall y.A]^2}{A}\,\forall \mathrm{E}}{\exists x.A}\,\exists \mathrm{I}}{\forall y.\exists x.A}\,\forall \mathrm{I}}{\forall y.\exists x.A}\,\exists \mathrm{E}^2}{(\exists x.\forall y.A) \supset \forall y.\exists x.A}\,\supset \mathrm{I}^1$$

Examples

Example 8.17
Substitute the "forall introduction" rule of natural deduction with the following rule with infinite premises:

$$\dfrac{\left\{\begin{array}{c}\Gamma\\ \vdots\\ A[t/x]\end{array}\right\}_{t \text{ term}}}{\forall x.A}\,\forall \mathrm{I}^{\infty}$$

Show that every formula which can be deduced in this calculus, can be deduced in natural deduction too.

In every proof π in which the $\forall \mathrm{I}^{\infty}$ rule occurs, only a finite number of assumptions in Γ is used. So, in particular only a finite number of variables occur in these assumptions. Pick a new z not among these variables. Then there is a proof $\theta\colon \Gamma \vdash A[z/x]$ among the premises of $\forall \mathrm{I}^{\infty}$. Hence, in the usual natural deduction θ can be used to deduce $\forall z.A[z/x]$ by $\forall \mathrm{I}$, which is equal to $\forall x.A$ by renaming the bound variable.

Examples

Example 8.18
Show that every formula which can be proved in natural deduction, can be inferred in the calculus with the $\forall \mathrm{I}^{\infty}$ rule.

Consider

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ A\end{array}}{\forall x.A}\,\forall \mathrm{I}$$

with x not occurring free in the assumptions. By an easy induction on the structure of the proofs it can be shown that if $\pi\colon \Delta \vdash B$ then $\pi[t/z]\colon \Delta[t/z] \vdash B[t/z]$ is a proof. By substituting t for x in the premise of $\forall \mathrm{I}$ for all the terms t of the same sort of x, we see that all the premises of $\forall \mathrm{I}^{\infty}$ are valid as well as its conclusion.

References

Exercises could be found in any standard textbook, see, e.g., Chapter 1 of *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977).

Some further exercises are available on the course web site

Mathematical Logic

Lecture 9

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

First order logic:

- Informal meaning
- Semantics
- Examples
- Soundness

Informal meaning

Fixed a signature $\langle S; F; R\rangle$ the intended interpretation of a sort $s \in S$ is a specific set; the intended interpretation of a function symbol is a function; and the intended interpretation of a relation symbol is a relation.

The intended meaning of equality, $=: s \times s$, when present in the language, is the identity of the interpretation of its arguments.

Thus the intended meaning of a term is an element, which is identified via the interpretation of functions and the evaluation of variables, in the universe, the collection of all the sets denoted by sorts.

Informal meaning

In turn formulæ stand for a truth value, either true or false, as in the propositional case. And connectives have the intended propositional meaning we already illustrated.

Atomic formulæ, $r(t_1, \ldots, t_n)$, are true when the interpretation of the argument $(t_1, \ldots, t_n)$ is in the relation denoted by r.

A formula is universally valid, that is $\forall x : s.A$ holds, when A is true in whatever way we interpret x as an element of the set denoted by s.

Symmetrically, a formula is existentially valid, that is $\exists x : s.A$ holds, when there is an element e in the set denoted by s such that interpreting x as e makes A true.

Semantics

The standard semantics for first-order logic, due to Alfred Tarski, directly formalises the intended interpretation.

Definition 9.1 (Σ-structure)

Let $\Sigma = \langle S; F, R\rangle$ be a first-order signature.
Then a Σ-*structure* $\mathscr{M} = \langle U; \mathscr{F}; \mathscr{R}\rangle$ is composed by

- a finite collection $U = \{u_s\}_{s\in S}$ of non-empty sets, called the *universe*,
- a collection of functions over the universe $\mathscr{F} = \{g_f : u_{s_1} \times \cdots \times u_{s_n} \to u_{s_0} \mid f : s_1 \times \cdots \times s_n \to s_0 \in F\}$,
- a collection of relations over the universe $\mathscr{R} = \{\rho_r : u_{s_1} \times \cdots \times u_{s_n} \mid r : s_1 \times \cdots \times s_n \in R\}$.

Semantics

To make clear the relation between a signature Σ and a Σ-structure, we use the following notation:

- for each $s \in S$, $[\![s]\!] = u_s$;
- for each $f : s_1 \times \cdots \times s_n \to s_0 \in F$, $[\![f]\!] = g_f$;
- for each $r : s_1 \times \cdots \times s_n \in R$, $[\![r]\!] = \rho_r$.

This is called the *interpretation of the signature* Σ in the Σ-structure.

Semantics

Definition 9.2 (Interpretation of terms)

Let $\Sigma = \langle S; F, R\rangle$ be a signature and let $\mathscr{M}$ be a Σ-structure with the notation as before. Also let $\nu = \{\nu_s\}_{s\in S}$ be a family $\nu_s : \{v \mid v : s \in V\} \to [\![s]\!]$ of functions mapping the variables of type s into the corresponding set $[\![s]\!]$.
Then a term t is interpreted according to the following inductive definition on its structure:

- if $t \in V$ is a variable of type s then $[\![t]\!] = \nu_s(t)$;
- if $t \equiv f(t_1, \ldots, t_n)$ then $[\![t]\!] = [\![f]\!]([\![t_1]\!], \ldots, [\![t_n]\!])$.

Semantics

Definition 9.3 (Interpretation of formulæ)

Let $\Sigma = \langle S; F, R\rangle$ be a signature, let $\mathscr{M}$ be a Σ-structure, and let ν be an *evaluation of variables* with the notation as before.
Then a formula A is interpreted according to the following inductive definition on its structure:

- if $A \equiv \top$, $[\![A]\!] = 1$; if $A \equiv \bot$, $[\![A]\!] = 0$;
- if $A \equiv r(t_1, \ldots, t_n)$, $[\![A]\!] = 1$ if $([\![t_1]\!], \ldots, [\![t_n]\!]) \in [\![r]\!]$ and $[\![A]\!] = 0$ otherwise;
- if $A \equiv \neg B$, $A \equiv B \wedge C$, $A \equiv B \vee C$, $A \equiv B \supset C$ then $[\![A]\!]$ is defined as in the truth-table semantics;
- if $A \equiv \forall x : s.B$ or $A \equiv \exists x : s.B$, let $\xi = \{\xi_s\}_{s\in S}$ be an evaluation of variables such that $\xi_\alpha = \nu_\alpha$ for each $\alpha \neq s$, and $\xi_s(v) = \nu_s(v)$ for each $v \neq x$.
  Then $[\![\forall x : s.B]\!]_\nu = 1$ if, for all the possible ξ, $[\![B]\!]_\xi = 1$, and $[\![\forall x : s.B]\!]_\nu = 0$ otherwise. Also $[\![\exists x : s.B]\!]_\nu = 1$ if there is a ξ such that $[\![B]\!]_\xi = 1$, and $[\![\exists x : s.B]\!]_\nu = 0$ otherwise.

Semantics

We stipulate that when equality is in the language, $[\![t_1 = t_2]\!] = 1$ exactly when $[\![t_1]\!] = [\![t_2]\!]$. If one prefers $[\![=_s]\!]$, the equality on the sort s, represents the *diagonal relation* $\{(x,x) : x \in [\![s]\!]\}$.

It is worth remarking that equality is always typed: $t_1 = t_2$ is a valid formula if and only if t_1 and t_2 are terms of the same sort s, and the relation symbol $=$ should be read as a shorthand for $=_s$, which stands for the diagonal relation on the set denoted by the sort s.

Examples

Example 9.4

Fix the signature of arithmetic and consider the standard model of natural numbers. Then the formula $S0 + S0 = SS0$ is interpreted in $[\![S0 + S0 = SS0]\!] = 1$ since

1. $[\![S0 + S0]\!] = [\![+]\!]([\![S0]\!], [\![S0]\!]) = +([\![S]\!]([\![0]\!]), [\![S]\!]([\![0]\!])) = +(1+0, 1+0) = 1+1 = 2$;
2. $[\![SS0]\!] = [\![S]\!]([\![S0]\!]) = [\![S]\!]([\![S]\!]([\![0]\!])) = 1 + (1+0) = 1+1 = 2$;
3. $[\![S0 + S0 = SS0]\!] = 1$ if and only if $[\![S0 + S0]\!] = [\![SS0]\!]$, that is if and only if $2 = 2$.

Examples

Example 9.5

Fix the signature of arithmetic and consider the standard model of natural numbers. Consider $[\![x = (SS0)y]\!]$. Applying the definition of semantics, $[\![x = (SS0)y]\!] = 1$ if and only if $[\![x]\!] = 2[\![y]\!]$, that is if and only if x is interpreted in a number which is two times the value y is interpreted in.

So, if x is interpreted in 6 and y in 3, the formula is true, while if x is interpreted in 6 but y in 5, the formula is false.

Examples

Example 9.6

Fix the signature of arithmetic and consider the standard model of natural numbers. Consider $[\![\exists x. x = (SS0)x]\!]$. Applying the definition of semantics, $[\![\exists x. x = (SS0)x]\!] = 1$ if and only if there is an assignment ξ of variables, identical to the one fixed in the model except for the value it assigns to x, such that $[\![x = (SS0)x]\!] = 1$. But whenever $\xi(x) = 0$, $[\![x = (SS0)x]\!] = 1$ since both sides evaluate to 0 so the initial formula is true.

Consider $[\![\forall x. x = (SS0)x]\!]$. Applying the definition of semantics, $[\![\forall x. x = (SS0)x]\!] = 1$ if and only if for each assignment ξ of variables, identical to the one fixed in the model except for the value it assigns to x, it holds that $[\![x = (SS0)x]\!] = 1$. But when $\xi(x) = 1$, $[\![x = (SS0)x]\!] = 0$ since the left side evaluates to 1 and the right side to 2.

Examples

Example 9.7
Fix the signature of arithmetic and consider the standard model of natural numbers. Consider $[\![\forall x.\exists y.x = (SS0)y]\!]$. Applying the definition of semantics, the formula holds if for each assignment ξ of variables, identical to the one fixed in the model except for the value of x, it holds that $[\![\exists y.x = (SS0)y]\!] = 1$. In turn this happens when there is an assignment ξ', identical to ξ except for the value of y such that $[\![x = (SS0)y]\!] = 1$.

For each ξ as above fix $\xi'(y) = x/2$, the integer division of x by 2. Whenever x is even it is immediate to check that $[\![x = (SS0)y]\!] = 1$ holds. On the contrary, when x is odd $[\![x = (SS0)y]\!] = 0$.

It is evident that there is no possibility to find an assignment ξ' as above for every possible choice of ξ, so the initial formula is false.

Soundness

Definition 9.8 (Validity)
A formula A is *valid* or *true* in a Σ-structure $\mathscr{M}$ together with an interpretation v of variables when $[\![A]\!] = 1$.
A set of formulæ is *valid* or *true* when each formula in the set is valid.
The pair $(\mathscr{M}, v)$ is a *model* for the theory T when it makes every formula in T true.

Soundness

Proposition 9.9
Let $x : s$ be a variable and $t : s$ a term. Let v be an evaluation and let ξ coincide with v except that $\xi_s(x) = [\![t]\!]_v$. Then, for each term T and for each formula A it holds that $[\![T[t/x]]\!]_v = [\![T]\!]_\xi$ and $[\![A[t/x]]\!]_v = [\![A]\!]_\xi$.

Proof. (i)
By induction on the term T:

- if $T \equiv x$ then $[\![T[t/x]]\!]_v = [\![t]\!]_v = \xi_s(x) = [\![T]\!]_\xi$.
- if $T \not\equiv x$ and $T : s'$ is a variable. Then

$$[\![T[t/x]]\!]_v = [\![T]\!]_v = v_{s'}(T) = \xi_{s'}(T) = [\![T]\!]_\xi \ .$$

- if $T \equiv f(T_1, \ldots, T_n)$ then

$$\begin{aligned}[\![T[t/x]]\!]_v &= [\![f]\!]([\![T_1[t/x]]\!]_v, \ldots, [\![T_n[t/x]]\!]_v) \\ &= [\![f]\!]([\![T_1]\!]_\xi, \ldots, [\![T_n]\!]_\xi) = [\![T]\!]_\xi\end{aligned}$$

where $[\![T_j[t/x]]\!]_v = [\![T_j]\!]_\xi$ by inductive hypothesis. ↪

Soundness

↪ **Proof. (ii)**
By induction on the formula A, where $y_1, \ldots, y_n$ are distinct variables and $z_1, \ldots, z_n$ are distinct and new variables, to show that

$$[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_v = [\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi$$

Observe that the statement of the proposition follows when $n = 0$.

- if $A \equiv \top$ then $[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_v = \top = [\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi$. The case $A \equiv \bot$ is analogous.
- if $A \equiv B \wedge C$ then

$$\begin{aligned}&[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_v \\ &= [\![(B[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_v \wedge [\![(C[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_v \\ &= [\![B[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi \wedge [\![C[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi \\ &= [\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi\end{aligned}$$

The cases $A \equiv \neg B$, $A \equiv B \vee C$ and $A \equiv B \supset C$ are analogous. ↪

Soundness

↪ Proof. (iii)

- if $A \equiv r(T_1, \ldots, T_m)$ then

$$\begin{aligned}
&[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_\nu \\
&= [\![r]\!]\left([\![(T_1[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_\nu, \ldots, [\![(T_m[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_\nu\right) \\
&= [\![r]\!]\left([\![T_1[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi, \ldots, [\![T_m[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi\right) \\
&= [\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi
\end{aligned}$$

where we used the already proven statement on terms. ↪

Soundness

↪ Proof. (iv)

- if $A \equiv \forall x\colon s.B$ then

$$\begin{aligned}
&[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_\nu \\
&= [\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\nu \\
&= [\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi
\end{aligned}$$

since the x variable is not free in the evaluated formula.
The case $A \equiv \exists x\colon s.B$ is analogous. ↪

Soundness

↪ Proof. (v)

- if $A \equiv \forall y_{n+1}\colon s'.B$ with $y_{n+1} \not\equiv x$ then, fixed z_{n+1} new,

$$\begin{aligned}
&[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_\nu \\
&= [\![\forall z_{n+1}\colon s'.(B[z_1/y_1, \ldots, z_{n+1}/y_{n+1}])[t/x]]\!]_\nu \quad .
\end{aligned}$$

Call $\sigma \setminus z$ the set of evaluations which are identical to σ except in z, and call $\sigma[z \mapsto e]$ the evaluation which is identical to σ except that it maps z to e.

Then $[\![(A[z_1/y_1, \ldots, z_n/y_n])[t/x]]\!]_\nu = 1$ if and only if $[\![(B[z_1/y_1, \ldots, z_{n+1}/y_{n+1}])[t/x]]\!]_{\nu'}$ for every $\nu' \in \nu \setminus z_{n+1}$.

Similarly $[\![A[z_1/y_1, \ldots, z_n/y_n]]\!]_\xi = 1$ if and only if $[\![B[z_1/y_1, \ldots, z_{n+1}/y_{n+1}]]\!]_{\xi'} = 1$ for every $\xi' \in \xi \setminus z_{n+1}$. ↪

Soundness

↪ Proof. (vi)

Let $\alpha\colon \nu \setminus z_{n+1} \to \xi \setminus z_{n+1}$ and $\beta\colon \xi \setminus z_{n+1} \to \nu \setminus z_{n+1}$ be

$$\begin{aligned}
\alpha(\nu') &= \xi\left[z_{n+1} \mapsto \nu'_{s'}(z_{n+1})\right] \\
\beta(\xi') &= \nu\left[z_{n+1} \mapsto \xi'_{s'}(z_{n+1})\right] \quad .
\end{aligned}$$

Observe how

$$\begin{aligned}
\beta(\alpha(\nu')) &= \beta\left(\xi\left[z_{n+1} \mapsto \nu'_{s'}(z_{n+1})\right]\right) = \nu\left[z_{n+1} \mapsto \nu'_{s'}(z_{n+1})\right] = \nu' \\
\alpha(\beta(\xi')) &= \alpha\left(\nu\left[z_{n+1} \mapsto \xi'_{s'}(z_{n+1})\right]\right) = \xi\left[z_{n+1} \mapsto \xi'_{s'}(z_{n+1})\right] = \xi'
\end{aligned}$$

that is, α e β are one the inverse of the other.
Moreover $\alpha(\nu') = \nu'[x \mapsto [\![t]\!]_\nu]$ by definition of ξ. ↪

Soundness

↪ Proof. (vii)

Hence, for every $\nu' \in \nu \setminus z_{n+1}$ and $\xi' \in \xi \setminus z_{n+1}$,

$$[\![(B[z_1/y_1,\ldots,z_{n+1}/y_{n+1}])[t/x]]\!]_{\nu'} = [\![B[z_1/y_1,\ldots,z_{n+1}/y_{n+1}]]\!]_{\alpha(\nu')}$$
$$[\![(B[z_1/y_1,\ldots,z_{n+1}/y_{n+1}])[t/x]]\!]_{\beta(\xi')} = [\![B[z_1/y_1,\ldots,z_{n+1}/y_{n+1}]]\!]_{\xi'}$$

by inductive hypothesis.

Therefore,

$$[\![(A[z_1/y_1,\ldots,z_n/y_n])[t/x]]\!]_{\nu} = [\![A[z_1/y_1,\ldots,z_n/y_n]]\!]_{\xi} \quad .$$

The case $A \equiv \exists y_{n+1}\colon s'.B$ with $y_{n+1} \not\equiv x$ is completely analogous. □

Soundness

Theorem 9.10 (Soundness)

In any model $(\mathscr{M},\nu)$ of the theory T, which makes true the assumptions in the set Δ, if $\pi\colon \Delta \vdash_T A$ then A is valid.

Soundness

Proof. (i)

First, we observe that by Definition 9.3 the connectives act in the Boolean algebra on $\{0,1\}$ with $0<1$, so the $\wedge$, $\vee$, $\neg$ operations are defined as in the truth-table semantics.
The proof is by induction on the structure of the proof π: we prove that the interpretation of the conclusion A is 1 when the interpretation of each G in the finite set of assumption Γ is 1:

- if π is a proof by assumption then $A \in \Gamma$ and by hypothesis $[\![A]\!] = 1$.
- if π is a proof by axiom then $A \in T$ and by hypothesis $[\![A]\!] = 1$.
- if π is an instance of the Law of Excluded Middle then $A \equiv B \vee \neg B$ and $[\![A]\!] = [\![B \vee \neg B]\!] = [\![B]\!] \vee \neg [\![B]\!] = 1$ by definition of complement.
- if π is an instance of $\top$-introduction then $A \equiv \top$ so $[\![A]\!] = 1$.
- if π is an instance of refl then $A \equiv \forall x\colon s.x = x$, so $[\![A]\!] = 1$ when $[\![x = x]\!] = 1$ for each possible evaluation of the variable x in $[\![s]\!]$. So if x gets mapped to $e \in [\![s]\!]$, $(e,e) \in \{(z,z)\colon z \in [\![s]\!]\}$, so $[\![x = x]\!] = 1$ for any e. ↪

Soundness

↪ Proof. (ii)

- if π is an instance of sym then $A \equiv \forall x\colon s.\forall y\colon s.x = y \supset y = x$, so $[\![A]\!] = 1$ when $[\![x = y \supset y = x]\!] = 1$ for each possible evaluation of the variables x and y in $[\![s]\!]$. So if x gets mapped to $e_x \in [\![s]\!]$ and y to $e_y \in [\![s]\!]$, if $(e_x,e_y) \in \{(z,z)\colon z \in [\![s]\!]\}$ then $e_x = e_y$, thus $(e_y,e_x) \in \{(z,z)\colon z \in [\![s]\!]\}$, that is $[\![x = y \supset y = x]\!] = 1$.
- if π is an instance of trans then $A \equiv \forall x\colon s.\forall y\colon s.\forall z\colon s.x = y \wedge y = z \supset x = z$, so $[\![A]\!] = 1$ when $[\![x = y \wedge y = z \supset x = z]\!] = 1$ for each possible evaluation of the variables x, y, and z in $[\![s]\!]$. So if x gets mapped to $e_x \in [\![s]\!]$, y to $e_y \in [\![s]\!]$, and z in $e_z \in [\![s]\!]$, if $(e_x,e_y) \in \{(z,z)\colon z \in [\![s]\!]\}$ and $(e_y,e_z) \in \{(z,z)\colon z \in [\![s]\!]\}$ then $e_x = e_y = e_z$, and thus $(e_x,e_z) \in \{(z,z)\colon z \in [\![s]\!]\}$, that is $[\![x = y \wedge y = z \supset x = z]\!] = 1$. ↪

Soundness

↪ Proof. (iii)

- if π is an instance of fun then $A \equiv \forall x_1 : s_1 \ldots \forall x_n : s_n . \exists! z : s_0 . z = f(x_1, \ldots, x_n)$, so $[\![A]\!] = 1$ exactly when z can be uniquely mapped into a value e_z in $[\![s_0]\!]$ so that $(e_z, [\![f]\!](e_{x_1}, \ldots, e_{x_n})) \in \{(z, z) : z \in [\![s]\!]\}$, which is evidently true for $e_z = [\![f]\!](e_{x_1}, \ldots, e_{x_n})$.
- if π is an instance of subst then by induction hypothesis $[\![A[t/x]]\!] = 1$ and $[\![t = r]\!] = 1$, that is $[\![t]\!] = [\![r]\!]$. The conclusion follows by an easy induction on the structure of the formula A.
- if π is an instance of $\bot$-elimination then by induction hypothesis $0 = [\![\bot]\!] = 1$. Thus $[\![A]\!] = 1$ since interpretation is a total function. ↪

Soundness

↪ Proof. (iv)

- if π is an instance of $\wedge$-introduction then $A \equiv B \wedge C$, and by induction hypothesis twice $[\![B]\!] = 1$ and $[\![C]\!] = 1$. Thus $1 = [\![B]\!] \wedge [\![C]\!] = [\![A]\!]$.
- if π is an instance of $\wedge_1$-elimination then by induction hypothesis for some formula B, $[\![A \wedge B]\!] = [\![A]\!] \wedge [\![B]\!] = 1$. Thus by definition of $\wedge$, $[\![A]\!] = 1$.
- if π is an instance of $\wedge_2$-elimination then by induction hypothesis for some formula B, $[\![B \wedge A]\!] = [\![B]\!] \wedge [\![A]\!] = 1$. Thus by definition of $\wedge$, $[\![A]\!] = 1$.
- if π is an instance of $\vee_1$-introduction then $A \equiv B \vee C$ and by induction hypothesis $[\![B]\!] = 1$. So by definition of $\vee$, $1 = [\![B]\!] \vee [\![C]\!] = [\![A]\!]$.
- if π is an instance of $\vee_2$-introduction then $A \equiv B \vee C$ and by induction hypothesis $[\![C]\!] = 1$. So by definition of $\vee$, $1 = [\![B]\!] \vee [\![C]\!] = [\![A]\!]$.
- if π is an instance of $\vee$-elimination then by induction hypothesis for some formulæ B and C, $[\![B \vee C]\!] = [\![B]\!] \vee [\![C]\!] = 1$, if $[\![B]\!] = 1$ then $[\![A]\!] = 1$, and if $[\![C]\!] = 1$ then $[\![A]\!] = 1$. By definition of $\vee$ either $[\![B]\!] = 1$, thus $[\![A]\!] = 1$, or $[\![C]\!] = 1$, thus $[\![A]\!] = 1$. ↪

Soundness

↪ Proof. (v)

- if π is an instance of $\supset$-introduction then $A \equiv B \supset C$ for some formulæ B and C. By induction hypothesis if $[\![B]\!] = 1$ then $[\![C]\!] = 1$. So by definition of $\supset$, $[\![A]\!] = 1$.
- if π is an instance of $\supset$-elimination then for some formula B by induction hypothesis twice $[\![B \supset A]\!] = 1$ and $[\![B]\!] = 1$. By definition of $\supset$, $[\![A]\!] = 1$.
- if π is an instance of $\neg$-introduction then $A \equiv \neg B$ for some formula B. So by induction hypothesis if $[\![B]\!] = 1$ then $0 = [\![\bot]\!] = 1$. Thus, $[\![\neg B]\!] = 1$ as either $[\![B]\!] = 0$ or $0 = 1$.
- if π is an instance of $\neg$-elimination then $A \equiv \bot$ and by induction hypothesis twice $[\![\neg B]\!] = 1$ and $[\![B]\!] = 1$. So by definition of complement $0 = 1$. Thus $0 = [\![A]\!] = 1$. ↪

Soundness

↪ Proof. (vi)

- if π is an instance of $\forall$-introduction then $A \equiv \forall x : s . B$, and by induction hypothesis $[\![B]\!] = 1$ for every evaluation of variables which makes the assumptions true. But since $x : s$ does not appear free in any assumption, $[\![B]\!] = 1$ for any way we may evaluate x in $[\![s]\!]$, that is $[\![A]\!] = 1$.
- if π is an instance of $\forall$-elimination then $A \equiv B[t/x]$, and by induction hypothesis $[\![\forall x : s . B]\!] = 1$. So in particular when x evaluates to $[\![t]\!]$, $[\![A]\!] = [\![B[t/x]]\!] = 1$. ↪

Soundness

↪ Proof. (vii)

- if π is an instance of $\exists$-introduction then $A \equiv \exists x\colon s.B$, and by induction hypothesis $[\![B[t/x]]\!] = 1$. So the evaluation of variables ξ_s which is the same as v_s except for $\xi_s(x) = [\![t]\!]$ makes A valid.
- if π is an instance of $\exists$-elimination then by induction hypothesis $[\![\exists x\colon s.B]\!] = 1$ and if $[\![B]\!] = 1$ then A is valid. But $[\![\exists x\colon s.B]\!] = 1$ means that there is way to evaluate x in $[\![s]\!]$ which makes B valid. Applying this evaluation to the second induction hypothesis, we get that A is valid. □

Mathematical Logic

Lecture 10

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

References

The interpretation of formulæ as illustrated in this lesson has been formalised first by Alfred Tarski. This is a classical definition and it can be found in most textbooks.

The notion of model, that is a Σ-structure which satisfies all the axioms in a theory, is analysed in depth in the branch of Logic called *model theory*. A standard reference is *Chen Chung Chang* and *Howard Jerome Keisler*, Model Theory, Studies in Logic and the Foundations of Mathematics, 3rd edition, Elsevier, (1990). Nevertheless this text is quite dated and an introduction to the basics of contemporary model theory can be found in *Wilfrid Hodges*, A Shorter Model Theory, Cambridge University Press, (1997).

The soundness theorem is a classical result and its proof can be found in most textbooks. Our treatment follows the already cited *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977). It is worth comparing the proof in this lesson with the propositional proof using the truth-tables semantics.

Syllabus

First order logic:

- Completeness

Strategy

The completeness theorem is difficult, both technically and conceptually.

The strategy to prove it is indirect:

- Suppose A is true in any model satisfying Γ. Then $\Gamma \cup \{\neg A\}$ has no model.
- Every set of formulæ Δ which is consistent, i.e., non allowing to derive a contradiction, has a model. This is proved by constructing a sufficiently big set Θ containing Δ which has enough information to synthesise a model for itself.
- So, $\Gamma \cup \{\neg A\}$ must be non consistent. Which means that $\Gamma \vdash A$.

We need to prove each step. And we will start from the end.

Consistency

Definition 10.1 (Consistent set)

Fixed a first-order signature, a set of formulæ Γ on it is *consistent* when it does not happen that $\Gamma \vdash A$ and $\Gamma \vdash \neg A$ for any formula A in the language.

Definition 10.2 (Maximal consistent set)

Fixed a first-order signature, a set of formulæ Γ on it is *maximal consistent* when it is consistent and for any other set Δ on the same language such that $\Gamma \subset \Delta$, Δ is not consistent.

It should be stressed that being maximal consistent is a property which is **not** invariant with respect to the language.

Consistency

Proposition 10.3

For any set of formulæ Γ and any formula A,

- $\Gamma \cup \{\neg A\}$ *is not consistent if and only if* $\Gamma \vdash A$*;*
- $\Gamma \cup \{A\}$ *is not consistent if and only if* $\Gamma \vdash \neg A$*.*

Proof.

If $\Gamma \cup \{\neg A\}$ is non consistent then $\Gamma \cup \{\neg A\} \vdash B$ and $\Gamma \cup \{\neg A\} \vdash \neg B$ for some B. So, by implication introduction $\Gamma \vdash \neg A \supset B$ and $\Gamma \vdash \neg A \supset \neg B$. Since $\vdash (\neg A \supset B) \wedge (\neg A \supset \neg B) \supset A$ can be easily proved using the double negation law, see Example 3.17, it follows that $\Gamma \vdash A$.
Conversely $\Gamma \cup \{\neg A\} \vdash A$ by hypothesis, and $\Gamma \cup \{\neg A\} \vdash \neg A$ by the assumption rule, so $\Gamma \cup \{\neg A\}$ is not consistent.
By the double negation law, $\Gamma \cup \{A\}$ is non consistent if and only if $\Gamma \cup \{\neg\neg A\}$ is non consistent, thus the second part follows from the first one. □

Consistency

The completeness theorem says that: if a formula A is true in every model of the theory Γ then there is a proof of A from Γ.

Now, by Proposition 10.3 it suffices to prove that: if a formula A is true in every model of the theory Γ then $\Gamma \cup \{\neg A\}$ is not consistent.

We note that any super set of a set of non consistent formulæ is non consistent, too. The idea we want to pursue is to construct a sufficiently rich super set of any consistent set that allows to build a model.

Consistency

Proposition 10.4

A set Γ is maximal consistent if and only if it is consistent and for every formula A either $A \in \Gamma$ or $\neg A \in \Gamma$.

Proof.

Suppose Γ is maximal consistent. Then it is consistent by definition. Also, suppose there is A such that $A \notin \Gamma$ and $\neg A \notin \Gamma$ then $\Gamma \cup \{A\}$ and $\Gamma \cup \{\neg A\}$ must be both non consistent by definition. Thus, by Proposition 10.3 $\Gamma \vdash \neg A$ and $\Gamma \vdash A$, making Γ non consistent, which is a contradiction.
Conversely, suppose $\Gamma \subset \Delta$. Then there is $A \in \Delta$ such that $A \notin \Gamma$. So by hypothesis $\neg A \in \Gamma \subset \Delta$. Thus, $\Delta \vdash A$ and $\Delta \vdash \neg A$ by assumption. □

Corollary 10.5

If Γ is maximal consistent and $\Gamma \vdash A$ then $A \in \Gamma$.

Proof.

Otherwise $\neg A \in \Gamma$ thus $\Gamma \vdash \neg A$ making Γ non consistent. □

Closure of maximal consistent sets

Proposition 10.6

Let Γ be a maximal consistent set. Then the following facts hold:

1. $\top \in \Gamma$; $\bot \notin \Gamma$;
2. *if* $A \equiv r(t_1, \ldots, t_n)$ *then either* $A \in \Gamma$ *or* $\neg A \in \Gamma$;
3. *if* $\neg\neg A \in \Gamma$ *then* $A \in \Gamma$;
4. *if* $A \wedge B \in \Gamma$ *then* $A \in \Gamma$ *and* $B \in \Gamma$; *if* $\neg(A \wedge B) \in \Gamma$ *then* $\neg A \in \Gamma$ *or* $\neg B \in \Gamma$;
5. *if* $A \vee B \in \Gamma$ *then* $A \in \Gamma$ *or* $B \in \Gamma$; *if* $\neg(A \vee B) \in \Gamma$ *then* $\neg A \in \Gamma$ *and* $\neg B \in \Gamma$;
6. *if* $A \supset B \in \Gamma$ *then* $\neg A \in \Gamma$ *or* $B \in \Gamma$; *if* $\neg(A \supset B) \in \Gamma$ *then* $A \in \Gamma$ *and* $\neg B \in \Gamma$;
7. *if* $\forall x\colon s.A \in \Gamma$ *then* $A[t/x] \in \Gamma$ *for each term* $t\colon s$;
8. *if* $\neg(\exists x\colon s.A) \in \Gamma$ *then* $\neg A[t/x] \in \Gamma$ *for each term* $t\colon s$.

Proof. (i)

Since $\Gamma \vdash \top$ by truth introduction, $\top \in \Gamma$. Hence $\bot \notin \Gamma$ since $\neg\top$ is equivalent to $\bot$. The condition on atomic formulæ follows from Proposition 10.4. ↪

Closure of maximal consistent sets

↪ **Proof. (ii)**

If $A \wedge B \in \Gamma$ then $\Gamma \vdash A$ and $\Gamma \vdash B$ by conjunction elimination. So, by Corollary 10.5 $A \in \Gamma$ and $B \in \Gamma$. Moreover, by De Morgan's Laws $\neg(A \vee B)$ is equivalent to $\neg A \wedge \neg B$, so the required result follows. Also since $\neg(A \supset B)$ is equivalent to $A \wedge \neg B$, the required result follows.
If $A \vee B \in \Gamma$ and $A \notin \Gamma$ it must be $\neg A \in \Gamma$. So it is immediate to see that $\Gamma \vdash B$, i.e., $B \in \Gamma$. Moreover by De Morgan's Laws $\neg(A \wedge B)$ is equivalent to $\neg A \vee \neg B$, so the required result follows.
If $A \supset B \in \Gamma$ and $\neg A \notin \Gamma$ it must be $A \in \Gamma$. So it is immediate to see that $\Gamma \vdash B$, i.e., $B \in \Gamma$. Also by the double negation law $\Gamma \vdash \neg\neg A \supset A$, so if $\neg\neg A \in \Gamma$, $A \in \Gamma$, too.
If $\forall x\colon s.A \in \Gamma$, by the forall elimination rule $\Gamma \vdash A[t/x]$ for any term $t\colon s$. Thus $A[t/x] \in \Gamma$. Also, since $\neg\exists x\colon s.A$ is equivalent to $\forall x\colon s.\neg A$, the required result follows. □

Closure of maximal consistent sets

Proposition 10.7

Let Γ be a maximal consistent set in a language with equality. Then the following facts hold:

1. $t = t \in \Gamma$ *for all terms* t;
2. *if* $t = r \in \Gamma$ *then also* $r = t \in \Gamma$;
3. *if* $t = r \in \Gamma$ *and* $r = u \in \Gamma$ *then also* $t = u \in \Gamma$;
4. *if* $t_i = r_i \in \Gamma$ *for each* $1 \leq i \leq n$ *then* $f(t_1, \ldots, t_n) = f(r_1, \ldots, r_n) \in \Gamma$ *for every* $f\colon s_1 \times \cdots \times s_n \to s_0$ *in the language;*
5. *if* $t_i = r_i \in \Gamma$ *for each* $1 \leq i \leq n$ *then* $p(t_1, \ldots, t_n) \supset p(r_1, \ldots, r_n) \in \Gamma$ *for every* $p\colon s_1 \times \cdots \times s_n$ *in the language.*

Proof.

Since all these equalities can be deduced from Γ applying the inference rules in an elementary way, by Corollary 10.5 the results follow. □

Closure of maximal consistent sets

Two evident conditions are lacking from Proposition 10.6:

- if $\exists x\colon s.A \in \Gamma$ then $A[t/x] \in \Gamma$ for some term $t\colon s$;
- if $\neg(\forall x\colon s.A) \in \Gamma$ then $\neg A[t/x] \in \Gamma$ for some term $t\colon s$.

Indeed, the second condition is equivalent to the first one since $\neg(\forall x\colon s.A)$ is equivalent to $\exists x\colon s.\neg A$.

The first condition is lacking simply because it does not hold for any maximal consistent set. Take the language with just equality and let $U = \{u, v\}$. Consider the variable evaluation σ which maps every variable x in u. Call Ψ the collection of all true formulæ on the model U under the evaluation σ.

Evidently, Ψ is consistent since it has a model: if a theory T has a model and it is inconsistent, it would follow by the Soundness Theorem 9.10 that it would make true both a formula and its negation. Moreover, for any formula A either it is true or false in that particular model, so either $A \in \Psi$ or $\neg A \in \Psi$.

But $\exists x.\neg x = y$, with x and y distinct variables, is true while $(\neg x = y)[t/x]$ is false for any term t because the only terms are variables and all of them are interpreted into the same element u.

Henkin sets

Definition 10.8 (Henkin set)

A set of formulæ Γ in a language is a *Henkin set* when Γ is maximal consistent in that language and

- if $\exists x\colon s.A \in \Gamma$ then $A[t/x] \in \Gamma$ for some term $t\colon s$;
- if $\neg(\forall x\colon s.A) \in \Gamma$ then $\neg A[t/x] \in \Gamma$ for some term $t\colon s$.

Thus, Henkin sets form a proper subclass of maximal consistent sets, and they are the *right* objects to look at as they contain enough information to construct a model for themselves.

Canonical model

Lemma 10.9

If Γ is a Henkin set then it has a model $(\mathscr{M}, \sigma)$.

Proof. (i)

Let T be the set of terms in the language. Define $t \sim r$ when $t\colon s, r\colon s \in T$ and $t = r \in \Gamma$ (if the language does not contain an equality on the sort s, let $t \sim r$ when $t \equiv r$). By the properties of a Henkin set, see Proposition 10.7, $\sim$ is an equivalence relation. So it induces a partition on T. Thus we define $U = \{\{[t]_\sim\colon t\colon s \in T\}\}_{s \in S}$, grouping partitions by sort.

For each function symbol $f\colon s_1 \times \cdots \times s_n \to s_0$ in Σ,

$$[\![f]\!]([t_1]_\sim, \ldots, [t_n]_\sim) = [f(t_1, \ldots, t_n)]_\sim \quad .$$

Note how this definition is legitimate, since the class $[f(t_1, \ldots, t_n)]_\sim$ does not depend on the choice of the representatives $[t_1]_\sim$, $\ldots$, $[t_n]_\sim$ by a direct application of Proposition 10.7. ↪

Canonical model

↪ Proof. (ii)

For each relation symbol $p\colon s_1 \times \cdots s_n$ in Σ,

$$[\![p]\!] = \{([t_1]_\sim, \ldots, [t_n]_\sim) : p(t_1, \ldots, t_n) \in \Gamma\} \quad .$$

Again, this definition is legitimate since it does not depend on the choice of the representatives $[t_1]_\sim$, $\ldots$, $[t_n]_\sim$ by Proposition 10.7.

So let $\mathscr{M}$ be the Σ-structure having U as its universe, and interpreting function symbols and relation symbols as above.

Define σ, the evaluation of variables as $\sigma(x\colon s) = [x]_\sim$.

By induction on the structure of terms we show that $[\![t]\!] = [t]_\sim$:

- if $t \equiv x\colon s$ is a variable, $[\![t]\!] = \sigma(x\colon s) = [t]_\sim$;
- if $t \equiv f(t_1, \ldots, t_n)$, $[\![t]\!] = [\![f]\!]([\![t_1]\!], \ldots, [\![t_n]\!])$, and by induction hypothesis $[\![t]\!] = [\![f]\!]([t_1]_\sim, \ldots, [t_n]_\sim) = [f(t_1, \ldots, t_n)]_\sim = [t]_\sim$. ↪

Canonical model

↪ Proof. (iii)

By induction on the structure of formulæ we show that, when $A \in \Gamma$, $[\![A]\!] = 1$, and when $\neg A \in \Gamma$, $[\![A]\!] = 0$.

- if $A \equiv \top$ then $A \in \Gamma$ and by definition $[\![A]\!] = 1$.
- if $A \equiv \bot$, then $\neg A \in \Gamma$ and by definition $[\![A]\!] = 0$.
- if $A \equiv p(t_1, \ldots, t_n)$, $[\![A]\!] = 1$ if and only if $([\![t_1]\!], \ldots [\![t_n]\!]) \in [\![p]\!]$, that is $([t_1]_\sim, \ldots, [t_n]_\sim) \in [\![p]\!]$, and by definition of the model this happens exactly when $p(t_1, \ldots, t_n) \in \Gamma$, i.e., when $A \in \Gamma$. When $\neg A \in \Gamma$, being Γ maximal consistent $A \notin \Gamma$, so $[\![A]\!] = 0$.
- if $A \equiv t = r$, $[\![A]\!] = 1$ exactly when $[\![t]\!] = [\![r]\!]$, which is equivalent to $[t]_\sim = [r]_\sim$, and by definition of the model $t = r \in \Gamma$. Again, if $\neg t = r \in \Gamma$, being Γ maximal consistent $t = r \notin \Gamma$, and $[\![A]\!] = 0$.
- if $A \equiv \neg B$, $[\![A]\!] = 1$ exactly when $[\![B]\!] = 0$, and by induction hypothesis this happens exactly when $B \notin \Gamma$. Conversely, if $A \notin \Gamma$ then $B \in \Gamma$ being Γ maximal consistent, so by induction hypothesis $[\![B]\!] = 1$, i.e., $[\![A]\!] = 0$. ↪

Canonical model

↪ Proof. (iv)

- if $A \equiv B \wedge C$, $[\![A]\!] = 1$ if and only if $[\![B]\!] = 1$ and $[\![C]\!] = 1$, but by induction hypothesis this happens exactly when $B \in \Gamma$ and $C \in \Gamma$. So, when $A \in \Gamma$, by Proposition 10.6, $B \in \Gamma$ and $C \in \Gamma$, thus $[\![A]\!] = 1$. On the contrary, when $\neg A \in \Gamma$, by Proposition 10.6 $\neg B \in \Gamma$ or $\neg C \in \Gamma$, and being Γ maximal consistent either $B \notin \Gamma$ or $C \notin \Gamma$. In both cases, $[\![A]\!] \neq 1$, so $[\![A]\!] = 0$.
- if $A \equiv B \vee C$, $[\![A]\!] = 1$ if and only if $[\![B]\!] = 1$ or $[\![C]\!] = 1$, but by induction hypothesis this happens exactly when $B \in \Gamma$ or $C \in \Gamma$. So, when $A \in \Gamma$, by Proposition 10.6 $B \in \Gamma$ or $C \in \Gamma$, thus $[\![A]\!] = 1$. On the contrary, when $\neg A \in \Gamma$, by Proposition 10.6 $\neg B \in \Gamma$ and $\neg C \in \Gamma$, and being Γ maximal consistent $B \notin \Gamma$ and $C \notin \Gamma$. Hence $[\![A]\!] \neq 1$, so $[\![A]\!] = 0$.
- if $A \equiv B \supset C$, $[\![A]\!] = 1$ if and only if $[\![B]\!] = 0$ or $[\![C]\!] = 1$, but by induction hypothesis this happens exactly when $\neg B \in \Gamma$ or $C \in \Gamma$. So, when $A \in \Gamma$, by Proposition 10.6 $\neg B \in \Gamma$ or $C \in \Gamma$, thus $[\![A]\!] = 1$. On the contrary, when $\neg A \in \Gamma$, by Proposition 10.6 $B \in \Gamma$ and $\neg C \in \Gamma$, and being Γ maximal consistent $B \in \Gamma$ and $C \notin \Gamma$. Hence $[\![A]\!] \neq 1$, so $[\![A]\!] = 0$. ↪

Canonical model

↪ Proof. (v)

- if $A \equiv \forall x\colon s.\, B$, $[\![A]\!] = 1$ exactly when in whatever way $x\colon s$ is interpreted in U, $[\![B]\!] = 1$. Since U is composed by equivalence classes of terms, $x\colon s$ is interpreted in $[t]_\sim$ for any term $t\colon s$. This means that $[\![B[t/x]]\!] = 1$ in the σ evaluation of variables. By Proposition 10.6 when $A \in \Gamma$, $B[t/x] \in \Gamma$ for every term $t\colon s$, so by induction hypothesis $[\![B[t/x]]\!] = 1$ for any term $t\colon s$, thus $[\![A]\!] = 1$. Furthermore, when $\neg A \in \Gamma$, being Γ a Henkin set there is a term $t\colon s$ such that $\neg B[t/x] \in \Gamma$, so by induction hypothesis $[\![B[t/x]]\!] = 0$, thus $[\![A]\!] = 0$.
- if $A \equiv \exists x\colon s.\, B$, $[\![A]\!] = 1$ exactly when, there is a way to interpret $x\colon s$ in U such that $[\![B]\!] = 1$. By definition of U, $x\colon s$ is interpreted in $[t]_\sim$ for some term $t\colon s$. This means that $[\![B[t/x]]\!] = 1$ in the σ evaluation of variables. Being Γ a Henkin set when $A \in \Gamma$, $B[t/x] \in \Gamma$ for some term $t\colon s$, so by induction hypothesis $[\![B[t/x]]\!] = 1$, thus $[\![A]\!] = 1$. Also, when $\neg A \in \Gamma$, by Proposition 10.6 for all the terms $t\colon s$ it holds that $\neg B[t/x] \in \Gamma$, so by induction hypothesis $[\![B[t/x]]\!] = 0$, thus $[\![A]\!] = 0$. ↪

Canonical model

↪ Proof. (vi)

Summarising, we have constructed a Σ-structure $\mathscr{M}$ and an evaluation of variables σ such that each formula $A \in \Gamma$ is true in $\mathscr{M}$ under σ. □

Corollary 10.10

The $\mathscr{M}$ model has a universe which does not exceed the size of the collection of all terms.

References

The first completeness proof for first-order logic has been given by Kurt Gödel. The proof presented in this lesson follows the techniques introduced by Leon Henkin.

Our treatment follows *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977).

Mathematical Logic

Lecture 11

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

First order logic:

- Completeness

Existence of Henkin sets

Proposition 11.1

Let Γ be a consistent set of formulæ on the signature Σ. Then there is a set of formulæ Δ on a signature Σ' extending Σ with constants such that Δ is a Henkin set and $\Gamma \subseteq \Delta$.

Proof. (i)

Warning: we anticipate some set theory here!

Let λ be the cardinality of the collection of terms on Σ. Let

$$C = \bigcup_{s \in S} \{c_i^s : s \mid i < \lambda\}$$

be a collection of symbols for constants such that no $c_i^s : s$ appears in Σ. Let Σ' be Σ extended with the set of constants in C.

The collection of all formulæ over Σ' is a set with cardinality λ as it is easy to verify by cardinal arithmetic. So, it can be well-ordered in the sequence $\mathbb{S} = \{S_i : i < \lambda\}$ by means of an equivalent of the Axiom of Choice. ↪

Existence of Henkin sets

Although the concepts will be made precise later in the course, some intuitions are useful to understand what we are doing:

- a *cardinal* is a measure of the size of a set;
- every cardinal is an *ordinal*;
- an ordinal is an extension and abstraction over the structure of naturals;
- one can add and multiply ordinals and cardinals;
- every ordinal is well-ordered by its own definition, so also a cardinal is so;
- the Axiom of Choice tells that every set has a cardinality, thus it can be enumerated by the elements of the (unique) associated cardinal.

Existence of Henkin sets

An ordinal α can be thought as an initial segment of the order $\mathcal{O}$ defined as

- $0 \in \mathcal{O}$;
- if $n \in \mathcal{O}$ then $n+1 \in \mathcal{O}$, that is $\mathcal{O}$ is closed under the successor operator;
- if E is an initial segment of $\mathcal{O}$, then $E \in \mathcal{O}$ (this is not precise).

The result is that every natural number is an ordinal; but also ω, the collection of all natural numbers is an ordinal; thus $\omega+1$, $\omega+2$, $\omega+3$, ... are ordinals; then $\omega+\omega$ is an ordinal; and so on.

Thinking to this structure as inductively generated by three steps (zero, successor, limit), we get an induction principle, called *transfinite induction*.

Existence of Henkin sets

↪ Proof. (ii)

By transfinite induction on λ we define for every $i \leq \lambda$ a set Γ_i of formulæ such that

1. $\Gamma_j \subseteq \Gamma_i$ for every $j < i$;
2. Γ_i is consistent;
3. no more than $\max(i, \omega)$ constants in C occur in Γ_i.

We pose $\Gamma_0 = \Gamma$.
Condition (1) holds vacuously;
(2) holds by hypothesis;
(3) holds since no constant in C appears in Γ by definition. ↪

Existence of Henkin sets

↪ Proof. (iii)

If $i \leq \lambda$ is a limit ordinal, we put $\Gamma_i = \bigcup_{j<i} \Gamma_j$.
By definition condition (1) holds.
If Γ_i is not consistent, $\Gamma_i \vdash A$ and $\Gamma_i \vdash \neg A$ then each proof uses only a finite subset of assumptions Γ_i^A and $\Gamma_i^{\neg A}$. But every finite subset of Γ_i is contained in some Γ_j, with $j < i$, so there is $m < i$ such that $\Gamma_i^A \subseteq \Gamma_m$ and $\Gamma_i^{\neg A} \subseteq \Gamma_m$, thus $\Gamma_m \vdash A$ and $\Gamma_m \vdash \neg A$, contradicting the inductive hypothesis that Γ_m is consistent. So Γ_i must be consistent, proving (2).
Finally, since (3) holds for any $j < i$, because of (1) it must hold also for i by simple cardinal arithmetic, proving (3). ↪

Existence of Henkin sets

↪ Proof. (iv)

If $i < \lambda$ is a successor ordinal say $i = k+1$, we distinguish three cases:

- If $\Gamma_k \cup \{S_k\}$ is non consistent then $\Gamma_i = \Gamma_k$, and the three conditions clearly hold by inductive hypothesis.
- If $\Gamma_k \cup \{S_k\}$ is consistent and S_k is not of the form $\exists x\colon s.A$ or $\neg\forall x\colon s.A$ then $\Gamma_i = \Gamma_k \cup \{S_k\}$. Evidently, the three conditions hold by inductive hypothesis and by construction of Γ_i since we are not adding more than a finite number of new constants, those appearing in S_k. ↪

Existence of Henkin sets

↪ Proof. (v)

- If $\Gamma_k \cup \{S_k\}$ is consistent and S_k has the form $\exists x\colon s.A$ or $\neg\forall x\colon s.A$ then by (3) there is $c\colon s$ in C not occurring in Γ_k and S_k.
  So, $\Gamma_i = \Gamma_k \cup \{S_k, B[c/x]\}$ with $B \equiv A$ when $S_k \equiv \exists x\colon s.A$, and $B \equiv \neg A$ when $S_k \equiv \neg\forall x\colon s.A$.
  Clearly, (1) and (3) hold for Γ_i because we are adding no more than a finite number of new constants.
  Suppose Γ_i non consistent. Then $\Gamma_k \cup \{S_k\} \vdash \neg B[c/x]$. Since c is new, it could be regarded as a variable free in the assumptions, so $\Gamma_k \cup \{S_k\} \vdash \forall x\colon s.\neg B$. If $S_k \equiv \exists x\colon s.A$, $B \equiv A$, then $\Gamma_k \cup \{S_k\} \vdash \bot$ by $\exists$-elimination. If $S_k \equiv \neg\forall x\colon s.A$, $B \equiv \neg A$, then $\Gamma_k \cup \{S_k\} \vdash \bot$ since $\neg B$ is equivalent to A. In both cases, $\Gamma_k \cup \{S_k\}$ is non consistent, contradicting the assumption. Thus, Γ_i must be consistent. ↪

Existence of Henkin sets

↪ Proof. (vi)

Let $\Delta = \Gamma_\lambda$. By (1) $\Gamma = \Gamma_0 \subset \Delta$, and by (2) Δ is consistent.

Let A be a formula on Σ' such that $A \notin \Delta$. Since $A \equiv S_k$ for some $k < \lambda$, Γ_{k+1} must not contain A, which means by construction of the sequence of Γ_i's that $\Gamma_k \cup \{A\}$ is non consistent, thus also $\Delta \cup \{A\}$ is non consistent. Therefore, Δ is maximal consistent.

If $\exists x\colon s.A \in \Delta$ then $\exists x\colon s.A \equiv S_k$ for some $k < \lambda$, so Γ_{k+1} contains $A[c/x]$ for some new constant $c\colon s$.
Similarly, if $\neg\forall x\colon s.A \in \Delta$ then $\neg\forall x\colon s.A \equiv S_k$ for some $k < \lambda$, so Γ_{k+1} contains $\neg A[c/x]$ for some new constant $c\colon s$. Thus, Δ is a Henkin set. □

Completeness

Theorem 11.2

If Γ is a consistent set of formulæ on a signature Σ then Γ is true on a model whose universe has a cardinality less or equal than the cardinality of the formulæ in the language on Σ.

Proof.

By Proposition 11.1 Γ can be extended to a Henkin set Δ. By Lemma 10.9 Δ, and thus Γ, has a model satisfying the cardinality constraints. □

Theorem 11.3 (Completeness)

If every model of Γ makes A true then $\Gamma \vdash A$.

Proof.

Clearly, if every model of Γ makes A true then $\Gamma \cup \{\neg A\}$ has no model. Thus, by Theorem 11.2 $\Gamma \cup \{\neg A\}$ is non consistent.
Then, by Proposition 10.3 $\Gamma \vdash A$. □

References

The first completeness proof for first-order logic has been given by Kurt Gödel. The proof presented in this lesson follows the techniques introduced by Leon Henkin.

Our treatment follows *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977).

Gödel's proof was his doctoral dissertation and it is based on an obscure formalism. Henkin's proof is a substantial reorganisation of Gödel's proof, emphasising that it involves the construction of a model.

Mathematical Logic

Lecture 12

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

First-order logic

- Compactness
- Löwenheim-Skolem Theorems
- Discussion

Compactness

Theorem 12.1 (Compactness)

For any set of formulæ Γ, if every finite subset of Γ has a model then Γ has a model too.

Proof.

By hypothesis, applying the Soundness Theorem 9.10 every finite subset of Γ is consistent.

Suppose Γ non consistent: then $\Gamma \vdash A$ and $\Gamma \vdash \neg A$. Since a finite number of assumptions occur in each proof, there are two finite subsets such that $\Gamma_1 \vdash A$ and $\Gamma_2 \vdash \neg A$. Consider $\Gamma_\omega = \Gamma_1 \cup \Gamma_2$. It is evidently finite and non consistent leading to a contradiction. Thus Γ must be consistent.

So by Theorem 11.2 Γ has a model. □

Compactness

Proposition 12.2

Fix a language with a single sort. If a set of sentences S has arbitrarily large finite models then it has an infinite model.

Proof.

Define $\tau_n = \exists x_1, \ldots, x_n . \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j$. Clearly, τ_n holds in any model whose universe has at least n distinct elements.

Consider any finite subset $F \subseteq S \cup \{\tau_n : n \in \mathbb{N}\}$.
Since F is finite, $m = \max\{n : \tau_n \in F\}$ is defined (pose $m = 0$ if F contains no τ_n). Thus, since S has arbitrarily large finite models by hypothesis, F must have a finite model larger than m.

Thus, by Theorem 12.1, $S \cup \{\tau_n : n \in \mathbb{N}\}$ has a model $\mathscr{M}$. Since τ_n must hold for every $n \in \mathbb{N}$, $\mathscr{M}$ must have more than n distinct elements in its universe for every $n \in \mathbb{N}$, thus it must be infinite.
Observe how $\mathscr{M}$ is a model of S to conclude. □

Examples

The signature of real numbers is set to have a single sort, the 0 constant, the plus and times operations, and equality and $<$ as relations. The theory R of real numbers is the collection of true formulæ on the model whose sort is the reals, and whose symbols are interpreted as one expects.

Example 12.3

There is a model of R in which infinity is a number.

Let us extend the signature with the new constant ∞. Let $T = \{n < \infty : n \in \mathbb{N}\}$ and consider the theory $R \cup T$.

If $F \subseteq R \cup T$ is finite then the maximum m such that either $(m < \infty) \in F$ or $m = 0$ is defined. Thus interpreting ∞ in $m+1$ in the standard model of reals validates F.

Hence, by the Compactenss Theorem 12.1 $R \cup T$ has a model, and ∞ must be interpreted in an element which is bigger than any natural number. The same model makes true R so it is an alternative model of reals with an infinite element.

Examples

Example 12.4

There is a model of reals with infinitesimals.

Let us extend the signature with the new constant k.
Let $T = \{0 < k < \frac{1}{n+1} : n \in \mathbb{N}\}$.

If $F \subseteq R \cup T$ is finite, there is the maximum m for which either $(0 < k < \frac{1}{m+1}) \in F$ or $m = 0$. Then interpreting k in $\frac{1}{m+2}$, F is valid in the standard model of reals.

Hence, by the Compactenss Theorem 12.1 $R \cup T$ has a model, and k has to be an infinitesimal. The same model validates R, so it is an alternative model of reals with an infinitesimal element.

Löwenheim-Skolem

Let $\aleph_0 = |\mathbb{N}|$. A classical result in Model Theory is

Theorem 12.5 (Downward Löwenheim-Skolem)

Let T be a theory on the signature Σ with just one sort. If T has an infinite model of cardinality $\alpha \geq |\Sigma|$ then T has a model of cardinality $\max(|\Sigma|, \aleph_0)$.

Proof.

Extend the signature Σ by adding $\max(|\Sigma|, \aleph_0)$ new constants k_i.
Let $T' = T \cup \{k_i \neq k_j : i \neq j\}$.

By the Completeness Theorem 11.2 T' has a model of cardinality less or equal than $\max(\aleph_0, |\Sigma|)$.

Conversely, since all the k_i must be distinct every model of T' must have a cardinality greater of equal than $\max(\aleph_0, |\Sigma|)$. □

Löwenheim-Skolem

Corollary 12.6

Any consistent theory T (on a single sort) such that $|T| \leq \aleph_0$ has a model whose universe has cardinality at most $\aleph_0$.

Proof.

Being consistent T has a model. Either T has an infinite model or it does not. In the latter case, the result is obtained.

In the former case, by Theorem 12.5 the result follows since the language can be limited to $\aleph_0$ using only the symbols appearing in the theory T, which form a countable set by hypothesis. □

Note how the Completeness Theorem 11.2 allows to prove a weaker result, since the model has the cardinality of the formulæ on the language, which can be bigger than $\aleph_0$ if the signature contains more than $\aleph_0$ symbols.

Löwenheim-Skolem

Theorem 12.7 (Upward Löwenheim-Skolem)

Let T be a theory on the signature Σ with just one sort. If T has a model of cardinality $\alpha \geq \aleph_0$ then T has a model of any cardinality $\beta \geq \max(\alpha, |\Sigma|)$.

Proof.

Fix any $\beta \geq \max(\alpha, |\Sigma|)$ and extend the signature Σ by adding β new constants k_i, $i < \beta$. Let $T' = T \cup \{k_i \neq k_j : i < j < \beta\}$. Clearly, T' is a theory on the extended signature.

Let $F \subseteq T'$ be any finite subset of T'. Since it contains only a finite number of axioms of the form $k_i \neq k_j$, F has a model because the model for T being infinite allows to validate the axioms $k_i \neq k_j$, and clearly it makes true the other axioms in F.

Thus by Compactness T' has a model $\mathcal{M}$ and it must contain at least β distinct elements. But by Theorem 12.5 there is model having exactly cardinality β, using the cardinality of the extended Σ as an upper bound. □

Löwenheim-Skolem

Theorem 12.8 (Löwenheim-Skolem)

Let T be a theory on the signature Σ with just one sort. If T has a model of cardinality $\alpha \geq \aleph_0$ then T has a model of each cardinality $\beta \geq \max(|\Sigma|, \aleph_0)$.

Proof.

Immediate by combining the upward and downward Löwenheim-Skolem theorems. □

Corollary 12.9

If T is a consistent theory on the signature Σ with just one sort then either T has a finite model or it has a model for any cardinality greater than $\max(|\Sigma|, \aleph_0)$.

Discussion

The Compactness Theorem 12.1 is a consequence of the completeness result. One of its consequences is Proposition 12.2.

Thus, it is **impossible** to write a first-order theory which captures the notion of having finite models only. Indeed, any theory T either has finite models with a limit on their cardinality, or it has at least an infinite model.

Hence, the compactness result reveals a first, intrinsic limit to what can be expressed in the first-order language.

Discussion

The Löwenheim-Skolem Theorems provide other limitations to what can be expressed in a first-order theory.

For example, Corollary 12.6 says that every 'effective' and consistent theory has a model whose cardinality is either finite or $\aleph_0$. Here, by 'effective' we mean really writable, thus at least, with finite or denumerable symbols.

As a concrete instance we get that the theory of real numbers as developed in any textbook of mathematical analysis, which can be formally rendered as a first-order theory, has a countable model, which is much smaller than $\mathbb{R}$.

Saying the same thing in another, provocative way, Mathematical Analysis does not speak about real (or complex) numbers. It speaks about an infinite set which is much smaller than $\mathbb{R}$ or $\mathbb{C}$. So small that it disregards most of the reals (or complex numbers), which play no rôle in Analysis.
[Analysts are greatly disturbed by this sentence but nevertheless it is true when we regard Mathematical Analysis as a formal theory!]

Discussion

Of course, we are investigating formal first-order theories. In this respect, the Löwenheim-Skolem Theorems say that not only every 'effective' theory has a finite or countable model, but if it has an infinite model, it has a model of any infinite cardinality.

This has a deep impact. Consider for example a formal and effective theory of arithmetic. Natural numbers form an obvious model and the theory is intuitively consistent.
So, by Corollary 12.9 it has models of any infinite cardinality.

In other words, **without even writing the formal theory**, as far as we require it to be effective we know that it does not capture only the model of natural numbers. It must have models for each cardinal above $\aleph_0$.

Comparing models

Let Σ be a signature with just one sort and let T be a theory.
We have seen that T may have more than one model.

This means that we have a way to distinguish models. From the outside of a theory this is obvious. But, from the inside?

If $\mathfrak{M}$ and $\mathfrak{N}$ are both models for T and they are distinct, we would like to find a formula δ in the language on Σ which holds in $\mathfrak{M}$ but is false in $\mathfrak{N}$.

The question is: can we always find such a formula?

Comparing models

Completeness is a property of a formal system which says that whatever is true in any model, it can be derived.

But there is an alternative notion of completeness which says

Definition 12.10 (Completeness)

A theory T on the signature Σ is *complete* if for every sentence ϕ on the same language, either ϕ is true in any model of T, or $\neg\phi$ is true in any model of T.

Here, by 'sentence' we mean a first-order formula with no free variables. Hence it does not depend on the interpretation of variables, which simplifies the analysis.

Also we will write $T \vDash \phi$ to say that every model of T makes ϕ true.

So we have another question: are the two notions of completeness equivalent?

Comparing models

Example 12.11
The simplest example of complete theory is $\mathrm{Th}(\mathfrak{M}) = \{\phi : \phi \text{ is true in } \mathfrak{M}\}$ with $\mathfrak{M}$ any model on the signature Σ.

The key in the example is that since we are working in classical logic, every sentence is either true or false in a model. So given two models $\mathfrak{M}$ and $\mathfrak{N}$, we can compare them by comparing $\mathrm{Th}(\mathfrak{M})$ and $\mathrm{Th}(\mathfrak{N})$. When these theories are different we know the models are different too. And there is at least one sentence $\delta \notin \mathrm{Th}(\mathfrak{M}) \cap \mathrm{Th}(\mathfrak{N})$, which can be used to distinguish the models.

But, when they are equal?

Comparing models

Actually, the answer is simple: if a model $\mathfrak{M}$ is infinite then the theory $\mathrm{Th}(\mathfrak{M})$ must have models of any infinite cardinality beyond the size of the language by Theorem 12.7.

If we take one of those models, call it $\mathfrak{N}$, whose size is greater than the cardinality of $\mathfrak{M}$ we know that these models are distinct.

Consider $\mathrm{Th}(\mathfrak{N})$: since $\mathfrak{N}$ validates each formula in $\mathfrak{M}$, it makes true $\mathrm{Th}(\mathfrak{M})$, that is for every $\phi \in \mathrm{Th}(\mathfrak{M})$ it holds that $\phi \in \mathrm{Th}(\mathfrak{N})$.

Since every sentence ϕ in the language on Σ is either in $\mathrm{Th}(\mathfrak{M})$ or $\neg\phi \in \mathrm{Th}(\mathfrak{M})$ then $\mathrm{Th}(\mathfrak{M}) = \mathrm{Th}(\mathfrak{N})$.

So, we may have **different** models which are indistinguishable by what we can express in the language.

Our counterexample shows that the models are distinguishable because they have different cardinality, a fact that cannot be expressed *inside* a theory.

References

The notion of compactness is fundamental in model theory since it allows to construct models of an infinite theory by considering only finite subsets of formulæ. This fact turns out to be critical in many situations. A good starting reference is *Wilfrid Hodges*, A Shorter Model Theory, Cambridge University Press, (1997).

The exposition of Löwenheim-Skolem theorems follows *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977) omitting the parts on elementary equivalence of models.

A comprehensive text on model theory which is approachable and contains many examples of the application of logic to other fields, is *David Marker*, Model Theory: An Introduction, Graduate Texts in Mathematics 217, Springer (2002).

Mathematical Logic

Lecture 13

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Set theory:

- Language
- Classes and sets
- Paradoxes
- Comparing sets

Language

The language of the theory of sets is the classical first order language with equality plus one additional symbol: $\in$. The corresponding signature is

$$\langle \{S\}; \emptyset; \{=: S \times S, \in: S \times S\} \rangle$$

Since there is a unique sort we omit sort specifications from the syntax.

The intended meaning is that S stands for the collection of all possible sets while $\in$ denotes membership.
Note how there are no objects apart sets in the universe.

Language

It is important to distinguish between *formal* set theory which is the first order theory we are going to introduce, and *informal* set theory which is used to describe the formal theory.

Although the former intends to model the latter, the latter is assumed in the definition of the former. With this distinction in mind we cannot say that set theory is constructed out of itself.

As we have already seen set theory admits a countable model, so the collection of all sets seen 'from the outside' may be assumed to have the same cardinality as the natural numbers. But looking 'from the inside', the collection of all sets is much bigger.

This is just one of the various bizarre phenomena we should expect when dealing with the formal theory.

Language

The basic language of set theory is very poor so it is enriched via a number of definitions which are universally quantified:

- x not equal to y, $x \neq y$ abbreviates $\neg x = y$;
- x not in y, $x \notin y$ abbreviates $\neg x \in y$;
- x is a subset of y, $x \subseteq y$ abbreviates $\forall z. z \in x \supset z \in y$;
- there is x in y such that A, $\exists x \in y. A$ abbreviates $\exists x. x \in y \wedge A$;
- for all x in y, A, $\forall x \in y. A$ abbreviates $\forall x. x \in y \supset A$;
- for some subset x of y, A, $\exists x \subseteq y. A$ abbreviates $\exists x. x \subseteq y \wedge A$;
- for every subset x of y, A, $\forall x \subseteq y. A$ abbreviates $\forall x. x \subseteq y \supset A$.

Classes and sets

Informally a set is a collection of elements. Although this is very intuitive and helpful, the structure of a set is much more subtle and delicate.

We stipulate that collections of elements are called *classes*. This is part of the intended meaning of set theory. Sets in the intended meaning are classes which behave in a *regular* way.

As we will see there are classes which cannot be sets, while all sets are also classes in the intended meaning. Each formal set has an *extension*, which is the class representing the collection of its element in the intended model of the theory. But a set is **not** its extension, although we would like to say the converse, that is to every extension corresponds a unique set.

As we will see sets will have properties not shared by classes, e.g., sets should have a *cardinality* while proper classes cannot. These properties are what identify the *structure* of sets, and they are what we are allowed to use when proving properties of sets, or when using sets in our proofs.

Paradoxes

A very simple theorem we will be able to derive in set theory will be: for any formula A such that $x \notin \mathrm{FV}(A)$,

$$(\exists x.\forall y.(y \in x) = A) \supset (\exists! x.\forall y.(y \in x) = A) \quad .$$

It means that when there is a set x whose members are exactly those making the formula A true then the set x is uniquely defined. In other words the property A *defines* the set x.

It is tempting to carry on this result by thinking that any formula A defines a set. This amounts to assume

$$\exists x.\forall y.(y \in x) = A$$

as an axiom schema. This schema is called the unrestricted *Comprehension Axiom* and it has been used to define sets by Gottlob Frege.

Paradoxes

Unfortunately the unrestricted Comprehension Axiom is untenable as shown by **Russell's paradox**: take $A \equiv y \notin y$. Then by the axiom we have $\exists x.\forall y.y \in x = y \notin y$, and specialising we obtain $\exists x.x \in x = x \notin x$, allowing to derive $\bot$, i.e., showing that the theory of sets is non consistent.

It is important to understand the key point: the collection of sets making A true is a class. To be a set it has to show a 'reasonable' behaviour. In logical terms a minimal reasonable behaviour is not to allow to derive a contradiction.

Thus, what the Russell's paradox tells is

- there are classes which are not sets;
- every formula uniquely identifies a class: the elements which make it true. This class may be *proper* that is, not a set.

Paradoxes

Sets are a delicate concept. When we fix a universe which is a set and we do mathematics within that universe, we do not see the problems sets pose. But when we consider the totality of sets, things change.

Consider the following reasoning:

1. Let $X = \{x : x \in x \supset Y\}$
2. Thus, $X \in X$ is equivalent to $X \in X \supset Y$
3. Then, from $X \in X \supset (X \in X \supset Y)$ one gets the equivalent $X \in X \wedge X \in X \supset Y$, that is $X \in X \supset Y$
4. Hence, from $(X \in X \supset Y) \supset X \in X$ and the previous step, one infers $X \in X$
5. Therefore, Y holds by steps 2 and 4

Since Y can be any formula, fix $Y \equiv \bot$ and set theory becomes non-consistent. This is known as **Curry's paradox**, and step 2 is the wrong part since it assumes X to be a set.

Paradoxes

Sets and properties as already seen are linked but different. Consider for example the **hyper-game paradox**. Let G be the collection of all games which can be played by two players by making successive alternate moves. A game in G is said to be *finite* if in whatever way the players move, the game terminates after a finite number of steps. When a game is not finite it is said to be *infinite*.

Take tic-tac-toe: it must end at most after 9 moves so it is a finite game.

Define the *super-game* as the game in which the first player chooses a game $g \in G$, and then the second player starts playing g.

Is the super-game finite?

Paradoxes

Since the first player may choose an infinite game, the super-game is clearly infinite. So define a variant: the *hyper-game* is played like the super-game, but the first player must choose a finite game.

Since the first player chooses a finite game g then the hyper-game takes one move more than the moves to conclude g. But the moves to conclude g are always finite so the hyper-game is finite.

Hence, the first player may choose the hyper-game as the game to play and the second player may do the same. Forever. So the hyper-game is infinite.

Thus the first player cannot choose the hyper-game being infinite, and thus the hyper-game always terminate in a finite number of steps.

The problem here is that the collection of all finite games is a class and we define the hyper-game as a particular element which depends on the whole class. This is something we want to do but, as the paradox shows, it cannot be freely done with classes: a certain amount of 'regularity' in the class is needed to define an element which depends on it.

Comparing sets

Although many other paradoxes can be formed on sets, most of them require some knowledge that we have not yet explained.

Comparing two sets means to establish a correspondence between them. A function, mapping all the elements of one set in the elements of another does not say much. But when the function is bijective, we may think that the two sets are equal except for a renaming of the elements in their extensions. We write $A \cong B$ to indicate that there is bijective map between the sets A and B.

Intuitively a set A is smaller than a set B when it can be embedded into B modulo a renaming: formally this intuition is modelled by the existence of an injective function $A \to B$. Symmetrically A is greater than B when there is a surjective function $A \to B$.

Comparing sets

This way of comparing sets is standard and it works as one expects when dealing with finite sets. But on infinite sets it reveals that sets are far more complex objects than we may imagine at a first sight.

Theorem 13.1 (Schröder-Bernstein)

If $f\colon A \to B$ is injective and $g\colon B \to A$ is injective then $A \cong B$.

Proof. (i)

Let $C_0 = A \setminus g(B)$ and by induction $C_{n+1} = \{g(x)\colon x \in D_n\}$ and $D_n = \{f(x)\colon x \in C_n\}$. Define $h\colon A \to B$

$$h(x) = \begin{cases} f(x) & \text{if } x \in C_n \text{ for some } n \\ g^{-1}(x) & \text{otherwise} \end{cases}$$

This definition makes sense as $g^{-1}(x)$ is defined on $g(B)$. ↪

Comparing sets

↪ Proof. (ii)

Let $x, y \in A$. Suppose $h(x) = h(y)$:

- if $x \in C_m$ and $y \in C_k$ for some m and k then $h(x) = f(x) = f(y) = h(y)$, so $x = y$ being f injective;
- if $x \notin C_n$ and $y \notin C_n$ for any n then $h(x) = g^{-1}(x) = g^{-1}(y) = h(y)$, so $g(g^{-1}(x)) = x = y = g(g^{-1}(y))$;
- if $x \in C_m$ for some m and $y \notin C_n$ for any n, $h(x) = f(x) = g^{-1}(y) = h(y)$, so $(g \circ f)(x) = y$, that is, $y \in C_{m+1}$ which is impossible.

Thus h is injective. ↪

Comparing sets

↪ Proof. (iii)

We must show that $h(A) = B$.
Observe that for any n and any $z \in D_n$, $z = f(x)$ for some $x \in C_n$, so by definition $z = h(x)$.
Then let $z \in B \setminus \bigcup_n D_n$. Evidently, $g(z) \notin C_n$ for any n (otherwise $z \in D_n$ if $n > 0$, and $C_0 = A \setminus g(B)$), thus $h(g(z)) = g^{-1}(g(z)) = z$.
So h is surjective. □

It is surprising how difficult is to prove this result, which is completely elementary in the finite case using, e.g., the pigeon hole principle.

Comparing sets

Example 13.2

Let $P = \{2n : n \in \mathbb{N}\}$.
Since $f : P \to \mathbb{N}$ such that $f(x) = x$ is injective, and $g : \mathbb{N} \to P$ such that $g(x) = 2x$ is injective, by Theorem 13.1 we conclude that $P \cong \mathbb{N}$.

In general, an infinite set A is such that it is possible to find a proper subset $B \subset A$ such that $A \cong B$.
We can even use this property as a *definition* of being infinite.

Comparing sets

Example 13.3

$\mathbb{N} \times \mathbb{N} \cong \mathbb{N}$

Evidently, the function $f : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ mapping $x \mapsto (x, x)$ is injective.
Oppositely, the function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined as $g(x, y) = (x + y)(x + y + 1)/2 + y$ is injective as it is easy to prove. Informally it counts the pairs using diagonals which justifies the claim of being injective: the formal proof is just arithmetic.
Thus, by Theorem 13.1 the result follows.

This result can be generalised to arbitrary infinite sets, although the proof requires some technicalities.

A simpler result, which is immediately obtained by induction, is that $\mathbb{N}^k \cong \mathbb{N}$ for any $k > 0$.

Comparing sets

Example 13.4
The collection of finite sequences of naturals $\mathbb{N}^* \cong \mathbb{N}$

Obviously, the function $f : \mathbb{N} \to \mathbb{N}^*$ mapping $x \mapsto \{x\}$ is injective.
Oppositely, calling $g_n : \mathbb{N}^n \to \mathbb{N}$ the bijection from the Cartesian product of $n \geq 1$ copies of $\mathbb{N}$ to $\mathbb{N}$, we may define a function $h : \mathbb{N}^* \to \mathbb{N} \times \mathbb{N}$ by $h\left(\{x_i\}_{1 \leq i \leq n}\right) = (n, g_n(x_1, \ldots, x_n))$. For $n = 0$ let $h(\emptyset) = (0,0)$.
Evidently h is injective since g_n is for each $n \geq 1$. So the composition $g_2 \circ h$ is injective and the result follows by Theorem 13.1.

Again, the result can be generalised to arbitrary infinite sets, essentially by the same proof.

Comparing sets

An application of what has been obtained till now to logic is immediate: let Σ be a signature with a finite number of symbols. Since the variables of sort s are in a bijective correspondence with $\mathbb{N}$, the collection of all variables is in bijection with $\mathbb{N}$.

Then the sequences of symbols given by the function symbols, the parentheses, the commas, and the variables is in bijection with $\mathbb{N}$. So the collection of all terms on Σ being an infinite subset of that set, is in bijection with $\mathbb{N}$ too.

Analogously, the collection of all formulæ on Σ being an infinite subset of the collection of sequences of symbols of Σ plus a finite set of logical symbols, is in bijection with $\mathbb{N}$.

All these result can be easily extended to arbitrary signatures, using the generalised versions of the previous examples.

Comparing sets

Example 13.5
$\wp(\mathbb{N}) \ncong \mathbb{N}$.

This result, which specialises a famous theorem by Cantor, says that the collection of subsets of $\mathbb{N}$ is **not** in bijection with $\mathbb{N}$. The proof is a classical masterpiece that introduces a technique called *diagonalisation*.

We can identify each subset $A \subseteq \mathbb{N}$ with its characteristic function $\chi_A : \mathbb{N} \to \{0,1\}$. Suppose that all these functions are in bijection with $\mathbb{N}$. So we have $\wp(\mathbb{N}) \cong \{\chi_{A_i}\}_{i \in \mathbb{N}}$. Observe how each function $f : \mathbb{N} \to \{0,1\}$ uniquely identifies the subset $\{x : f(x) = 1\} \subseteq \mathbb{N}$.
Define $\Delta : \mathbb{N} \to \{0,1\}$ as $\Delta(x) = 1 - \chi_{A_x}(x)$. Thus Δ must appear somewhere in the sequence, i.e., $\Delta = \chi_{A_k}$ for some $k \in \mathbb{N}$. Which is impossible since $\chi_{A_k}(k) = \Delta(k) = 1 - \chi_{A_k}(k)$ and $\chi_{A_k} \in \{0,1\}$. Hence the characteristic functions are not in bijection with $\mathbb{N}$, that is $\wp(\mathbb{N}) \ncong \mathbb{N}$.

As a side effect, since the functions $\mathbb{N} \to \{0,1\}$ are in bijection with the real interval $[0,1]$ we get that $\mathbb{R} > \mathbb{N}$ strictly. In other words, infinity is not unique!

References

Probably, the best introductory text to set theory is *Paul Halmos*, Naive Set Theory. D. Van Nostrand Company, (1960), reprinted by Springer-Verlag, (1974), reprinted by Martino Fine Books (2011).

Mathematical Logic

Lecture 14

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Set theory:

- Axioms
- Ordinals
- Induction
- Arithmetic

Axiomatic set theory

The axioms of Zermelo-Frænkel set theory are presented and discussed in the following slides.

As said in the previous lecture, they are deputed to model sets but not classes, avoiding the formation of paradoxes, which arise when a naive notion of sets is adopted.

Also, in the background, we already know that formal sets will be bizarre mathematical objects, in which infinity is not unique. To start analysing this fact, which proves to be fundamental to sketch and use set theory, we will introduce the notion of *ordinal*.

Axioms: extensionality

Informally a set is uniquely determined by its extension. This fact is captured by the following axiom:

Axiom (Extensionality)

$\forall x.\forall y.\,(\forall z.(z \in x) = (z \in y)) \supset x = y.$

Proposition 14.1

If $x \notin \mathrm{FV}(A)$ *then* $\vdash (\exists x.\forall y.(y \in x) = A) \supset (\exists! x.\forall y.(y \in x) = A)$.

Proof.

The formal proof is easy but long to write down. Essentially if z is another set satisfying $\forall y.(y \in z) = A$, it must be that $x = z$ by extensionality. □

The content of the proposition is that whenever the collection of the y's satisfying a formula A corresponds to the extension of a set, it identifies a unique set.

Axioms: empty set

Axiom (Empty set)

$\exists x. \forall y. y \notin x.$

Since by Proposition 14.1 the set x is unique, we will denote it by $\emptyset$ as usual. This axiom establishes that there is at least one set, the empty one.

Axioms: pairs

Axiom (Pair)

$\forall x. \forall y. \exists z. \forall u. (u \in z) = (u = x \vee u = y).$

This axiom says that given two elements x and y we can form the set z whose extension contain exactly x and y. Again, we adopt the standard notation $\{x, y\}$ since by extensionality a pair set is uniquely identified.

Note that when $x = y$, we have *singletons*, $\{x\}$.

Axioms: union

Axiom (Union)

$\forall x. \exists y. \forall z. (z \in y) = (\exists u \in x. z \in u).$

The axiom says that given a set x, we can form another set y whose extension is the collection of elements in the members of x. Since as usual, the set y is unique by extensionality, we adopt the standard notation $\bigcup x$ for it, or also, we write $\{z : \exists u \in x. z \in u\}$, or also $\bigcup_{u \in x} u$. When x is a pair $\{A, B\}$ we write $A \cup B$ for y.

Observe how, up to now, all the sets which have been stated to exist by the axioms, are finite.

Axioms: infinity

Axiom (Infinity)

$\exists x. \emptyset \in x \wedge \forall y. y \in x \supset y \cup \{y\} \in x.$

In general we will write $\mathrm{Succ}(x)$ for $x \cup \{x\}$, and we will call it the *successor* of x. The axiom says that there is at least one set which is non empty containing the empty set and which is closed under the successor operation.

It is possible to formally prove by extensionality that there is a unique set that satisfies the axiom minimally, that is, its extension is minimal among all the collections containing the empty set and closed under the successor operation. This set is clearly in biijection with the set of natural numbers. We will denote this minimal set as ω in the following.

Axioms: power set

Axiom (Power set)

$\forall x.\exists y.\forall z.(z \in y) = (z \subseteq x)$.

The power set of x has as extension the collection of all the subsets of x. We will denote it as $\wp(x)$, or also $\{z\colon z \subseteq x\}$.

By extensionality we get that if $\wp(x) = x$ then $\forall y \in \wp(x).y \in x$, but $x \in \wp(x)$ so $x \in x$. Thus, as this behaviour is something we want to ban from our set theory in order to prevent paradoxes, we want to introduce an axiom which forbids this phenomenon to happen. The consequence will be that $\wp(x) \neq x$ for every set x.

Axioms: regularity

Axiom (Regularity)

$\forall x.x \neq \emptyset \supset \exists y \in x.\neg\exists z.z \in x \wedge z \in y$.

Similarly to extensionality and differently from the preceding axioms, regularity states a property of all non empty sets instead of providing a way to construct new sets. Precisely, it says that each non empty set x contains an element y which is disjoint from x.

It is a bit technical to show and beyond the aims of this course, but the axioms prevents the construction of circular chains of membership, banning the existence of a set x satisfying $x \in x$, or $x \in y \in x$, ...

Thus paradoxes like the hyper-game and Russell's cannot be constructed in the framework of formal set theory.

Axioms: separation

Axiom (Separation)

Let P be a formula such that $\mathrm{FV}(P) = \{u\}$ *then*
$\forall x.\exists y.\forall z.(z \in y) = (z \in x \wedge P[z/u])$.

Properly speaking separation provides an *axiom schema*, i.e., a family of axioms one for each possible instance of P.

It says that given a set x, the collection of elements in x satisfying P is the extension of a set y.
Sometimes, this axiom is called restricted or bounded comprehension.

An immediate application is the construction of intersection: $A \cap B$ is defined as the set formed by separation from A applying the property $P(u) = (u \in B)$.

Another immediate application is the intensional construction of subsets: $\{x \in A\colon P\}$ is the result of applying separation to A with the property P.

Axioms: replacement

Axiom (Replacement)

Let P be a formula such that $\mathrm{FV}(P) = \{x, y\}$ *then*
$(\forall x.\exists! y.P) \supset \forall z.\exists u.\forall y.(y \in u) = (\exists x \in z.P)$.

It says that whenever P behaves like a function mapping x to y, the image of any set x through P is a set.

Again, replacement is an axiom schema whose instance are defined as soon as P is given.

Further definitions

With these fundamental definitions together with their justifying axioms, we can easily define the usual operations on sets like difference, Cartesian product, sequence, ...

The set theory developed so far is interesting by itself: it is called **ZF**, for Zermelo-Frænkel, its creators.

Although set theory is an important branch of mathematical logic, its development is far beyond the aim of this course and involves some of the most stunning results of XX$^{\text{th}}$ century.

As a matter of fact, the collection of axioms we have shown so far is enough to develop most of elementary mathematics with a few exceptions for which in the following we will introduce another couple of axioms. In particular the so-called *Axiom of Choice* has a special rôle as it is needed to prove some fundamental results in algebra, although it is also responsible for theorems which are really counter-intuitive like the Tarski-Banach Theorem.

Well orders

Definition 14.2

An order $\langle A;\leq\rangle$ is *total* when for each pair $x,y\in A$ either $x\leq y$ or $y\leq x$.

Definition 14.3

An order $\langle A;\leq\rangle$ is a *well order* when every non empty subset $S\subseteq A$ has a minimum, i.e., there is $m\in S$ such that for every $x\in S$, $m\leq x$.

Fixed a set A it is always possible to add a relation to it so to make it an order, e.g., take $\leq$ to be equality. Also, it is immediate to define an order relation on A which makes it a total order, e.g., take $\leq$ to be the set $A\times A$. But it is not clear whether it is always possible to define an order relation which makes it a well order.

However, a well order, as we will see soon, allows for an induction principle that is a very powerful instrument to reason about the set and its properties.

Ordinals

Definition 14.4

A set S is an *ordinal* if and only if $\langle S;\in\cup=\rangle$ is a total well order and for each $x\in S$, $x\subseteq S$.

Thus an ordinal S is a set totally well ordered by the strict order given by $\in$ and, moreover, if $x\in S$ then $x\subseteq S$.

Ordinals

Proposition 14.5

Every ordinal S is totally well ordered by inclusion.

Proof.

Consider the structure $\langle S;\subseteq\rangle$. Clearly, $\subseteq$ forms an ordering relation. Also, being S an ordinal, for each $A,B\in S$,

- $A=B$ or
- $A\in B$, which implies for all $x\in A$, $x\in B$ by transitivity, i.e., $A\subseteq B$, or
- $B\in A$, which implies by the same argument $B\subseteq A$.

So, the structure is totally ordered.
Moreover being S an ordinal, for each non empty $A\subseteq S$ there is $m\in A$ such that for all $x\in A$ either $m=x$ or $m\in x$, that is, $m\subseteq x$. So, A is well ordered by inclusion, too. □

Ordinals

Proposition 14.6

If S is an ordinal and $x \in S$ then x is an ordinal and $S = \bigcup_{x \in S} x$.

Proof.

Immediate since $x \in S$ implies $x \subseteq S$ being S an ordinal. □

Proposition 14.7

The collection of all ordinals is not a set.

Proof.

Suppose $\mathrm{Ord} = \{x : x \text{ is an ordinal}\}$ is a set. Then it is immediate to check that Ord must be an ordinal. So $\mathrm{Ord} \in \mathrm{Ord}$ contradicting regularity. □

Admitting Ord to be a set generates a contradiction.
This argument is called the *Burali-Forti* paradox.

Transfinite induction

Proposition 14.6 intuitively justifies

Principle 14.8 (Transfinite induction)

If P is a property and assuming that P holds for every ordinal less than α, we can prove that P holds for α, then P holds for any ordinal.

This principle can be relativised to all the ordinals less than some fixed ordinal β, leading to

Principle 14.9 (Transfinite induction)

If P is a property and assuming that P holds for every ordinal $\alpha < \beta$, we can prove that P holds for α, then P holds for any ordinal less than β.

Transfinite induction

We have to prove that transfinite induction is a sound principle, that is, it does not allow to derive false consequences from true statements.

Proposition 14.10

If P is a property and assuming that P holds for every ordinal less than α, we can prove that P holds for α, then P holds for any ordinal.

Proof.

Assume that if $P(x)$ is true for every ordinal $x \in \alpha$ then $P(\alpha)$ holds. And by contradiction assume there is an ordinal β for which $P(\beta)$ is false.

Since β is an ordinal, it is well-ordered. Then there exists the minimal ordinal $\gamma \leq \beta$ such that $P(\gamma)$ is false.

Being γ minimal, for every $x \in \gamma$, $P(x)$ is true. So by hypothesis $P(\gamma)$ holds, which contradicts the existence of γ, and thus, the existence of β. □

The relativised principle is an immediate corollary by considering the property $\beta \leq x \vee P(x)$.

Transfinite induction

Since $\emptyset$ is an ordinal and whenever x is an ordinal, its successor $x \cup \{x\}$ is an ordinal too, we can classify ordinals in three classes:

- the empty ordinal $\emptyset$;
- the *successor ordinals* x such that there is an ordinal y for which $x = y \cup \{y\}$;
- the *limit ordinals* x, which are those ones not falling in the previous classes. These are characterised by $x = \bigcup_{y<x} y$.

It is worth reminding that the set of natural numbers is in bijection with ω, the ordinal containing $\emptyset$ and closed under the successor operation.

Transfinite induction

Principle 14.11 (Transfinite induction)

If P is a property and

- *if P holds for $\emptyset$;*
- *supposing P holds for an ordinal x then P holds for the successor of x;*
- *supposing P holds for any ordinal $y < x$ with x a limit ordinal then P holds also for x;*

we can conclude that P holds for any ordinal. Of course, as before, the principle can be relativised to the ordinals less than β.

Transfinite induction is a powerful instrument to reason about infinite sets: we already used it to prove the Completeness Theorem 11.3 for first order logic.

Also, note how the usual induction principle on natural numbers is equivalent to the transfinite induction principle relativised to ω.

Transfinite induction

Proposition 14.12

If α and β are ordinals and $\alpha \cong \beta$ monotonically then $\alpha = \beta$.

Proof.

Let $f : \alpha \to \beta$ be a monotone bijection between the ordinals whose is inverse is monotone. Consider the property $P(x) \equiv \forall u \in \mathrm{Ord}. x \cong u \supset x = u$.

By transfinite induction on α we show $P(\alpha)$.
If $x \in \alpha$ then $f(x) \in \beta$, but also $x \subseteq \alpha$ since α is an ordinal. The restriction of the bijection f to x is a monotone bijection so $f(x) \cong x$ and by induction hypothesis $f(x) = x$, thus $x \in \beta$. Hence, $\alpha \subseteq \beta$.

By transfinite induction on β we show $P(\beta)$.
If $y \in \beta$ then $f^{-1}(y) \in \alpha$ but also $y \subseteq \beta$ since β is an ordinal. The restriction of the bijection f^{-1} to y is a monotone bijection so $f^{-1}(y) \cong y$, and by induction hypothesis $f^{-1}(y) = y$, thus $y \in \alpha$. Hence, $\beta \subseteq \alpha$. □

Transfinite induction

We state without proving the following fact, which is deduced by a rather involved transfinite induction

Proposition 14.13

If $\mathbb{S} = \langle S; \leq \rangle$ is a total well order then there is a unique ordinal α such that $\alpha \cong S$ monotonically.

Ordinal arithmetic

Definition 14.14 (Ordinal addition)

Let α and β be ordinals then $\alpha + \beta$ is the unique ordinal such that there is $h : S \to \alpha + \beta$ biijective and monotone, i.e., such that $x \leq y$ in S implies $h(x) \leq h(y)$ in $\alpha + \beta$ where $S = \langle \alpha \sqcup \beta; \leq \rangle$, the disjoint union of α and β, and $x \leq y$ if and only if $x \leq y$ in α, or $x \leq y$ in β, or $x \in \alpha$ and $y \in \beta$.

On finite ordinals, i.e., on natural numbers it is just arithmetical addition. But on infinite ordinals, it is not commutative. For example $1 + \omega = \omega$ but $\omega + 1 \neq \omega$ since $\omega + 1$ has a maximum, while ω has not.

The intuition one should keep in mind is that $\alpha + \beta$ is α followed by β.

Ordinal arithmetic

We state without proof the following properties of ordinal sum.

Proposition 14.15

Let α, β, and γ be ordinals. Then

1. $\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma$;
2. $\alpha+0=\alpha$;
3. $\alpha+1=Succ(\alpha)$;
4. $\alpha+Succ(\beta)=Succ(\alpha+\beta)$;
5. *if β is a limit ordinal then* $\alpha+\beta=\bigcup_{\xi<\beta}(\alpha+\xi)$.

Ordinal arithmetic

Definition 14.16 (Ordinal multiplication)

Let α and β be ordinals then $\alpha\beta$ is the unique ordinal such that there is $h\colon S\to\alpha\beta$ biijective and monotone where $S=\langle\bigsqcup_{i\in\beta}\alpha;\leq\rangle$ with $x\leq y$ in S when either $x\in\alpha_i$ and $y\in\alpha_j$ with $i<j$, or $x,y\in\alpha_i$ and $x\leq y$ in α.

On finite ordinals it is just arithmetical multiplication, but on infinite ordinals it is not commutative. For example 2ω is the total order formed by ω copies of $0<1$, so $2\omega=\omega$. On the contrary, $\omega 2=\omega+\omega\neq\omega$ since there is a limit ordinal, ω, inside $\omega+\omega$ while there is none in ω.

The intuition behind ordinal multiplication is that $\alpha\beta$ is the ordinal consisting of the sequence composed by β copies of α.

Ordinal arithmetic

We state, without proving

Proposition 14.17

Let α, β, and γ be ordinals. Then

- $\alpha(\beta\gamma)=(\alpha\beta)\gamma$;
- $\alpha 0=0$;
- $\alpha 1=\alpha$.
- $\alpha Succ(\beta)=\alpha\beta+\alpha$;
- *If β is a limit ordinal,* $\alpha\beta=\bigcup_{\xi\in\beta}(\alpha\xi)$;
- $\alpha(\beta+\gamma)=\alpha\beta+\alpha\gamma$.

Note how most of these properties do not commute when ordinals are infinite. For example it is possible that $(\beta+\gamma)\alpha\neq\beta\alpha+\gamma\alpha$: indeed

$$(1+1)\omega=2\omega=\omega\neq 1\omega+1\omega=\omega+\omega\ .$$

References

The axioms of set theory and the treatment of ordinals derive from the presentation in *Kenneth Kunen*, Set Theory: An Introduction to Independence Proofs, Studies in Logic and the Foundations of Mathematics 102, Elsevier, (1980). This book covers very advanced material, which lies far beyond the scope of the course. An alternative introduction can be found in *Jon Barwise*, Handbook of Mathematical Logic, Studies in Logic and the Foundations of Mathematics 90, North-Holland, (1977).

The theory **ZF** has been first proposed by Ernst Zermelo in 1908. Then, Abraham Fraenkel in 1921 pointed out that the original theory was not able to prove a number of natural properties of sets so he and Thoralf Skolem in 1922 independently proposed an improved formulation, the one we introduced.

Ordinals form in a sense the backbone of set theory providing the main tool to prove properties of sets at large: transfinite induction.

Mathematical Logic

Lecture 15

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Set theory:

- Cardinals
- Arithmetic

Comparing sets, again

Definition 15.1

For any pair of sets A and B, $A \preceq B$ if and only if there is an injective function $A \to B$. Also, we write $A \approx B$ when there is a bijective function $A \to B$. Finally $A \prec B$ when $A \preceq B$ but $B \not\approx A$.

Proposition 15.2

The relation $\preceq$ is reflexive and transitive, while $\approx$ is an equivalence relation.

Proof.

Since the identity function is bijective, $x \preceq x$ and $x \approx x$. Since the composition of injective (bijective) functions is injective (bijective), $\preceq$ ($\approx$) is transitive. Finally, since the inverse of a bijective function is bijective, $\approx$ is symmetric. □

Theorem 15.3 (Schröder-Bernstein)

If $A \preceq B$ and $B \preceq A$ then $A \approx B$.

Cardinals

Definition 15.4 (Cardinality)

If the set A can be well ordered, $|A|$, the *cardinality* of A is the least ordinal α such that $A \approx \alpha$.

Observe how, when A can be well ordered, it holds that $A \approx \alpha$ for some ordinal α which depends on the well ordering, see Proposition 14.13. Forming the set of ordinals $\{\alpha : A \approx \alpha\}$ it has a minimum, so the definition of cardinality is well-founded.

Definition 15.5 (Cardinal)

An ordinal α is a *cardinal* if and only if $\alpha = |\alpha|$.

Equivalently, the ordinal α is a cardinal whenever for all $\beta \in \alpha$, $\beta \not\approx \alpha$.

Cardinals

Proposition 15.6

Let α and β be ordinals. If $|\alpha| \leq \beta \leq \alpha$ then $|\alpha| = |\beta|$.

Proof.

Since $\beta \leq \alpha$ then $\beta \subseteq \alpha$, thus $\beta \preceq \alpha$. Also $\alpha \approx |\alpha|$ by definition of cardinality and $|\alpha| \leq \beta$ implies $|\alpha| \subseteq \beta$, thus $\alpha \preceq \beta$. Then $\alpha \approx \beta$ by Theorem 15.3. Thus $|\alpha| \approx \alpha \approx \beta \approx |\beta|$ so $|\alpha| = |\beta|$ by Proposition 14.12. □

Proposition 15.7

If $n \in \omega$ then $n \not\approx n+1$ and for every ordinal α, if $\alpha \approx n$, then $\alpha = n$.

Proof.

By induction on n it follows immediately that $n \not\approx n+1$. The second part is an instance of Proposition 15.6 by noting that $|n| = n$. □

Cardinals

Corollary 15.8

Each $n \in \omega$ is a cardinal and ω is a cardinal.

Definition 15.9

A set A is *finite* if and only if $|A| < \omega$; A is *countable* if and only if $|A| \leq \omega$. *Infinite* means not finite, and *uncountable* means not countable.

Note that when $A \approx \alpha$ with α an ordinal, then A can be well ordered by the relation which is the image of $\subseteq$ through the bijection $\alpha \to A$.
Hence, $|A|$ is defined.

If A cannot be well ordered, which is possible in the framework described so far, A is both infinite and uncountable.

Cardinal arithmetic

Definition 15.10

Let α and β be cardinals. Then $\alpha \oplus \beta = |\alpha \sqcup \beta|$ and $\alpha \otimes \beta = |\alpha \times \beta|$.

Note how cardinal addition and cardinal product are different from ordinal addition and product.

Proposition 15.11

Cardinal addition and product are associative and commutative operations with units.

Proof.

Since $\alpha \sqcup \beta \approx \beta \sqcup \alpha$ and $\alpha \times \beta \approx \beta \times \alpha$, commutativity follows. Also, associativity derives from the corresponding property of $\sqcup$ and $\times$, up to $\approx$. It is immediate to check that 0 and 1 are the units of addition and multiplication, respectively. □

Cardinal arithmetic

Proposition 15.12

Let α and β be cardinals. Then

1. $|\alpha + \beta| = |\beta + \alpha| = \alpha \oplus \beta$;
2. $|\alpha\beta| = |\beta\alpha| = \alpha \otimes \beta$.

Proof.

Immediate unfolding the definitions of the ordinal and cardinal sum and product. □

Proposition 15.13

For $n, m \in \omega$, $n \oplus m = n + m$ and $n \otimes m = nm$.

Proof.

By induction on $m \in \omega$. □

Cardinal arithmetic

Proposition 15.14

Every infinite cardinal is a limit ordinal.

Proof.

If α is an infinite cardinal and $\alpha = \beta + 1$, since $1 + \beta = \beta$, $\alpha = |\alpha| = |\beta + 1| = |1 + \beta| = |\beta|$, a contradiction. □

We state without proving

Proposition 15.15

If α is an infinite cardinal, $\alpha \otimes \alpha = \alpha$.

Corollary 15.16

Let either α or β be an infinite cardinal. Then $\alpha \oplus \beta = \alpha \otimes \beta = \max\{\alpha, \beta\}$.

Cardinal arithmetic

Theorem 15.17 (Cantor)

For any set A, $A \prec \wp(A)$.

Proof.

Clearly, $A \preceq \wp(A)$ by the mapping $x \in A \mapsto \{x\}$.
Suppose there is a surjective map $f : A \to \wp(A)$, and define
$B = \{x \in A : x \notin f(x)\}$, which is a set by the Separation Axiom.
Since $B \subseteq A$, $B \in \wp(A)$, thus there is a $y \in A$ such that $f(y) = B$.
Now, if $y \in B$ then $y \in f(y)$, which is impossible by definition of B.
Conversely, if $y \notin B$ then $y \in B$ by definition of B, another contradiction.
Thus f cannot be surjective. □

Hierarchy of cardinals

We state without proving that

Proposition 15.18

For every cardinal α there is a cardinal β such that $\alpha < \beta$ strictly.

Definition 15.19

For any cardinal α, α^+ is the least cardinal strictly greater than α.
We say that the cardinal β is a *successor* cardinal when $\beta = \alpha^+$ for some cardinal α.
We say that the cardinal β is a *limit* cardinal when $\beta > \omega$ and β is not a successor cardinal.

Hierarchy of cardinals

Definition 15.20

By transfinite induction define the map $\aleph$ from ordinals to infinite cardinals:

- $\aleph_0 = \omega$;
- $\aleph_{\alpha+1} = (\aleph_\alpha)^+$;
- for γ a limit ordinal, $\aleph_\gamma = \bigcup_{\alpha<\gamma} \aleph_\alpha$.

By transfinite induction on the ordinal α one shows

Proposition 15.21

Each $\aleph_\alpha$ is a cardinal and every infinite cardinal equals $\aleph_\alpha$ for some α. Also, the map $\aleph$ is monotone, $\aleph_\alpha$ is a limit cardinal if and only if α is a limit ordinal, and $\aleph_\alpha$ is a successor cardinal exactly when α is a successor ordinal.

Hierarchy of cardinals

Proposition 15.22

$\aleph_1 \preceq \wp(\aleph_0)$.

Proof.
By Definition 15.20 $\aleph_0 < \aleph_1$. By Theorem 15.17 $\aleph_0 \prec \wp(\aleph_0)$.
By definition $\aleph_1$ is the least cardinal greater than $\aleph_0$, so $\aleph_1 \leq |\wp(\aleph_0)| \approx \wp(\aleph_0)$, i.e. $\aleph_1 \preceq \wp(\aleph_0)$. □

This result can be immediately extended to any ordinal α.

Since, but we are not going to prove this fact, the collection of functions from the cardinal α to 2, the finite cardinal composed by two distinct elements, has the same cardinality as $\wp(\alpha)$, the notation $2^\alpha = \wp(\alpha)$ is common.

References

The presentation of cardinals derives from *Kenneth Kunen*, Set Theory: An Introduction to Independence Proofs, Studies in Logic and the Foundations of Mathematics 102, Elsevier, (1980). An alternative introduction can be found in *Jon Barwise*, Handbook of Mathematical Logic, Studies in Logic and the Foundations of Mathematics 90, North-Holland, (1977).

Mathematical Logic

Lecture 16

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Set theory:

- Axiom of choice
- The continuum hypothesis
- What is a set?

Axiom of Choice

We have mentioned the Axiom of Choice many times. In most cases we said that this principle allows to say that any set can be well ordered, or equivalently that any set is in bijection with a cardinal.

Axiom (Choice)

For any non empty family $\{X_i\}_{i\in I}$ of non empty sets such that $X_i \cap X_j = \emptyset$ for any $i, j \in I$, $i \neq j$, there exists a function $f : I \to \bigcup_{i\in I} X_i$ such that $f(i) \in X_i$ for every $i \in I$.

The meaning is that, whenever we are given such a family, we have the ability to make a choice that simultaneously picks an element from each set.

Although this principle seems very natural it cannot be derived from the **ZF** set theory. So when we adopt this axiom, we will speak of **ZFC**, the Zermelo-Frænkel set theory with the Axiom of Choice.

Axiom of Choice

As a matter of fact, when I, the index set of the family, is finite the Axiom of Choice can be derived from **ZF** by induction on $|I|$. But when I is infinite, this is not possible in general.

Some important results in Mathematics require the Axiom of Choice to be proved: as a small collection of examples

- every non empty vector space has a base;
- every field has an algebraic closure, which is unique modulo isomorphisms;
- the notion of adjunction in category theory;
- the compactness theorem in first order logic.

Axiom of Choice

But the Axiom of Choice allows to prove critical results, like the the Tarski-Banach theorem.

Its geometric form is: given a sphere S in the usual 3-dimensional Euclidean space, it is possible to divide it into a finite set of pieces so to obtain, using only rotations and translations, a reassembling of those pieces in two spheres both identical to S.

Of course, this seems to be impossible since we consider pieces which are measurable, or if you prefer, they possess a volume. On the other hand, if we take pieces, i.e., subspaces of the sphere for which the notion of volume is meaningless, the above composition becomes possible. In the proof the pieces are constructed using the Axiom of Choice.

Axiom of Choice

There a number of equivalent formulation of the Axiom of Choice: the most common and useful ones are

- the Well Ordering Theorem
- the Zorn Lemma
- Hartogs's Theorem
- the Cartesian product of a non empty family $\{X_i\}_{i\in I}$ of non empty sets, is non empty.

The last form is relaxed version, which is easily derived by imposing disjointness via the isomorphism $X_i \approx \{i\} \times X_i$.

Well ordering theorem

Theorem 16.1 (Well ordering)

For any set X, $X \approx |X|$.

Proof.

By the Axiom of Choice there is function $c\colon \wp(X)\setminus\{\emptyset\} \to \bigcup\wp(X) = X$ such that for every non empty $S \subseteq X$, $c(S) \in S$.
By transfinite induction we define a bijection s between X and some ordinal α: assuming $s(\beta)$ has been defined for all $\beta \in \alpha$, if $X \setminus \{s(\beta)\colon \beta \in \alpha\} \neq \emptyset$ then $s(\alpha) = c(X \setminus \{s(\beta)\colon \beta \in \alpha\})$.
Observe how $s(\alpha) \neq s(\beta)$ for every $\beta \in \alpha$, thus s is injective. Also, when $X \setminus \{s(\beta)\colon \beta \in \alpha\} = \emptyset$, $s\colon \alpha \to X$ is surjective.
Suppose $X \setminus \{s(\beta)\colon \beta \in \alpha\} \neq \emptyset$ for every ordinal α. Then s defines a functional map from Ord to X which is invertible on some $Y \subseteq X$, the image of Ord in X. Hence, by the Replacement Axiom $s^{-1}(Y)$ is a set since X is a set and thus so are all its subsets by the Separation Axiom. But $s^{-1}(Y) = \mathrm{Ord}$, a proper class, thus we have a contradiction.
By definition $|X|$ is the least ordinal which is in bijection with X, and we know that there is at least one, α. □

Well ordering theorem

Assuming the Well Ordering Theorem as an axiom we can prove the Axiom of Choice: let $\mathscr{F}$ be a non empty family of non empty, pairwise disjoint sets.
Consider $\bigcup_{X\in\mathscr{F}} X$: by the Well Ordering Theorem for each $X \in \mathscr{F}$, $X \approx I_X$ for some ordinal I_X, that is, there is $g_X\colon I_X \to X$ bijective.
Then we can define a choice function $f\colon \mathscr{F} \to \bigcup_{X\in\mathscr{F}} X$ as $f(X) = g_X(\emptyset)$.

Zorn lemma

Theorem 16.2 (Zorn Lemma)

If $\langle X;\leq\rangle$ is a non empty order such that every proper ordered subset has an upper bound then $\langle X;\leq\rangle$ contains a maximal element, i.e., an element which is not smaller than any other element in X.

Theorem 16.3 (Hartogs)

If A and B are two sets it holds that either $A \preceq B$ or $B \preceq A$.

Although we are not going to prove these results, they shed some light to the meaning of the Axiom of Choice: indeed, they say that the notion of cardinality takes the usual, intuitive meaning only when we assume that principle to hold.

For this reason when no set theory is specified usually **ZFC** is intended.

Continuum Hypothesis

Another axiom which is commonly considered in the theory of sets is the so-called *Continuum Hypothesis*:

Axiom (Continuum Hypothesis)

$\aleph_1 = 2^{\aleph_0}$.

It admits an obvious generalisation:

Axiom (Generalised Continuum Hypothesis)

$\aleph_{i+1} = 2^{\aleph_i}$ *for every ordinal i.*

Although the generalised Continuum Hypothesis implies the plain version the converse does not hold. And both the versions are independent from **ZFC**, that is, they cannot be proved from the axioms of **ZFC** nor it can be proved them to be false.

Continuum Hypothesis

While the Axiom of Choice justifies the intuitive notion of cardinality, the (generalised) Continuum Hypothesis is more technical and not easy to accept.

In fact, assuming the Continuum Hyothesis the collection of all sets becomes a quite regular structure. On the contrary assuming the Continuum Hypothesis to be false, the collection of all sets provides a very rich universe.

Intuition does not help: the effects of the Continuum Hypothesis are sensible for *large* sets and the trade between regularity and wealth becomes difficult. In the common practice of higher set theory, which is far beyond the scope of this course, the Continuum Hypothesis is generally assumed not to hold, although some weaker regularity conditions may be considered.

What is a set?

As we said in the beginning the notion of set is not simple.

The intuitive notion of a set as a collection of elements does not work because of Russell's paradox. So, formal theories like **ZFC** have been introduced.

In those theories a large number of principles, like the Axiom of Choice or the Continuum Hypothesis, are admissible but not provable: they are consistent with the theory but also their negation is consistent with it.

So, *at least from the formal point of view* we do not know exactly what is a set. We have a variety of structures (theories if you prefer) that provide a reasonable notion of set. In some of these structures we are able to prove results which are difficult to accept, like the Tarski-Banach Theorem. But avoiding the principles underlying these structures, like the Axiom of Choice, we loose some basic, intuitive notion like the cardinality of a set.

References

A nice reference to elementary set theory, which explains the nature of the Axiom of Choice with some detail is *Patrick Suppes*, Axiomatic Set Theory, Dover Publishing, (1972).

The classical text *Kenneth Kunen*, Set Theory: An Introduction to Independence Proofs, Studies in Logic and the Foundations of Mathematics 102, Elsevier, (1980) provides a more in-depth discussion extending far beyond the limits of this course.

Another reference of interest is *Nicholas Bourbaki*, Elements of Mathematics: Theory of Sets, Springer, (1968).

The continuum hypothesis is the main subject of the essays in *Paul Joseph Cohen*, Set Theory and the Continuum Hypothesis, Dover Publishing, (2008). This text contains the proof that the continuum hypothesis is independent from the other axioms of **ZFC**. Students should be warned that its content is advanced material.

Mathematical Logic

Lecture 17

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Computability:

- Motivation
- Recursive functions
- Main properties

Motivation

Computability theory is the branch of logic which studies the notion of 'computation'. Generally, it is considered at the borderline between mathematics and theoretical computer science, but at least historically, it has been the part of logic from which computer science was born.

From a mathematical point of view describing what can be really computed is an essential part of the XX^{th} century's mathematics. Consider the notion of algorithm and how fundamental it revealed in many fields.

For logicians computability theory is an essential ingredient to understand the reasons behind constructive mathematics. But it is also the fundamental tool to prove the results about the limits of formal reasoning.

Computable functions

Computability theory aims at describing the functions $\mathbb{N} \to \mathbb{N}$ which can be effectively calculated.

We observe how the vast majority of functions from naturals to naturals cannot be calculated. Indeed if we think that calculation is a process which mechanically transforms the argument of a function in its result, we have to pose a few limits on this process:

- it must take a finite amount of time;
- it must operate on a finitely generated formal language;
- it must rely on a finite description of the process which precisely describes the steps to be performed.

Computable functions

We have a language, which is used to describe the process, on a finite or countable alphabet. No matter how we interpret the language we know that the set of all the possible procedures is contained in the collection of finite sequences of symbols in the alphabet. So, the cardinality of the language is at most $\aleph_0$ since the alphabet is just finite or countable. It is evident that it is at least $\aleph_0$ as we may write an infinite amount of procedures. But the cardinality of the set of functions from $\mathbb{N}$ to $\mathbb{N}$ is $2^{|\mathbb{N}|} = 2^{\aleph_0}$, which is strictly greater than $\aleph_0$. So most functions are **not** computable.

Computable functions

There are many ways to describe computations. For our purposes, which are not aimed at studying computations themselves but rather using the computable functions to reason about what can be effectively proved inside a formal system, we will use *partial recursive functions*.

In fact, we admit a computation may not terminate, hence partial functions, in which non termination is modelled as the function being undefined for the non terminating input.

Instead of using some abstract machine which 'performs' the computation we will directly define computable functions as the class of functions that can be written in a special form. Although it is not immediately clear that this class contains all the computable functions, it is best suited to application in logic.

Primitive recursive functions

Definition 17.1 (Primitive recursive functions)

A function $f : \mathbb{N}^k \to \mathbb{N}$ is *primitive recursive* when

1. f is the *zero* function $\underline{0}(n) = 0$ for all $n \in \mathbb{N}$;
2. f is the *successor* function $\mathsf{S}(n) = n+1$ for all $n \in \mathbb{N}$;
3. f is a *projection* function $U_i^k(n_1, \ldots, n_k) = n_i$ with $k \geq 1$, $1 \leq i \leq k$;
4. f is obtained by *substitution*: if $g, h_0, \ldots, h_m$ are primitive recursive functions, $f(n_1, \ldots, n_k) = g(h_0(n_1, \ldots, n_k), \ldots, h_m(n_1, \ldots, n_k))$;
5. f is obtained by *primitive recursion*: if g and h are primitive recursive functions, $f(n_1, \ldots, n_k, 0) = g(n_1, \ldots, n_k)$ and $f(n_1, \ldots, n_k, m+1) = h(n_1, \ldots, n_k, m, f(n_1, \ldots, n_k, m))$.

It is clear that primitive recursive functions are computable. It is also evident that there are computable functions which are not primitive recursive: for example the function everywhere undefined.

Observe how function composition $f \circ g$ is a special case of substitution.

Primitive recursive functions

Example 17.2
The *identity* function $\mathrm{id}(x) = x$ is primitive recursive: $\mathrm{id} = U_1^1$.

Example 17.3
The constant function $\underline{k}(x) = k$ is primitive recursive. Indeed by induction on k, if $k = 0$, $\underline{0}$ is primitive recursive by definition; if $k = k' + 1$, $\underline{k} = \mathsf{S} \circ \underline{k'}$ by substitution and $\underline{k'}$ is primitive recursive by induction hypothesis.

Example 17.4
Addition, multiplication and exponentiation are primitive recursive.

$$n + 0 = \mathrm{id}(n) \qquad n \cdot 0 = \underline{0}(n)$$
$$n + (m+1) = \mathsf{S}\left(U_3^3(n, m, n+m)\right) \qquad n \cdot (m+1) = m + \underline{0}(n) + m \cdot n$$
$$n^0 = \underline{1}(n)$$
$$n^{m+1} = n \cdot \underline{1}(m) \cdot n^m$$

Note how $0^0 = 1$, which sounds odd.

Primitive recursive functions

Example 17.5
The *predecessor* function defined by

$$\mathrm{pred}(n) = \begin{cases} n-1 & \text{when } n > 0 \\ 0 & \text{otherwise} \end{cases}$$

is primitive recursive: $\mathrm{pred}(0) = \underline{0}(0)$, and $\mathrm{pred}(n+1) = U_1^2(n, \mathrm{pred}(n))$.

Example 17.6
The *recursive difference* defined by

$$m \dot{-} n = \begin{cases} m-n & \text{if } m \geq n \\ 0 & \text{otherwise} \end{cases}$$

is primitive recursive: $m \dot{-} 0 = m$ and $m \dot{-} (n+1) = \mathrm{pred}(m \dot{-} n)$.

Primitive recursive functions

Example 17.7
The *absolute difference* $|m-n|$ is primitive recursive:

$$|m-n| = (m \dot{-} n) + (n \dot{-} m) \ .$$

Example 17.8
The *sign* function defined by

$$\mathrm{sg}(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{otherwise} \end{cases}$$

is primitive recursive: $\mathrm{sg}(0) = \underline{0}(0)$, and $\mathrm{sg}(n+1) = U_2^2(n, \underline{1}(n))$.

Primitive recursive functions

Example 17.9
Integer division and the remainder function are primitive recursive: write $x/y = d(y,x)$ and $x \bmod y = r(y,x)$, then

$$\begin{aligned} r(n,0) &= 0 \\ r(n,m+1) &= \mathrm{sg}(n \dot{-} \mathsf{S}(r(n,m))) \cdot \mathsf{S}(r(n,m)) \ , \end{aligned}$$

and

$$\begin{aligned} d(n,0) &= 0 \\ d(n,m+1) &= d(n,m) + \mathrm{sg}(n \dot{-} \mathsf{S}(r(n,m))) \ . \end{aligned}$$

Again, $0/0 = 0$ and $0 \bmod 0 = 0$ which sounds pretty bizarre.

Primitive recursive functions

Example 17.10
The integer logarithm is primitive recursive:

$$\begin{aligned} \log_b(0) &= 0 \\ \log_b(n+1) &= \log_b(n) + \mathrm{sg}\left(\mathsf{S}(n) \dot{-} b^{\mathsf{S}(\log_b(n))}\right) \ . \end{aligned}$$

Note how the logarithm is defined everywhere in $\mathbb{N} \times \mathbb{N}$, in opposition to the standard definition in Mathematics.

Primitive recursive functions

There are functions which are computable but not primitive recursive.

Definition 17.11 (Ackermann)
The *Ackermann's function* A is defined as

$$\begin{aligned} A(m,0) &= m+1 \\ A(0,n+1) &= A(1,n) \\ A(m+1,n+1) &= A(A(m,n+1),n) \ . \end{aligned}$$

To give an impression: $A(0,0) = 1$, $A(1,1) = 3$, $A(2,2) = 7$, $A(3,3) = 61$, but

$$A(4,4) = 2^{2^{65536}} \ .$$

The function $\mathbb{N} \to \mathbb{N}$ given by $n \mapsto A(n,n)$ can be shown to grow faster than any primitive recursive function, so it is **not** primitive recursive.

Partial recursive functions

Definition 17.12 (Partial recursive functions)

A partial function $f : \mathbb{N}^k \to \mathbb{N}$ is *recursive* when

1. f is the *zero* function $\underline{0}(n) = 0$ for all $n \in \mathbb{N}$;
2. f is the *successor* function $\mathsf{S}(n) = n+1$ for all $n \in \mathbb{N}$;
3. f is a *projection* function $U^k_i(n_1,\ldots,n_k) = n_i$ with $k \geq 1$, $1 \leq i \leq k$;
4. f is obtained by *substitution*: if $g, h_0, \ldots, h_m$ are partial recursive functions, $f(n_1,\ldots,n_k) = g(h_0(n_1,\ldots,n_k),\ldots,h_m(n_1,\ldots,n_k))$;
5. f is obtained by *primitive recursion*: if g and h are partial recursive functions, $f(n_1,\ldots,n_k,0) = g(n_1,\ldots,n_k)$ and $f(n_1,\ldots,n_k,m+1) = h(n_1,\ldots,n_k,m,f(n_1,\ldots,n_k,m))$;
6. f is obtained by *minimalisation*: if g is a partial recursive function, then $f(n_1,\ldots,n_k) = \mu m.\,(g(n_1,\ldots,n_k,m) = 0)$, with $(\mu m.\,P(m)) = m_0$ if and only if $P(m_0)$ holds and for all $m < m_0$, $P(m)$ does not.

We will speak of *recursive* functions when we will consider only computable total functions.

Partial recursive functions

Definition 17.13

Let S be a set and R a relation. The *characteristic functions* of S and R are given by

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S \end{cases}$$

$$\chi_R(x_1,\ldots,x_n) = \begin{cases} 1 & \text{if } (x_1,\ldots,x_n) \in R \\ 0 & \text{otherwise} \end{cases}$$

We say that S or R is *recursive* when χ_S or χ_R are total recursive functions. We say they are *primitive recursive* when the corresponding characteristic functions are.

Example 17.14

The relation $\leq \subseteq \mathbb{N} \times \mathbb{N}$ is primitive recursive: $\chi_{\leq}(n,m) = 1 \dot{-} \mathrm{sg}(n \dot{-} m)$.

Partial recursive functions

Example 17.15

If P and Q are (primitive) recursive relations on $\mathbb{N}^k$, then so are $\neg P$, $P \wedge Q$, and $P \vee Q$.

$$\begin{aligned} \chi_{\neg P}(x_1,\ldots,x_k) &= 1 \dot{-} \chi_P(x_1,\ldots,x_k) \\ \chi_{P \wedge Q}(x_1,\ldots,x_k) &= \chi_P(x_1,\ldots,x_k) \cdot \chi_Q(x_1,\ldots,x_k) \\ \chi_{P \vee Q}(x_1,\ldots,x_k) &= \mathrm{sg}(\chi_P(x_1,\ldots,x_k) + \chi_Q(x_1,\ldots,x_k)) \quad . \end{aligned}$$

Example 17.16

Every finite set is primitive recursive.

Example 17.17

If R and S are primitive recursive subsets of $\mathbb{N}$, so are $\mathbb{N} \setminus R$, $R \cap S$, and $R \cup S$.

Partial recursive functions

Proposition 17.18

If $R(n_1,\ldots,n_k,m)$ is a recursive relation, then $f : \mathbb{N}^k \to \mathbb{N}$ defined by

$$f(n_1,\ldots,n_k) = \mu m.\, R(n_1,\ldots,n_k,m)$$

i.e., the least m such that $R(n_1,\ldots,n_k,m)$ holds, is partial recursive.

Proof.

Immediate by noting that $f(n_1,\ldots,n_k) = \mu m.\,(\chi_{\neg R}(n_1,\ldots,n_k,m) = 0)$. □

Church-Turing Thesis

A function $f : \mathbb{N}^k \to \mathbb{N}$ is *computable* exactly when f is partial recursive.

Universal function

Theorem 17.19 (Enumeration)

There is a partial recursive function $e(x,y)$ that enumerates all the partial recursive functions, that is, defining $\phi_x(y) = e(x,y)$, $\{\phi_x\}_{x\in\mathbb{N}}$ is the collection of all the partial recursive functions.

Proof. (i)

In the first place, we note that since for any $k \in \mathbb{N}$, $\mathbb{N}^k \cong \mathbb{N}$ and the bijection is computable, we may enumerate the computable functions $\mathbb{N} \to \mathbb{N}$ only.

Partial recursive functions can be coded as naturals:

- $[\underline{0}] = 2$;
- $[\underline{S}] = 3$
- $[U_i^k] = 5 \cdot 17^k \cdot 19^i$;
- substitution:
  $[g(h_0(n_1,\ldots,n_k),\ldots,h_m(n_1,\ldots,n_k))] = 7 \cdot 17^{[g]} \cdot 19^{[h_0]} \cdot \ldots \cdot p_{7+m}^{[h_m]}$, with $\{p_i\}_{i\in\mathbb{N}}$ the sequence of prime numbers;

↪

Universal function

↪ **Proof. (ii)**

- primitive recursion: $[f] = 11 \cdot 17^{[g]} \cdot 19^{[h]}$;
- minimalisation: $[f] = 13 \cdot 17^{[g]}$.

The coding is injective thus invertible thanks to the unique factorisation in primes of any natural number. Moreover, it is computable and the inverse is computable, too. Precisely, the coding is primitive recursive, as it is immediate to check.

Defining $\underline{\perp}$ as the partial function which is everywhere undefined, we can invert the $[\_]$ coding:

$$\phi_n = \begin{cases} f & \text{if there is } f \text{ such that } [f] = n \\ \underline{\perp} & \text{otherwise} \end{cases}$$

Since $\underline{\perp}(x) = \mu m.(\underline{1}(x) = 0)$ the decoding is computable.

Then, e is defined by $e(x,y) \equiv \phi_x(y)$.
It enjoys the enumeration property by construction. □

Universal function

Proposition 17.20

There is no enumeration $\{f_x\}_{x\in\mathbb{N}}$ of all total computable functions which admits a computable enumeration function $e(x,z) = f_x(z)$.

Proof.

Consider the function $h(x) = f_x(x) + 1$. It is total since each f_x is.
Assume there is a recursive function e enumerating $\{f_x\}_{x\in\mathbb{N}}$.
Then $h(x) = e(x,x) + 1$, so h is recursive.
Thus h occurs in $\{f_x\}_{x\in\mathbb{N}}$, so there is $k \in \mathbb{N}$ such that $f_k = h$.
Thus $h(k) = e(k,k) + 1 = f_k(k) + 1 = h(k) + 1$ hence $0 = 1$, a contradiction. □

Universal function

Theorem 17.21

Let $m, n \geq 1$. Then there is a computable function $S_n^m : \mathbb{N}^{m+1} \to \mathbb{N}$ such that

$$f_\alpha(x_1,\ldots,x_m,y_1,\ldots,y_n) = f_{S_n^m(\alpha,x_1,\ldots,x_m)}(y_1,\ldots,y_n) \ .$$

Although we will not prove the theorem we want to remark its meaning: it shows that considering some arguments as parameters is an admissible operation in the computational world.

We can start the study of computable functions by considering a *good* enumeration of all of them which has a couple of properties: being computable, and satisfying the S_n^m theorem. Then

Theorem 17.22 (Turing, 1936)

There is a computable partial function $U : \mathbb{N}^2 \to \mathbb{N}$ such that $f_n(x) = U(n,x)$.

Such a function is called *universal* and it is the first computer. But this is another story...

Fixed points

Theorem 17.23 (Kleene)

If f is a computable partial function then exists $k \in \mathbb{N}$ in any good enumeration of the partial recursive functions such that $\phi_{f(k)} = \phi_k$ whenever $f(k)$ is defined.

Proof.

Let $h(x) = \phi_x(x)$. This partial function is computable because it can be written as $h(x) = U(x,x)$. Then $f \circ h$ is computable too.
So, $f \circ h = \phi_e$ for some $e \in \mathbb{N}$.
Therefore $\phi_{f(h(e))} = \phi_{\phi_e(e)} = \phi_{h(e)}$.
Thus $k = h(e)$ is the sought fixed point. □

References

Computability theory also known as recursion theory is a major branch of mathematical logic. A very nice introductory text is *Barry Cooper*, Computability Theory, Chapman & Hall/CRC Mathematics, (2004).

This lecture is mainly based on that text.

Mathematical Logic

Lecture 18

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Computability theory:

- λ calculus

λ-calculus

The λ-calculus is a family of formal systems based on Alonzo Church's work in the 1930s. These systems are deputed to describe computable functions using the simplest syntax. Surprisingly not only they describe computable functions, but when equipped with types they show a hidden and deep link between logic and computability.

In this lectures, we want to introduce the λ-calculus and its simplest typed version. Our aim is to illustrate the general aspects of the theory and to derive a few results we will use in the following lessons.

In many cases we will avoid proving all the results we will introduce. This is done on purpose: the simplicity of the formal system has as a natural counterpart a deep and complex technical development. Although this technical part has many pearls, which shed light to some important aspects of computability, it lies beyond the aims of this course.

λ-term

Definition 18.1 (λ-term)

Fixed a set V of *variables*, which has to be both infinite and recursive, a *λ-term* is inductively defined as:

- any $x \in V$ is a λ-term and $\mathrm{FV}(x) = \{x\}$;
- if M and N are λ-terms, so is $(M \cdot N)$ called *application* and $\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$;
- if $x \in V$ and M is a λ-term, so is $(\lambda x.M)$ called *abstraction* and $\mathrm{FV}(\lambda x.M) = \mathrm{FV}(M) \setminus \{x\}$.

The set $\mathrm{FV}(M)$ is called the set of *free variables* in M and the variables in M not occurring in $\mathrm{FV}(M)$ are said to be *bound*.

Example 18.2

$(\lambda x.x)$ is a λ-term with no free variables representing the identity function.

λ-term

As usual, to simplify notation we introduce a number of conventions:

- outermost parentheses are not written: $\lambda x.x$ instead of $(\lambda x.x)$;
- a sequence of consecutive abstractions is grouped: $\lambda x,y.x \cdot y$ instead of $\lambda x.(\lambda y.x \cdot y)$;
- we treat application as a product omitting the dot: xy instead of $x \cdot y$;
- we assume application associates to the left: xyz instead of $(xy)z$.

Also, we use the term *combinator* to denote a λ-term having no free variables.

Example 18.3

The following are combinators

- $I \equiv \lambda x.x$;
- $K \equiv \lambda x,y.x$;
- $S \equiv \lambda x,y,z.(xz)(yz)$;
- $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$.

Substitution

Definition 18.4 (Substitution)

For any M, N λ-terms and x variable, $M[N/x]$ is the *substitution* of x with N in M defined by induction on M as:

- $x[N/x] \equiv N$;
- $y[N/x] \equiv y$ when $x \not\equiv y$;
- $(PQ)[N/x] \equiv (P[N/x])(Q[N/x])$;
- $(\lambda x.P)[N/x] \equiv \lambda x.P$;
- $(\lambda y.P)[N/x] \equiv \lambda y.P[N/x]$ when $x \not\equiv y$ and $y \notin \mathrm{FV}(N)$;
- $(\lambda y.P)[N/x] \equiv \lambda z.(P[z/y])[N/x]$ when $x \not\equiv y$ and $y \in \mathrm{FV}(N)$ and $z \notin \mathrm{FV}(P) \cup \mathrm{FV}(N)$.

In the last clause the z variable is said to be *new* and it is always possible to choose a z which satisfies the constraint.

The purpose of the last clause is to prevent variable capturing.

α-equivalence

Definition 18.5 (α-equivalence)

The λ-terms M and N are α-equivalent, $M =_\alpha N$ when

- $M \equiv N$;
- $M \equiv PQ$, $N \equiv P'Q'$ and $P =_\alpha P'$, $Q =_\alpha Q'$;
- $M \equiv \lambda x.P$, $N \equiv \lambda x.P'$ and $P =_\alpha P'$;
- $M \equiv \lambda x.P$, $N \equiv \lambda y.P$, and $P =_\alpha P'[x/y]$, $P[y/x] =_\alpha P'$.

So two λ-terms are α-equivalent when they differ for the names of bound variables only.

From now on, we identify terms which are α-equivalent.

α-equivalence

It is immediate to see that α-equivalence is an equivalence relation, but it is also a *congruence* with respect to substitution:

Proposition 18.6

If $M =_\alpha M'$ and $N =_\alpha N'$ then $M[N/x] =_\alpha M'[N'/x]$.

Therefore, using α-equivalence as equality between λ-terms is sound.

As a side note, we observe that α-equivalence is *decidable*, i.e., there is a recursive function to decide whether two λ-terms are α-equivalent.

β-reduction

Definition 18.7 (β-reduction)

The binary relation between λ-terms $M \rhd_{1,\beta} N$, spelt *M β-reduces to N in one step*, holds if and only if $M \equiv M'[(\lambda x.P)\cdot Q/z]$ and $N \equiv M'[(P[Q/x])/z]$, where the z variable occurs in M' exactly once.
We say that *M β-reduces* to N, $M \rhd_\beta N$, when there is a finite sequence $P_1, \ldots, P_n$ such that $M = P_1$, $N = P_n$ and for each $1 \leq i < n$, $P_i \rhd_{1,\beta} P_{i+1}$.

In the λ-calculus computation is performed by β-reduction.

Definition 18.8 (β-normal form)

A term N is said to be in *β-normal form* when it does not contain any subterm of the form $(\lambda x.P)Q$.

With respect to computations λ-terms in β-normal form represent the values.

Church-Rosser theorem

Theorem 18.9 (Church-Rosser)

If $M \rhd_\beta P$ and $M \rhd_\beta Q$ then there is a λ-term R such that $P \rhd_\beta R$ and $Q \rhd_\beta R$.

Corollary 18.10

If $M \rhd_\beta N$ and N is a β-normal form then N is unique up to α-equivalence.

Church-Rosser Theorem and its corollary say that although the computation in λ-calculus is non-deterministic, the resulting value, when it exists, is uniquely determined.

β-equality

Definition 18.11 (β-equality)

We say that P is β-equivalent to Q, $P =_\beta Q$, when there is a finite sequence $R_1, \ldots, R_n$ such that

- $P \equiv R_1$,
- $Q \equiv R_n$,
- for all $1 \leq i < n$, $R_i \rhd_{1,\beta} R_{i+1}$, or $R_{i+1} \rhd_{1,\beta} R_i$.

β-equivalence models the fact that two λ-terms are equal as computations.

It is easy to prove that β-equality is an equivalence relation, and a congruence with respect to substitution.

It is significant that β-equality, in general, is not decidable, i.e., its characteristic function is not computable.

Fixed point theorem

Theorem 18.12 (Fixed point)

There is a combinator $\mathbf{Y}$ *such that* $\mathbf{Y}x =_\beta x(\mathbf{Y}x)$.

Proof.
Let $U \equiv \lambda u, x.x(uux)$ and let $\mathbf{Y} \equiv UU$. Then
$\mathbf{Y}x \equiv (\lambda u, x.x(uux))Ux \rhd_\beta (\lambda x.x(UUx))x \rhd_\beta x(UUx) \equiv x(\mathbf{Y}x)$. □

The proof of the fixed point theorem as above is due to Alan Turing.

The fixed point theorem says that every λ-term, when thought of as a function, has a fixed point which is calculated by the $\mathbf{Y}$ combinator. This is an important property which suggests that each function which can be represented as a λ-term, has to be continuous in an appropriate space.

Representable functions

Definition 18.13 (Numerals)

For every $n \in \mathbb{N}$, the *Church numeral* $\overline{n}$ is a λ-term inductively defined as:

- $\overline{0} = \lambda x, y.y$;
- $\overline{n+1} = \lambda x, y.x(\overline{n}xy)$.

Definition 18.14 (Representable functions)

Let $f : \mathbb{N}^k \to \mathbb{N}$ be a partial function. A λ-term F is said to *represent* the function f when

- for all $n_1, \ldots, n_k \in \mathbb{N}$ if $f(n_1, \ldots, n_k) = m$ then $F\overline{n_1}, \ldots, \overline{n_k} =_\beta \overline{m}$;
- for all $n_1, \ldots, n_k \in \mathbb{N}$ if $f(n_1, \ldots, n_k)$ is undefined then $F\overline{n_1}, \ldots, \overline{n_k}$ has no β-normal form.

Theorem 18.15

Every partial recursive function can be represented in the λ-calculus.

Representable functions

The proof of the theorem is difficult beyond the aim of this course. But we will show a few examples to justify it.

Example 18.16
The successor function is represented by $\lambda x, s, z.s(xsz)$.
Addition is represented by $\lambda x, y, s, z.xs(ysz)$.
Multiplication is represented by $\lambda x, y, s.x(ys)$.
Exponentiation is represented by $\lambda x, y.yx$

Representable functions

Example 18.17

The Boolean values $\top$ and $\bot$ are represented as $\lambda x,y.y$ and $\lambda x,y.x$, respectively. Then 'if x then y else z' is represented by $\lambda x,y,z.xzy$.

$$
\begin{aligned}
&\text{if } \overline{\bot} \text{ then } A \text{ else } B\\
&\equiv (\lambda x,y,z.xzy)(\lambda x,y.x)AB\\
&=_\beta (\lambda y,z.(\lambda x,y.x)zy)AB\\
&=_\beta (\lambda y,z.z)AB =_\beta B \ ,
\end{aligned}
$$

$$
\begin{aligned}
&\text{if } \overline{\top} \text{ then } A \text{ else } B\\
&\equiv (\lambda x,y,z.xzy)(\lambda x,y.y)AB\\
&=_\beta (\lambda y,z.(\lambda x,y.y)zy)AB\\
&=_\beta (\lambda y,z.y)AB =_\beta A \ .
\end{aligned}
$$

Representable functions

Example 18.18

$$
\begin{aligned}
&1+1\\
&\equiv (\lambda x,y,s,z.xs(ysz))11\\
&\rhd_\beta (\lambda y,s,z.1s(ysz))1\\
&\rhd_\beta \lambda s,z.1s(1sz)\\
&\equiv \lambda s,z.(\lambda x,y.x(0xy))s(1sz)\\
&\rhd_\beta \lambda s,z.s(0s(1sz))\\
&\equiv \lambda s,z.s((\lambda x,y.y)s(1sz))\\
&\rhd_\beta \lambda s,z.s(1sz)\\
&\equiv 2
\end{aligned}
$$

Representable functions

To get a clue why these representations work we could read them as computations over logical structures. For example natural numbers are inductively defined from 0 and the successor. Hence a model for the naturals is specified when we provide a set together with a way to interpret 0 as some specific element and the successor as a function which transforms an element into another.

Consider $\overline{0} \equiv \lambda x,y.y$: this is a function from the model which provides an element of the model. The model is specified by providing the specification of the successor and the specification of zero. The result is the specification of 0.

Representable functions

Consider $\overline{n+1} \equiv \lambda x,y.x(\overline{n}xy)$: since $\overline{n}$ transforms a model into a number, the term $\overline{n}xy$ evaluates to n in the model (x,y). But x stands for the successor function so we are taking the successor of n in the model.

Thus, $x+y \equiv \lambda x,y,s,z.xs(ysz)$ is calculated by interpreting x in a model where the successor function is the given one but the zero element is ysz, i.e., the number which stands for y in the model.

Similarly, the product xy is calculated by interpreting x in a model where the successor function moves by y steps at once.

References

A classical, and still excellent introduction to λ-calculus is *J. Roger Hindley* and *Jonathan P. Seldin*, Lambda-Calculus and Combinators, Cambridge University Press, (2008).

The classical reference for the λ-calculus is *Henk P. Barendregt*, Lambda-Calculus: Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics 103, North Holland, (1985).

Mathematical Logic

Lecture 19

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Computability theory:

- The simple theory of types

Simple theory of types

The simple theory of types is, in essence, a λ-calculus with an extended syntax, in which terms are equipped with types.

The general idea is that functions must take arguments in their domain, and produce results in their codomain.

The types are deputed to model this behaviour, and to prevent the formation of terms which do not conform.

Hence, typed terms behave as λ-terms with respect to computation, but they are a subset of all the possible λ-terms, so we do not expect they capture the whole realm of computable functions.

Types

Definition 19.1 (Type)

Fixed a denumerable set V_T of *type variables*, a *type* is inductively defined:

- $x \in V_T$ is a type;
- 0 and 1 are types;
- if α and β are types, so are $(\alpha \times \beta)$, $(\alpha + \beta)$, and $(\alpha \to \beta)$.

As usual, we omit parentheses when they are not strictly needed: $\times$ binds stronger that $+$, and $+$ binds stronger than $\to$, so

$$\alpha \times \beta + \gamma \to (\alpha + \gamma) \times (\beta + \gamma)$$

stands for

$$((\alpha \times \beta) + \gamma) \to ((\alpha + \gamma) \times (\beta + \gamma)) \ .$$

A type is used to constrain the main entity of interest, the *term*.

Terms

Definition 19.2 (Term)

Fixed a family $\{V_\alpha\}_\alpha$ of *variables* indexed by the collection of types such that for each α, V_α is denumerable and distinct from the set of type variables, and such that $V_\alpha \cap V_\beta = \emptyset$ whenever $\alpha \neq \beta$, a *term* $t\colon \alpha$ of type α along with the set of its *free variables* is inductively defined as:

- if $x \in V_\alpha$ for some type α, $x\colon \alpha$ is a term and $\mathrm{FV}(x\colon \alpha) = \{x\colon \alpha\}$;
- $*\colon 1$ is a term and $\mathrm{FV}(*\colon 1) = \emptyset$;
- for each type α, $\Box_\alpha\colon 0 \to \alpha$ is a term and $FV(\Box_\alpha\colon 0 \to \alpha) = \emptyset$;
- if $A\colon \alpha$ and $B\colon \beta$ are terms, $\langle A, B\rangle\colon \alpha \times \beta$ is a term and $\mathrm{FV}(\langle A, B\rangle\colon \alpha \times \beta) = \mathrm{FV}(A\colon \alpha) \cup \mathrm{FV}(B\colon \beta)$;
- if $A\colon \alpha \times \beta$ is a term, so are $\pi_1 A\colon \alpha$ and $\pi_2 A\colon \beta$ and $\mathrm{FV}(\pi_1 A\colon \alpha) = \mathrm{FV}(\pi_2 A\colon \beta) = \mathrm{FV}(A\colon \alpha \times B)$; ↪

Terms

↪ (Term)

- if $A\colon \alpha$ is a term then for any type β, $i_1^\beta A\colon \alpha + \beta$ and $i_2^\beta A\colon \beta + \alpha$ are terms and $\mathrm{FV}(i_1^\beta A\colon \alpha + \beta) = \mathrm{FV}(i_2^\beta A\colon \beta + \alpha) = \mathrm{FV}(A\colon \alpha)$;
- if $C\colon \alpha + \beta$, $A\colon \alpha \to \gamma$, and $B\colon \beta \to \gamma$ are terms, so is $\delta(C, A, B)\colon \gamma$ and $\mathrm{FV}(\delta(C, A, B)\colon \gamma) = \mathrm{FV}(C\colon \alpha + \beta) \cup \mathrm{FV}(A\colon \alpha \to \gamma) \cup \mathrm{FV}(B\colon \beta \to \gamma)$;
- if $A\colon \beta$ is a term and $x \in V_\alpha$ then $\lambda x\colon \alpha.A\colon \alpha \to \beta$ is a term and $\mathrm{FV}(\lambda x\colon \alpha.A\colon \alpha \to \beta) = FV(A\colon \beta) \setminus \{x\colon \alpha\}$;
- if $A\colon \alpha$ and $B\colon \alpha \to \beta$ are terms then $B \cdot A\colon \beta$ is a term and $\mathrm{FV}(B \cdot A\colon \beta) = FV(A\colon \alpha) \cup \mathrm{FV}(B\colon \alpha \to \beta)$.

Terms represent the primitive computational statements.

Reductions

Terms can be *reduced* according to the following rules where it is assumed that both sides of the equalities are correctly typed:

- $\pi_1\langle A, B\rangle = A$; $\pi_2\langle A, B\rangle = B$;
- $(\lambda x\colon \alpha.A) \cdot B = A[B/x]$ the act of substituting B for x (β-reduction);
- $\lambda x\colon \alpha.(A \cdot x) = A$ when $x\colon \alpha \notin \mathrm{FV}(A\colon \alpha \to \beta)$ (η-reduction).

It is clear that these rules, which should be read as oriented from left to right, are computable.

Observe how equality $=$ is α-equivalence, and substitution is defined analogously to the pure λ-calculus.

Reductions

Moreover, the $\Box$ operator is subject to the following reductions:

- $(\Box_{\alpha\to\beta}A)\cdot B = \Box_\beta A$
- $\pi_1\Box_{\alpha\times\beta}A = \Box_\alpha A$; $\pi_2\Box_{\alpha\times\beta}B = \Box_\beta B$;
- $\delta(\Box_{\alpha+\beta}A, B, C) = \Box_\gamma A$;
- $\Box_\alpha(\Box_0 A) = \Box_\alpha A$.

Although these reductions seem obscure, their meaning will become transparent when interpreting the type system in logic.

Reductions

Finally, the δ operator is subject to the following reductions:

- $\delta(i_1 C, A, B) = A\cdot C$; $\delta(i_2 C, A, B) = B\cdot C$;
- $\pi_1\delta(p_1,p_2,p_3) = \delta(p1, \pi_1 p_2, \pi_1 p_3)$; $\pi_2\delta(p_1,p_2,p_3) = \delta(p1, \pi_2 p_2, \pi_2 p_3)$;
- $\Box_\gamma\delta(p_1,p_2,p_3) = \delta(p1, \Box_\gamma p_2, \Box_\gamma p_3)$;
- $\delta(p_1,p_2,p_3)\cdot p_4 = \delta(p_1, p_2\cdot p_4, p_3\cdot p_4)$;
- $\delta(\delta(p_1,p_2,p_3),p_4,p_5) = \delta(p_1,\delta(p_2,p_4,p_5),\delta(p_3,p_4,p_5))$.

Observe how most of these reductions are distribution laws.
Again, to fully understand these reductions, one needs to interpret the type system in logic.

Intended interpretation

The intended meaning of types is not too difficult to grasp:

- a type variable stands for a generic type.
- 0 stands for the empty type, and 1 stands for the type inhabited by a single, distinct element, the term $*$.
- the $\alpha\times\beta$ type stands for the Cartesian product of the α and β types; it is inhabited by the pairs $\langle A, B\rangle$ and its computational behaviour tells that the first (second) projection π_1 (π_2) yields the first (second) element in a pair, and pairing both the projections of an element yields the element itself.
- the $\alpha+\beta$ type stands for the disjoint union of α and β, and the $i_1^\beta A$, $i_2^\beta A$ terms are the injections of the A term in the disjoint union, on the left and on the right, respectively.
- the $\alpha\to\beta$ type stands for the function space having α as domain and β as codomain. Application is then function application, and abstraction is like in the pure λ-calculus.

Intended interpretation

In this respect, the δ operator is a selector for the disjoint union: given an element A in the disjoint union, it computes an element in the γ type by applying to A the second argument if A lies in the first component of the union, and applying to A the third argument if A lies in the second component of the union.

The β-reduction rules tells that given the description of a function, its application to some argument can be computed by substituting the argument inside the description.

The η-reduction, $\lambda x\colon\alpha.(Ax) = A$, is more complex, and ultimately says that functions are to be interpreted extensionally, that is, $f = g$ if and only if, $fx = gx$ for every x.

Church-Rosser theorem

Church-Rosser theorem holds in the simple theory of types without the + types essentially via the same proof as for the pure λ-calculus.

To prove it, one has to show that reductions are closed with respect to typing, that is, if $A \triangleright B$ as pure terms, and $A\colon \alpha$ then $B\colon \alpha$.

This property, called *subject reduction*, is fundamental and subtle. Indeed, in type theories more complex than the simple theory of types, it may fail in unexpected ways.

Enriching the system

We can easily derive in the simple theory of types a representation of the natural numbers together with the operations of addition, multiplication and exponentiation, the Boolean values, the if-then-else construction, and so on. Indeed, these representations are nothing but the same we used for the pure, non-typed λ-calculus.

Simple theory of types

If we restrict to the subsystem whose types are those generated by type variables, $\rightarrow$, and $\times$, and whose terms are correspondingly the variables, and those of the form $\lambda x\colon \alpha.A\colon \alpha \rightarrow \beta$, called *abstractions*, $A \cdot B\colon \beta$, called *applications*, $\langle A, B\rangle\colon \alpha \times \beta$, called *pairs*, $\pi_1 A\colon \alpha$ and $\pi_2 A\colon \beta$, called *projections*, we get a subsystem of special interest, which corresponds to the original typed system introduced by Alonzo Church.

References

A classical, and still excellent introduction to the simple theory of types is *J.Roger Hindley* and *Jonathan P. Seldin*, Lambda-Calculus and Combinators, Cambridge University Press, (2008).

The link between logical systems, their semantics, and the simple theory of types is illustrated in *Peter Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002).

The link between λ-calculus, continuous functions, and topological spaces is explained in the fundamental paper *Dana Scott*, Continuous lattices, in F.W. Lawvere ed., Toposes, Algebraic Geometry and Logic: Dalhousie University, Halifax, January 16–19, 1971, pp. 97–136, Springer (1972).

Mathematical Logic

Lecture 20

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Computability theory:

- The strong normalisation of the simple theory of types

The formal system

We consider the subsystem of the Simple Theory of Types whose types are

- type variables;
- function spaces $A \to B$;
- products $A \times B$.

and whose terms are variables, applications $f \cdot t$, abstractions $\lambda x : A.\, t$, pairs $\langle t_1, t_2 \rangle$, and projections $\pi_1 t$, $\pi_2 t$.

The one-step reduction $\rhd_1$ is the congruence generated by

- $\pi_1 \langle a, b \rangle \blacktriangleright a$;
- $\pi_2 \langle a, b \rangle \blacktriangleright b$;
- $(\lambda x : A.\, t) \cdot a \blacktriangleright t[a/x]$.

As usual $\rhd$ is the reflexive and transitive closure of $\rhd_1$.

Strong normalisation

Definition 20.1 (Strongly normalisable)

A term t is said to be *strongly normalisable* when no reduction sequence starting from t, $t \rhd t_1 \rhd \cdots \rhd t_n$ can be indefinitely extended.

In computational terms, it means that t eventually terminates.

Definition 20.2 (Neutral term)

A term t is said to be *neutral* when t is a variable, a projection $t \equiv \pi_1 t_1$, $t \equiv \pi_2 t_2$, or an application $t \equiv t_1 \cdot t_2$.

A term is said to be neutral when it does not interact with the context in which it may appear.

Strong normalisation

Definition 20.3 (Reducibility candidates)

The set of *reducibility candidates* of type T, denoted as $\mathscr{R}(T)$, is a set of terms of type T defined by induction on the type T:

- if T is a type variable, $t \in \mathscr{R}(T)$ if and only if t is strongly normalisable;
- if $T \equiv A \times B$, $t \in \mathscr{R}(T)$ if and only if $\pi_1 t \in \mathscr{R}(A)$ and $\pi_2 t \in \mathscr{R}(B)$;
- if $T \equiv A \to B$, $t \in \mathscr{R}(T)$ if and only if, for all $a \in \mathscr{R}(A)$, $t \cdot a \in \mathscr{R}(B)$.

The idea is to collect all the strongly normalisable terms of a given type, and then to show that they comprehend all the possible terms of that type.

Strong normalisation

Proposition 20.4

The following three properties hold for reducibility candidates:

1. *If $t \in \mathscr{R}(T)$ then t is strongly normalisable;*
2. *If $t \in \mathscr{R}(T)$ and $t \rhd t'$ then $t' \in \mathscr{R}(T)$;*
3. *If t is neutral and for all t' such that $t \rhd_1 t'$, $t' \in \mathscr{R}(T)$ then $t \in \mathscr{R}(T)$.*

Strong normalisation

Proof. (i)

By induction on the type T.

If T is a type variable then:

1. the result follows by definition of $\mathscr{R}(T)$.
2. since t is strongly normalisable every term to which t reduces has to be strongly normalisable and thus in $\mathscr{R}(T)$.
3. A proper reduction sequence starting from t must pass through some t' as for the hypothesis. By definition of $\mathscr{R}(T)$ all these terms t' are strongly normalisable and thus also t must be, which implies $t \in \mathscr{R}(T)$. ↪

Strong normalisation

↪ Proof. (ii)

If $T \equiv A \times B$ then:

1. if $t \in \mathscr{R}(A \times B)$ then $\pi_1 t \in \mathscr{R}(A)$, so $\pi_1 t$ is strongly normalisable by inductive hypothesis. Every reduction sequence from t maps into a reduction sequence from $\pi_1 t$ applying π_1 to every element, thus t has to be strongly normalisable.
2. If $t \rhd t'$ then $\pi_1 t \rhd \pi_1 t'$ and $\pi_2 t \rhd \pi_2 t'$. Since $t \in \mathscr{R}(A \times B)$ then $\pi_1 t \in \mathscr{R}(A)$ and $\pi_2 t \in \mathscr{R}(B)$ by definition of $\mathscr{R}$.
   Hence, by inductive hypothesis $\pi_1 t' \in \mathscr{R}(A)$ and $\pi_2 t' \in \mathscr{R}(B)$, thus $t' \in \mathscr{R}(A \times B)$ by definition of $\mathscr{R}$.
3. If $t \rhd_1 t'$ then $\pi_1 t \rhd \pi_1 t'$. Since $t' \in \mathscr{R}(A \times B)$, $\pi_1 t' \in \mathscr{R}(A)$ by definition of $\mathscr{R}$. Also, since t is neutral, in particular not a pair, $\pi_1 t \rhd_1 \pi_1 t'$ in one step. Since $\pi_1 t$ is neutral, by inductive hypothesis $\pi_1 t \in \mathscr{R}(A)$.
   Symmetrically, one proves $\pi_2 t \in \mathscr{R}(B)$, so $t \in \mathscr{R}(A \times B)$ by definition. ↪

Strong normalisation

↪ Proof. (iii)

If $T \equiv A \to B$ then:

1. Let $x : A$ be a variable: x is irreducible and neutral so by inductive hypothesis on A, $x \in \mathscr{R}(A)$. Then $t \cdot x \in \mathscr{R}(B)$ since $t \in \mathscr{R}(A \to B)$. By inductive hypothesis on B, $t \cdot x$ is strongly normalisable. Hence, every reduction sequence starting from t can be mapped in a reduction sequence starting from $t \cdot x$ applying x to every element. Thus, the sequence from t cannot be indefinitely extended and so t is strongly normalisable.
2. Let $a \in \mathscr{R}(A)$. Since $t \in \mathscr{R}(A \to B)$, $t \cdot a \in \mathscr{R}(B)$ by definition of $\mathscr{R}$. Also $t \rhd_1 t'$ maps to $t \cdot a \rhd t' \cdot a$. By inductive hypothesis on B, $t' \cdot a \in \mathscr{R}(B)$, thus $t' \in \mathscr{R}(A \to B)$ by definition. ↪

Strong normalisation

↪ Proof. (iv)

If $T \equiv A \to B$ then:

3. Let $a \in \mathscr{R}(A)$. Then a is strongly normalisable by inductive hypothesis. By induction on the reduction sequences from a to show that if $t \cdot a \rhd_1 r$ then $r \in \mathscr{R}(B)$:
   - if $t \cdot a \rhd_1 t' \cdot a$ because $t \rhd_1 t'$ then $t' \in \mathscr{R}(A \to B)$ by hypothesis so $t' \cdot a \in \mathscr{R}(B)$ by definition of $\mathscr{R}$.
   - if $t \cdot a \rhd_1 t \cdot a'$ because $a \rhd_1 a'$ then $a' \in \mathscr{R}(A)$ by the main induction hypothesis (2) on A, so the secondary induction hypothesis on a' tells $t \cdot a' \in \mathscr{R}(B)$.
   - $t \cdot a$ does not reduce as a whole since t is neutral by hypothesis, in particular t is not an abstraction, so we already considered all the possible cases.

   Hence $t \cdot a \in \mathscr{R}(B)$ by induction hypothesis (3) on B.
   Thus by definition of $\mathscr{R}$, $t \in \mathscr{R}(A \to B)$. □

Strong normalisation

Proposition 20.5

If $a \in \mathscr{R}(A)$ and $b \in \mathscr{R}(B)$ then $\langle a, b\rangle \in \mathscr{R}(A \times B)$.

Proof.

By induction on the reduction sequences starting from a or b to show that if $\pi_1\langle a, b\rangle \rhd_1 t$ then $t \in \mathscr{R}(A)$:

- if $\pi_1\langle a, b\rangle \rhd_1 a$, $a \in \mathscr{R}(A)$ by hypothesis;
- if $\pi_1\langle a, b\rangle \rhd_1 \pi_1\langle a', b\rangle$ because $a \rhd_1 a'$ then $a' \in \mathscr{R}(A)$ by point 2 in Proposition 20.4, so $\pi_1\langle a', b\rangle \in \mathscr{R}(A)$ by inductive hypothesis;
- if $\pi_1\langle a, b\rangle \rhd_1 \pi_1\langle a, b'\rangle$ because $b \rhd_1 b'$ then $b' \in \mathscr{R}(B)$ by point 2 in Proposition 20.4, so $\pi_1\langle a, b'\rangle \in \mathscr{R}(A)$ by inductive hypothesis.

Since $\pi_1\langle a, b\rangle$ is neutral, by point 3 in Proposition 20.4 $\pi_1\langle a, b\rangle \in \mathscr{R}(A)$.
Symmetrically, one shows that $\pi_2\langle a, b\rangle \in \mathscr{R}(B)$.
Hence by definition of $\mathscr{R}$, $\langle a, b\rangle \in \mathscr{R}(A \times B)$. □

Strong normalisation

Proposition 20.6

If for all $a \in \mathscr{R}(A)$, $b[a/x] \in \mathscr{R}(B)$ then $\lambda x : A.\, b \in \mathscr{R}(A \to B)$.

Proof.

By point 3 of Proposition 20.4 $x \in \mathscr{R}(A)$, so $b[x/x] \equiv b \in \mathscr{R}(B)$ by hypothesis.
Fix $a \in \mathscr{R}(A)$. By induction on the reduction sequences starting from a or b to show that if $(\lambda x : A.\, b) \cdot a \rhd_1 t$ then $t \in \mathscr{R}(B)$:

- if $(\lambda x : A.\, b) \cdot a \rhd_1 b[a/x]$ then $b[a/x] \in \mathscr{R}(B)$ by hypothesis;
- if $(\lambda x : A.\, b) \cdot a \rhd_1 (\lambda x : A.\, b') \cdot a$ because $b \rhd_1 b'$ then $b' \in \mathscr{R}(B)$ by point 2 of Proposition 20.4, so $(\lambda x : A.\, b') \cdot a \in \mathscr{R}(B)$ by inductive hypothesis;
- if $(\lambda x : A.\, b) \cdot a \rhd_1 (\lambda x : A.\, b) \cdot a'$ because $a \rhd_1 a'$ then $a' \in \mathscr{R}(A)$ by point 2 of Proposition 20.4, so $(\lambda x : A.\, b) \cdot a' \in \mathscr{R}(B)$ by inductive hypothesis.

Since $(\lambda x : A.\, b) \cdot a$ is neutral, by point 3 of Proposition 20.4
$(\lambda x : A.\, b) \cdot a \in \mathscr{R}(B)$, so $\lambda x : A.\, b \in \mathscr{R}(A \to B)$ by definition of $\mathscr{R}$. □

Strong normalisation

Proposition 20.7

Let t be a term of type T and let $\mathrm{FV}(t) \subseteq \{x_1 : A_1, \ldots, x_n : A_n\}$. Let $a_i \in \mathscr{R}(A_i)$ for all $1 \leq i \leq n$. Then $t[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T)$.

Proof. (i)

By induction on the term t

- if $t \equiv x_i$ for some $1 \leq i \leq n$ then $T \equiv A_i$ and $t[a_1/x_1, \ldots, a_n/x_n] \equiv a_i \in \mathscr{R}(A_i)$ by hypothesis.
- if $t \equiv \pi_1 t'$ then $t'[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T \times B)$ by inductive hypothesis. So $t[a_1/x_1, \ldots, a_n/x_n] \equiv \pi_1 t'[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T)$ by definition of $\mathscr{R}$.
- if $t \equiv \pi_2 t'$ then $t'[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(B \times T)$ by inductive hypothesis. So $t[a_1/x_1, \ldots, a_n/x_n] \equiv \pi_2 t'[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T)$ by definition of $\mathscr{R}$.
- if $t \equiv \langle t_1, t_2 \rangle$ then $T \equiv T_1 \times T_2$ and by inductive hypothesis $t_1[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T_1)$ and $t_2[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T_2)$. Then $t[a_1/x_1, \ldots, a_n/x_n] \equiv \langle t_1[a_1/x_1, \ldots, a_n/x_n], t_2[a_1/x_1, \ldots, a_n/x_n] \rangle \in \mathscr{R}(T_1 \times T_2)$ by Proposition 20.5.

↪

Strong normalisation

↪ **Proof. (ii)**

- if $t \equiv t_1 \cdot t_2$ then $t_1[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(B \to T)$ and $t_2[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(B)$ by inductive hypothesis. Hence $t[a_1/x_1, \ldots, a_n/x_n] \equiv t_1[a_1/x_1, \ldots, a_n/x_n] \cdot t_2[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(T)$ by definition of $\mathscr{R}$.
- if $t \equiv \lambda y : B.\, t_1$ then $T \equiv B \to T_1$ and for all $b \in \mathscr{R}(B)$, $t_1[a_1/x_1, \ldots, a_n/x_n, b/y] \in \mathscr{R}(T_1)$ by inductive hypothesis. Hence $t[a_1/x_1, \ldots, a_n/x_n] \equiv \lambda y : B.\, t_1[a_1/x_1, \ldots, a_n/x_n] \in \mathscr{R}(B \to T_1)$ by Proposition 20.6. □

Strong normalisation

Theorem 20.8

For every term t of type T, $t \in \mathscr{R}(T)$.

Proof.

Let $\mathrm{FV}(t) \subseteq \{x_1 : A_1, \ldots, x_n : A_n\}$. By point 3 of Proposition 20.4 being x_i irreducible and neutral, $x_i \in \mathscr{R}(A_i)$.
Hence by Proposition 20.7, $t[x_1/x_1, \ldots, x_n/x_n] \equiv t \in \mathscr{R}(T)$. □

Theorem 20.9 (Strong normalisation)

Every term is strongly normalisable.

Proof.

Let t be a term of type T. By Theorem 20.8 $t \in \mathscr{R}(T)$ so t is strongly normalisable by point 1 in Proposition 20.4. □

Discussion

The strong normalisation property is extremely powerful: it tells that every element in the class of computable functions which can be represented in the simple theory of types is a total function.

In more complex type theories this result is critical, and often invalid. Also, it has deep consequences in logic, which will be remarked in due time, after introducing the so-called Curry-Howard isomorphism.

References

The proof of the Strong Normalisation Theorem can be found in *Jean-Yves Girard, Yves Lafont, Paul Taylor* Proofs and Types, Cambridge University Press (1989).

The proving technique is due to William Tait and refined by Jean-Yves Girard.

Mathematical Logic

Lecture 21

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Computability theory:

- Dependent types

Dependent types

The simple theory of types is a nice theory with many good properties, like strong normalisation. But it fails to model complex mathematics. The situation is similar to propositional logic which have to be extended to first-order logic to cope with real mathematical theories. Indeed, as we will see in the following, the comparison with propositional logic is quite precise.

There are many ways to extend the simple theory of types. However, one way, introducing *dependent types*, proved to be extremely useful and with deep consequences, both practical and theoretical.

In this lecture we want to sketch the fundamentals of dependent types, to grasp the main idea but leaving out most of the theory, which is far more complex than the one of simple types, and still researched.

Dependent types

The basic idea is to define types and terms together, by means of a set of inference rules. Along with the presentation of the inference rules, we describe and comment the related concepts.

The conclusion and the premises of each inference rule are *judgements*:

- a *context* judgement has the form $\Gamma\,\mathrm{ctx}$, where Γ is a *context*;
- a *regular* judgement, or simply a judgement, has the form $\Gamma \vdash t : T$ where t is a *term* and T is a *type*;
- an *equivalence* judgement, or simply an equivalence, has the form $\Gamma \vdash t_1 \equiv t_2 : T$ where t_1, t_2 are terms and T is a *type*.

Therefore, the terms and types are those expressions which are generated as conclusions of valid derivations formed by the inference rules shown in the following slides.

Contexts

First, we introduce *contexts*: a context is a finite list of distinct typed variables. As usual, we assume to have a predefined infinite and recursive set of variables. We write $A : \mathscr{U}_i$ with $i \in \mathbb{N}$ to mean that the expression A is a type (this will be refined shortly).

Then, the inference rules governing context judgements are

$$\frac{}{\bullet\,\mathrm{ctx}}\ \mathrm{ctx\text{–}EMP} \qquad \frac{\Gamma \vdash A : \mathscr{U}_i}{\Gamma, x : A\,\mathrm{ctx}}\ \mathrm{ctx\text{–}EXT}$$

that is, the empty context is a valid context, and if A is a type in Γ, then $\Gamma, x : A$ is a valid context **provided** the variable x does not appear in Γ.

The fundamental inference on a context allows to derive a typed variable:

$$\frac{x_1 : A_1, \ldots, x_n : A_n\,\mathrm{ctx}}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}\ \mathrm{Vble}$$

with $1 \leq i \leq n$.

Judgemental equivalence

Equivalence judgements are governed by a number of rules. Some of them are specific for each type constructor, and they will be introduced later. First of all, we want to say that $\equiv$ is an equivalence relation between terms:

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \equiv a : A}\ \equiv\text{–refl} \qquad \frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash b \equiv a : A}\ \equiv\text{–sym} \qquad \frac{\Gamma \vdash a \equiv b : A \quad \Gamma \vdash b \equiv c : A}{\Gamma \vdash a \equiv c : A}\ \equiv\text{–trans}$$

and substituting equivalent types is licit:

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash A \equiv B : \mathscr{U}_i}{\Gamma \vdash a : B}\ \equiv\text{–subst} \qquad \frac{\Gamma \vdash a \equiv b : A \quad \Gamma \vdash A \equiv B : \mathscr{U}_i}{\Gamma \vdash a \equiv b : B}\ \equiv\text{–subst–eq}$$

Equivalence, or *judgemental equivalence*, wants to model the fact that two terms are equivalent with respect to the reduction rules, similarly to β-equality. However, the reduction rules are more complex, and they will be specified as inference rules.

Universes

An important feature of dependent types, is that there is no strict distinction between terms and types: every type is also a term. However, types as terms have a *universe* as a type.

Universes are organised in a cumulative hierarchy indexed by natural numbers, thus $\mathscr{U}_i$ is a term in the type $\mathscr{U}_{i+1}$. Also, if A is type in the i-th universe, it is so in every universe above i.

$$\frac{\Gamma\,\mathrm{ctx}}{\Gamma \vdash \mathscr{U}_i : \mathscr{U}_{i+1}}\ \mathscr{U}\text{–intro}$$

$$\frac{\Gamma \vdash A : \mathscr{U}_i}{\Gamma \vdash A : \mathscr{U}_{i+1}}\ \mathscr{U}\text{–cumul} \qquad \frac{\Gamma \vdash A \equiv B : \mathscr{U}_i}{\Gamma \vdash A \equiv B : \mathscr{U}_{i+1}}\ \mathscr{U}\text{–cumul–eq}$$

Universes

It is important to remark that there is no maximal universe.
In fact, it could be proved that the existence of a maximal universe induces the Burali-Forti paradox.

Hence, it makes sense to not have universes at all, or to have an infinite amount of them, indexed by a limit ordinal. We consider the only possibility for the second case which provides an effective theory.

The system with no universes is interesting, anyway. Indeed, one can present the simple theory of types in it. And, even with dependent types, it enjoys some properties, like normalisation, which are difficult, if not impossible, to show in the system with universes.

This fact also makes evident that adding universes really makes the type system more powerful, essentially a real higher-order system.

Dependent function spaces

Dependent function spaces, sometimes called dependent products, are types describing the dependent functions from A to B.

A dependent function f from A to B, notation $f : \Pi x : A.B$, maps an element a of A in an element $f\,a$ of $B[a/x]$.

For example, an n-vector of reals can be represented as a list of reals whose length is n. Then, the 0 vector is a dependent function, taking n as an argument, and yielding the list of length n whose elements are all 0.

Indeed, to show significant examples, one should say that Π can be interpreted as the $\forall$ quantifier of x of type A, and the type B, depending on x, is a formula describing the elements of the target type.

When B does not depend on x, i.e., when $x \notin \mathrm{FV}(B)$, we get the usual function space $A \to B$ as in the simple theory of types.

Dependent function spaces

The rules to form a dependent function space are the following:

$$\frac{\Gamma \vdash A : \mathscr{U}_i \quad \Gamma, x : A \vdash B : \mathscr{U}_i}{\Gamma \vdash \Pi x : A.B : \mathscr{U}_i}\ \Pi\text{–form}$$

$$\frac{\Gamma \vdash A \equiv A' : \mathscr{U}_i \quad \Gamma, x : A \vdash B \equiv B' : \mathscr{U}_i}{\Gamma \vdash \Pi x : A.B \equiv \Pi x : A'.B' : \mathscr{U}_i}\ \Pi\text{–form–eq}$$

Dependent function spaces

The rules to construct a dependent function, using the λ constructor, are

$$\frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x : A.b : \Pi x : A.B}\ \Pi\text{–intro}$$

$$\frac{\Gamma, x : A \vdash b \equiv b' : B \quad \Gamma \vdash A \equiv A' : \mathscr{U}_i}{\Gamma \vdash \lambda x : A.b \equiv \lambda x : A'.b' : \Pi x : A.B}\ \Pi\text{–intro–eq}$$

The equivalence introduced in the $-$eq rule says that $\equiv$ is a congruence with respect to abstraction.

Dependent function spaces

The rules to apply a dependent function, using the $\cdot$ operator, are:

$$\frac{\Gamma \vdash f : \Pi x : A.\, B \quad \Gamma \vdash a : A}{\Gamma \vdash f\, a : B[a/x]}\ \Pi\text{–elim}$$

$$\frac{\Gamma \vdash f \equiv g : \Pi x : A.\, B \quad \Gamma \vdash a \equiv b : A}{\Gamma \vdash f\, a \equiv g\, b : B[a/x]}\ \Pi\text{–elim–eq}$$

Dependent function spaces

The reductions associated to dependent function are β and η, captured by the following inference rules:

$$\frac{\Gamma, x : A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x : A.\, b)\, a \equiv b[a/x] : B[a/x]}\ \Pi\text{–comp}$$

$$\frac{\Gamma \vdash f : \Pi x : A.\, B}{\Gamma \vdash \lambda x : A.\, f\, x \equiv f : \Pi x : A.\, B}\ \Pi\text{–uniq}$$

It is worth observing that, when $x \notin \mathrm{FV}(B)$, i.e., when the function space is not dependent, these rules coincide with the β and η reductions of the simple theory of types about the $A \to B$ type.

Dependent pair types

A dependent pair type, written as $\Sigma x : A.\, B$, also called dependent sum, is a type whose extension could be described as the set of pairs (a, b) such that $a : A$ and $b : B[a/x]$.

As an example, in Computer Science, consider a record in a data base describing employees at the university: if the employee is a professor, their record will contain a list of taught courses; if the employee is a lab technician, their record will contain a list of the served laboratories.

Indeed, many examples can be easily shown when considering that Σ can be interpreted as the $\exists$ quantifier of x of type A of a formula B depending on x.

When B does not depend on x, i.e., when $x \notin \mathrm{FV}(B)$, we get the usual Cartesian product $A \times B$ as in the simple theory of types.

Dependent pair types

The rule to form a dependent pair type is

$$\frac{\Gamma \vdash A : \mathscr{U}_i \quad \Gamma, x : A \vdash B : \mathscr{U}_i}{\Gamma \vdash \Sigma x : A.\, B : \mathscr{U}_i}\ \Sigma\text{–form}$$

And the rule to construct a dependent pair is

$$\frac{\Gamma \vdash \Sigma x : A.\, B : \mathscr{U}_i \quad \Gamma \vdash a : A \quad \Gamma, x : A \vdash b : B}{\Gamma \vdash (a, b) : \Sigma x : A.\, B}\ \Sigma\text{–intro}$$

Dependent pair types

The elimination rule for dependent pairs codes induction:

$$\frac{\Gamma \vdash \Sigma x : A.\, B : \mathcal{U}_i}{\begin{array}{l}\Gamma \vdash \mathrm{ind}_{\Sigma x:A.B} : \Pi C : (\Sigma x : A.\, B) \to \mathcal{U}_i, \\ \qquad g : \Pi x : A.\, \Pi y : B.\, C\,(x,y), \\ \qquad p : (\Sigma x : A.\, B).\, C\,p\end{array}}\ \Sigma\text{−elim}$$

It says that, given a formula C depending on a dependent pair, given a proof g mapping x and y to $C(x,y)$, and given a point p which is a dependent pair, the induction tells that the property C holds on p.

Dependent pair types

The reduction associated to dependent pairs is

$$\frac{\begin{array}{ll}\Gamma \vdash \Sigma x : A.\, B : \mathcal{U}_i & \Gamma \vdash C : (\Sigma x : A.\, B) \to \mathcal{U}_i \\ \Gamma \vdash (a,b) : \Sigma x : A.\, B & \Gamma \vdash g : \Pi x : A.\, \Pi y : B.\, C\,(x,y)\end{array}}{\Gamma \vdash \mathrm{ind}_{\Sigma x:A.B}\, C\, g\,(a,b) \equiv g\, a\, b : C\,(a,b)}\ \Sigma\text{−comp}$$

that is, computing induction on a pair, yields the property on that pair as established by g.

More types

In a similar vein, more types can be defined, each one equipped with a formation rule, a number of introduction rules, one for each constructor, an elimination rule stating the induction principle, and a number of computation rules, showing how induction reduces when applied to an instance of a constructor. Indeed, even a generic syntax to model types can be developed.

In particular, the unit type (1), the empty type (0), the coproduct type $A + B$ of the simple theory of types can be easily defined.

Interestingly, an important type can be defined: equality $a =_A b$, with $a, b : A$. Its elements are the ways to show that equality between a and b holds.

Dependent type theories

The type theory illustrated so far is one of the many variants of Martin-Löf type theory. This theory has been used to define many contemporary functional programming languages, e.g. Haskell, and it has many deep and not yet completely understood properties.

However, identity types, i.e. $a =_A b$, can be interpreted as the space of paths from a to b in a topological space. Together with a subtle axiom, *univalence*, the theory of dependent types can be then interpreted in homotopy spaces and shown to be a very good and deep description of them.
This theory is known as *homotopy type theory*, which has been under deep research in the last years.

References

Homotopy type theory has been introduced in *The Univalent Foundation Program*, Homotopy Type Theory: Univalent Foundations of Mathematics, Institute for Advanced Studies, Princeton.

The core of dependent type theory, as described in this lecture, has been derived from Chapter 1 of that book.

Mathematical Logic

Lecture 22

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Constructive mathematics:

- Motivation
- Intuitionistic logic
- Syntax
- Expressive power

Motivation

Consider the following

Proposition 22.1

There are a and b irrational numbers such that a^b is rational.

Proof.

Let $a = b = \sqrt{2}$. Then $a^b = \sqrt{2}^{\sqrt{2}}$ is either rational or irrational. In the former case the statement is proved, otherwise take $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$. Then $a^b = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \sqrt{2}^2 = 2$. □

This proof is correct but still unsatisfactory: at the end we don't know a pair of irrationals with the stated property. We have a choice between two candidate pairs but no way to decide which pair satisfies our requirement.

Motivation

On the contrary the following proof is different:

Proof.
Let $a = \sqrt{2}$ and $b = \log_2 9$. It is well known that a is irrational but also b is. Indeed, if $\log_2 9 = m/n$ for some $m, n \in \mathbb{N}$ then by the properties of logarithms, $2^m = 9^n$, which is impossible since the left-hand of the equality is even while the right-hand is odd. But $a^b = \sqrt{2}^{\log_2 9} = 2^{(\log_2 9)/2} = 2^{\log_2 3} = 3$. □

Here the statement says that there are two irrationals a and b such that a^b is rational and the proof provides an evidence for this exhibiting such a pair.

Motivation

In general, we would like that any time we have to prove a statement of the form $A \vee B$ or $\exists x.P$, we are able to indicate which disjunct holds between A and B or a value for x. And we would like that these pieces of information lie in the proof.

More precisely we would like to say that a proof for statements of this form would consist of an algorithm that indicate the true disjunct or *constructs* a value for x.

This attitude is perfectly reasonable but comes with a price: we cannot use anymore axioms that directly violate the requirement. Indeed, the Law of Excluded Middle says that $A \vee \neg A$ for any formula A, but it provides no way to decide which of these mutually exclusive facts holds.

Motivation

But rejecting the Law of Excluded Middle is not sufficient. There are a number of principles which pose problems.

For example the Axiom of Choice. In one of its consequences, the already cited Tarski-Banach theorem, we can cut a sphere into a finite number of pieces so that we can reassembly two spheres identical to the original one. The proof 'constructs' the pieces using the Axiom of Choice. But any non-mathematician would call that result a miracle unless you show **how to** cut the original sphere and **how to** reassemble the pieces! And any mathematician would note that the proof does not provide an **effective** way to calculate the shape of the pieces.

Motivation

Indeed, we are interested in systems where proofs are a sort of algorithm to construct the results implicit in their statements.

This attitude toward Mathematics is called *constructivism* and it produced a different kind of logical systems. In these systems, principles like the Law of Excluded Middle are rejected or accepted on the basis that they permit or deny the possibility to 'construct' the objects their statement imply to exists or the possibility to make the choices required in the proofs.

There are many constructive systems and many variations on the theme. Different philosophical foundations have been proposed to support the constructive approaches, and there are degrees of constructiveness in the logical system which claim themselves to adhere to these approaches.

An indisputable fact is that constructive mathematics had, have, and probably will have a deep impact in the study of computability.

Intuitionistic logic

Among the many constructive system, *intuitionistic logic* has a special place. Historically it has been the first attempt to capture in a formal system the original idea of a constructive approach to Mathematics. Practically, it is the simplest, most studied, and best understood system in this line of thought.

In the following we will introduce intuitionistic first-order logic showing some of its main features. Differently from the study we pursued of classical systems, we will not prove every result and we will easily skip over some important parts: the field of constructive mathematics is wide, deep, and complex, and our objective is to show how and why a non-classical system could be of interest.

Syntax

Syntactically, intuitionistic logic is very similar to classical logic. In the propositional case formulæ are formed in exactly the same way. In the first-order case terms and formulæ are constructed identically.

The difference lie in the construction of proofs: the valid intuitionistic proofs are the classical proofs in natural deductions where the Law of Excluded Middle does not appear. In other words, the propositional calculus and the first-order calculus are identical to the corresponding classical calculi except that the Law of Excluded Middle is dropped.

Expressive power

We may think that intuitionistic logic is less expressive than classical logic: possibly there are statements which are provable in the classical system, but cannot be proved in the intuitionistic system because they use the Law of Excluded Middle in an essential way. On the contrary, every result which can be proved in an intuitionistic system is also valid in the corresponding classical system because each intuitionistic proof is also a classical derivation where there is no application of the Law of Excluded Middle.

In a sense the above remark is correct. But in another sense it is not...

Expressive power

...since the ability to prove more having an additional inference rule, may lead to prove more theories to be non consistent.

For example, Church Thesis in computability theory says that a function $\mathbb{N} \to \mathbb{N}$ is computable if and only if there is a Turing machine computing it. If we say that every function we can write in arithmetic is computable, we get the so-called formal Church Thesis. It turns out that the formal theory of arithmetic plus formal Church thesis is a perfectly reasonable intuitionistic theory, which can be proved to be consistent with respect to (classical) arithmetic. On the contrary, the very same theory in classical logic turns out to be contradictory.

The reason is simple: in classical logic it is possible to prove that a non computable function exists by contradiction. So the formal Church thesis, which asserts that every function is computable, leads to a contradiction. In intuitionistic logic the proof of that function to be not computable cannot be carried on.

Expressive power

It is important to remark the deep mathematical meaning of the example: we would like to write a theory of computable functions. However, this is impossible in classical logic, unless we accept to describe a wider class of functions.

We really want to study what happens when we limit to consider computable functions only. In the end, it is the mathematical theory of Computer Science!

Expressive power

From another point of view, every theorem in classical logic can be proved in intuitionistic logic modulo a translation. The precise statement is as follows:

Definition 22.2

The *Gödel-Gentzen translation* is a map of formulæ to formulæ inductively defined as:

- $(\top)^N = \top$, $(\bot)^N = \bot$;
- for any A atomic $(A)^N = \neg\neg A$;
- $(A \wedge B)^N = (A)^N \wedge (B)^N$;
- $(A \vee B)^N = \neg\left(\neg (A)^N \wedge \neg (B)^N\right)$;
- $(A \supset B)^N = (A)^N \supset (B)^N$;
- $(\forall x\colon s.A)^N = \forall x\colon s.(A)^N$;
- $(\exists x\colon s.A)^N = \neg\forall x\colon s.\neg (A)^N$.

Expressive power

Proposition 22.3

In classical logic, for any formula A there is a pair of proofs $\pi_1\colon \vdash A \supset (A)^N$ and $\pi_2\colon \vdash (A)^N \supset A$.

Proof. (i)

By induction on the formula A:

- $A \equiv \bot, \top$: thus $(\bot)^N = \bot$ and $(\top)^N = \top$, so $\vdash \bot \supset \bot$ and $\vdash \top \supset \top$ by $\supset$I.
- A is atomic: hence $(A)^N = \neg\neg A$ and

$$\dfrac{\dfrac{\dfrac{[A]^1 \quad [\neg A]^2}{\bot}\,\neg\mathrm{E}}{\neg\neg A}\,\neg\mathrm{I}^2}{A \supset \neg\neg A}\,\supset\mathrm{I}^1 \qquad \dfrac{\dfrac{}{A \vee \neg A}\,\mathrm{lem} \quad \dfrac{\dfrac{[\neg\neg A]^2 \quad [\neg A]^1}{\bot}\,\neg\mathrm{E}}{A}\,\bot\mathrm{E} \quad [A]^1}{A}\,\vee\mathrm{E}^1}{\neg\neg A \supset A}\,\supset\mathrm{I}^2$$

↪

Expressive power

↪ Proof. (ii)

- $A \equiv B \wedge C$: by induction hypothesis there are $\vdash B \supset (B)^N$, $\vdash (B)^N \supset B$, $\vdash C \supset (C)^N$, $\vdash (C)^N \supset C$, and $(A)^N = (B)^N \wedge (C)^N$ so

$$\dfrac{\dfrac{\dfrac{[B \wedge C]^1}{B}\,\wedge\mathrm{E}_1 \quad \dfrac{[B \wedge C]^1}{C}\,\wedge\mathrm{E}_2}{(B)^N \quad (C)^N}\,\wedge\mathrm{I}}{(B)^N \wedge (C)^N}\,\supset\mathrm{I}^1}{B \wedge C \supset (B)^N \wedge (C)^N} \qquad \dfrac{\dfrac{\dfrac{[(B)^N \wedge (C)^N]^1}{(B)^N}\,\wedge\mathrm{E}_1 \quad \dfrac{[(B)^N \wedge (C)^N]^1}{(C)^N}\,\wedge\mathrm{E}_2}{B \quad C}\,\wedge\mathrm{I}}{B \wedge C}\,\supset\mathrm{I}^1}{(B)^N \wedge (C)^N \supset B \wedge C}$$

↪

Expressive power

↪ Proof. (iii)

- $A \equiv B \lor C$: by induction hypothesis there are $\vdash B \supset (B)^N$, $\vdash (B)^N \supset B$, $\vdash C \supset (C)^N$, $\vdash (C)^N \supset C$, and $(A)^N = \neg(\neg(B)^N \land \neg(C)^N)$ so

$$\dfrac{\dfrac{\dfrac{[B \lor C]^1 \quad \dfrac{\begin{matrix}[B]^2\\ \vdots\\ (B)^N\end{matrix} \quad \dfrac{[\neg(B)^N \land \neg(C)^N]^3}{\neg(B)^N}\land E_1}{\bot}\neg E \quad \dfrac{\begin{matrix}[C]^2\\ \vdots\\ (C)^N\end{matrix} \quad \dfrac{[\neg(B)^N \land \neg(C)^N]^3}{\neg(C)^N}\land E_2}{\bot}\neg E}{\bot}\lor E^2}{\neg\left(\neg(B)^N \land \neg(C)^N\right)}\neg I^3}{B \lor C \supset \neg\left(\neg(B)^N \land \neg(C)^N\right)}\supset I^1$$

Expressive power

↪ Proof. (iv)

$$\dfrac{\dfrac{B \lor \neg B}{}\text{lem} \quad \dfrac{[B]^1}{B \lor C}\lor I_1 \quad \dfrac{\dfrac{C \lor \neg C}{}\text{lem} \quad \dfrac{[C]^2}{B \lor C}\lor I_2 \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg B]^1 \quad \begin{matrix}[(B)^N]^3\\ \vdots\\ B\end{matrix}}{\bot}\neg E}{\neg(B)^N}\neg I^3 \quad \dfrac{\dfrac{[\neg C]^2 \quad \begin{matrix}[(C)^N]^4\\ \vdots\\ C\end{matrix}}{\bot}\neg E}{\neg(C)^N}\neg I^4}{\neg(B)^N \land \neg(C)^N}\land I \quad [\neg(\neg(B)^N \land \neg(C)^N)]^5}{\bot}\neg E}{B \lor C}\bot E}{B \lor C}\lor E^2}{\dfrac{B \lor C}{\neg\left(\neg(B)^N \land \neg(C)^N\right) \supset B \lor C}\supset I^5}\lor E^1$$

Expressive power

↪ Proof. (v)

- $A \equiv B \supset C$: by induction hypothesis there are $\vdash B \supset (B)^N$, $\vdash (B)^N \supset B$, $\vdash C \supset (C)^N$, $\vdash (C)^N \supset C$, and $(A)^N = (B)^N \supset (C)^N$ so

$$\dfrac{\dfrac{\begin{matrix}\dfrac{[B \supset C]^1 \quad \begin{matrix}[(B)^N]^2\\ \vdots\\ B\end{matrix}}{C}\supset E\\ \vdots\\ (C)^N\end{matrix}}{(B)^N \supset (C)^N}\supset I^2}{(B \supset C) \supset \left((B)^N \supset (C)^N\right)}\supset I^1 \qquad \dfrac{\dfrac{\begin{matrix}\dfrac{[(B)^N \supset (C)^N]^1 \quad \begin{matrix}[B]^2\\ \vdots\\ (B)^N\end{matrix}}{(C)^N}\supset E\\ \vdots\\ C\end{matrix}}{B \supset C}\supset I^2}{\left((B)^N \supset (C)^N\right) \supset (B \supset C)}\supset I^1$$

Expressive power

↪ Proof. (vi)

- $A \equiv \neg B$: by induction hypothesis there are $\vdash B \supset (B)^N$, $\vdash (B)^N \supset B$, and $(A)^N = \neg(B)^N$ so

$$\dfrac{\dfrac{\dfrac{[\neg B]^1 \quad \begin{matrix}[(B)^N]^2\\ \vdots\\ B\end{matrix}}{\bot}\neg E}{\neg(B)^N}\neg I^2}{\neg B \supset \neg(B)^N}\supset I^1 \qquad \dfrac{\dfrac{\dfrac{[\neg(B)^N]^1 \quad \begin{matrix}[B]^2\\ \vdots\\ (B)^N\end{matrix}}{\bot}\neg E}{\neg B}\neg I^2}{\neg(B)^N \supset \neg B}\supset I^1$$

Expressive power

↪ Proof. (vii)

- $A \equiv \forall x.B$: by induction hypothesis there are $\vdash B \supset (B)^N$, $\vdash (B)^N \supset B$, and $(A)^N \equiv \forall x.(B)^N$ so

$$\dfrac{\dfrac{[\forall x.B]^1}{B}\,\forall \mathsf{E}}{\vdots}\\ \dfrac{(B)^N}{\forall x.(B)^N}\,\forall \mathsf{I}}{(\forall x.B) \supset \left(\forall x.(B)^N\right)}\,\supset \mathsf{I}^1 \qquad \dfrac{\dfrac{[\forall x.(B)^N]^1}{(B)^N}\,\forall \mathsf{E}}{\vdots}\\ \dfrac{B}{\forall x.B}\,\forall \mathsf{I}}{\left(\forall x.(B)^N\right) \supset (\forall x.B)}\,\supset \mathsf{I}^1$$

↪

Expressive power

↪ Proof. (viii)

- $A \equiv \exists x.B$: by induction hypothesis there are $\vdash B \supset (B)^N$, $\vdash (B)^N \supset B$, and $(A)^N \equiv \neg\forall x.\neg(B)^N$ so

$$\dfrac{\dfrac{\dfrac{[\exists x.B]^1 \qquad \dfrac{\dfrac{[B]^2}{\vdots}}{(B)^N} \qquad \dfrac{[\forall x.\neg(B)^N]^3}{\neg(B)^N}\,\forall \mathsf{E}}{\bot}\,\neg \mathsf{E}}{\bot}\,\exists \mathsf{E}^2}{\neg\forall x.\neg(B)^N}\,\neg \mathsf{I}^3}{(\exists x.B) \supset \neg\forall x.\neg(B)^N}\,\supset \mathsf{I}^1$$

↪

Expressive power

↪ Proof. (ix)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[(B)^N]^2}{\vdots}}{B}}{\exists x.B}\,\exists \mathsf{I} \qquad [\neg(\exists x.B)]^1}{\bot}\,\neg \mathsf{E}}{\neg(B)^N}\,\neg \mathsf{I}^2}{\forall x.\neg(B)^N}\,\forall \mathsf{I} \qquad [\neg(\forall x.\neg(B)^N)]^3}{\bot}\,\neg \mathsf{E}}{\exists x.B}\,\bot \mathsf{E}$$

$$\dfrac{(\exists x.B) \vee \neg(\exists x.B)}{\exists x.B}\,\text{lem} \qquad [\exists x.B]^1\,\vee \mathsf{E}^1$$

$$\dfrac{\exists x.B}{\neg\left(\forall x.\neg(B)^N\right) \supset \exists x.B}\,\supset \mathsf{I}^3$$

□

Expressive power

Proposition 22.4

If $\pi : \Gamma \vdash A$ *in classical logic then there is* $\pi' : \left\{(\gamma)^N : \gamma \in \Gamma\right\} \vdash (A)^N$ *in intuitionistic logic.*

We will not prove this theorem: who is interested can inspect it having a look at the references at the end of this lesson.

The proposition has a number of consequences: the relevant ones to us are

- each classical theory and thus each classical proof can be translated into intuitionistic logic, yielding a classically equivalent result. So classical logic is not really more expressive than intuitionistic logic.
- Intuitionistic logic is more expressive than classical logic since it allows to distinguish formulæ which are classically equivalent.

References

A good introduction to the constructive way of reasoning can be found in *Anne Sjerp Troelstra* and *Dirk van Dalen*, Constructivism in Mathematics, volume I, Studies in Logic and the Foundations of Mathematics 121, Elsevier, (1988).

There are many ways to translate intuitionistic logic into classical logic. A survey can be found in *Anne Sjerp Troelstra* and *Helmut Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43, Cambridge University Press, (1996).

Syllabus

Constructive mathematics:

- Heyting algebra
- Semantics
- Soundness
- Completeness

Mathematical Logic

Lecture 23

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Heyting algebra

Definition 23.1 (Heyting algebra)

A *Heyting algebra* $\mathscr{H} = \langle H; \leq \rangle$ is a bounded lattice such that for every $x, y \in H$ there is $c \in H$, the *relative pseudo-complement* of x with respect to y, notation $x \supset y$, such that

1. $x \wedge c \leq y$;
2. for every $z \in H$ such that $x \wedge z \leq y$, $z \leq c$.

The relative pseudo-complement of $x \in H$ with respect to $\bot$ is called the *pseudo-complement* of x and it is denoted by $\neg x$.

Heyting algebra

Examples:

- Every Boolean algebra is also a Heyting algebra.
- Every totally ordered set forming a bounded lattice is a Heyting algebra. In particular $x \supset y = y$ when $y < x$, and $x \supset y = \top$ otherwise.
- The lattice of open sets in any topology is a Heyting algebra. In particular $A \supset B$ is the interior of $A^c \cup B$.

The last example shows that a Heyting algebra is not always a Boolean algebra since the interior of $A^c \cup B$ is usually different from $A^c \cup B$, or in logical terms $A \supset B \neq \neg A \vee B$.

Heyting algebra

Proposition 23.2 (Adjunction)

In every Heyting algebra, for every x, y, z, $x \leq y \supset z$ if and only if $x \wedge y \leq z$.

Proof.

If $x \leq y \supset z$ then $x \wedge y \leq y \wedge (y \supset z)$ since $\wedge$ is monotone, and $y \wedge (y \supset z) \leq z$ by definition of $\supset$, thus $x \wedge y \leq z$ by transitivity.

If $x \wedge y \leq z$ then $x \leq y \supset z$ by definition of $\supset$. □

Heyting algebra

Fact 23.3

In any Heyting algebra for each element x, $x \wedge \neg x = \bot$.

Proof.

By definition of bottom and pseudo-complement $\bot \leq x \wedge \neg x \leq \bot$. □

Fact 23.4

In any Heyting algebra for all elements x and y, $x \leq y$ if and only if $x \supset y = \top$.

Proof.

Since $x = x \wedge \top$ if $x \leq y$, $x \supset y = \top$ being $\top$ the maximal element z such that $x \wedge z \leq y$. Conversely, if $x \supset y = \top$ then $x \wedge (x \supset y) = x \wedge \top = x \leq y$ by definition of pseudo-complement. □

Heyting algebra

Fact 23.5

There is a Heyting algebra such that for some element x, $x \vee \neg x \neq \top$.

Proof.

Consider the total order $0 < 1/2 < 1$. It is immediate to check that it is a Heyting algebra. But $1/2 \vee \neg 1/2 = 1/2 \vee 0 = 1/2 \neq 1 = \top$. □

Heyting algebra

Proposition 23.6

Every Heyting algebra is a distributive lattice.

Proof.

It suffices to prove $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.
By definition of $\vee$, $y \leq y \vee z$ and $z \leq y \vee z$, thus by definition of $\wedge$, $x \wedge y \leq x$ and $x \wedge y \leq y \leq y \vee z$, so $x \wedge y \leq x \wedge (y \vee z)$.
Symmetrically, it holds that $x \wedge z \leq x \wedge (y \vee z)$.
Then by definition of $\vee$, $(x \wedge y) \vee (x \wedge z) \leq x \wedge (y \vee z)$.

Conversely, by definition of $\vee$:
$x \wedge y \leq (x \wedge y) \vee (x \wedge z)$ and $x \wedge z \leq (x \wedge y) \vee (x \wedge z)$.
So, by definition of $\supset$:
$y \leq (x \supset (x \wedge y) \vee (x \wedge z))$ and $z \leq (x \supset (x \wedge y) \vee (x \wedge z))$.
Thus by definition of $\vee$, $y \vee z \leq (x \supset (x \wedge y) \vee (x \wedge z))$.
Then by definition of $\supset$, $x \wedge (y \vee z) \leq (x \wedge y) \vee (x \wedge z)$. □

Propositional semantics

For the sake of simplicity we will consider just the pure logic instead of a generic theory in the following. The results can be naturally generalised.

Definition 23.7 (Semantics)

Fixed a Heyting algebra $\mathscr{H} = \langle H; \leq \rangle$ and a map $v \colon V \to H$ evaluating each variable in some element of H, the meaning $[\![A]\!]$ of a propositional formula A is a map from the set of formulæ to H inductively defined as

1. if $A \equiv x$ a variable, $[\![A]\!] = v(x)$;
2. $[\![\top]\!] = \top$ and $[\![\bot]\!] = \bot$;
3. $[\![B \wedge C]\!] = [\![B]\!] \wedge [\![C]\!]$, $[\![B \vee C]\!] = [\![B]\!] \vee [\![C]\!]$, $[\![B \supset C]\!] = [\![B]\!] \supset [\![C]\!]$, and $[\![\neg B]\!] = \neg [\![B]\!]$.

We say that a formula A is *valid* or *true* in the *model* $(\mathscr{H}, v)$ when $[\![A]\!] = \top$.

Soundness

Theorem 23.8 (Soundness)

If $\pi \colon \Gamma \vdash A$ is a proof in the intuitionistic natural deduction calculus then in every model $(\mathscr{H}, v)$ such that each $G \in \Gamma$ is valid, A is true.

Proof. (i)

Fixed a generic model by induction on the structure of a proof $\pi \colon \Delta \vdash B$ with Δ a finite set of assumptions, we prove that $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B]\!]$:

- if π is a proof by assumption $B \in \Delta$, so $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B]\!]$ by definition of $\wedge$.
- if π is an instance of $\top$-introduction $B \equiv \top$, thus by definition of $\top$, $\bigwedge_{D \in \Delta} [\![D]\!] \leq \top = [\![B]\!]$.
- if π is an instance of $\bot$-elimination by induction hypothesis and by definition of $\bot$, $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![\bot]\!] = \bot \leq [\![B]\!]$.
- if π is an instance of $\wedge$-introduction $B \equiv B_1 \wedge B_2$ and by induction hypothesis $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_1]\!]$ and $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_2]\!]$, so by definition of $\wedge$, $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_1]\!] \wedge [\![B_2]\!] = [\![B_1 \wedge B_2]\!] = [\![B]\!]$. ↪

Soundness

↪ Proof. (ii)

- if π is an instance of $\wedge_1$-elimination or $\wedge_2$-elimination then by induction hypothesis $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B \wedge B_1]\!] = [\![B]\!] \wedge [\![B_1]\!]$ or $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_1 \wedge B]\!] = = [\![B_1]\!] \wedge [\![B]\!]$, respectively. Thus by definition of $\wedge$, $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B]\!]$ in both cases.
- if π is an instance of $\vee_1$-introduction or $\vee_2$-introduction then $B \equiv B_1 \vee B_2$ and by induction hypothesis $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_1]\!]$ or $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_2]\!]$, respectively. Thus by definition of $\vee$, $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_1]\!] \vee [\![B_2]\!] = [\![B_1 \vee B_2]\!] = [\![B]\!]$ in both cases. ↪

Observe how all the cases till now have been proved exactly in the same way as we did on Boolean algebras.

Soundness

↪ Proof. (iii)

- if π is an instance of $\vee$-elimination, by induction hypothesis $[\![C_1]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$ and $[\![C_2]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$, so by definition of $\supset$, $[\![C_1]\!] \leq \bigwedge_{D\in\Delta}[\![D]\!] \supset [\![B]\!]$ and $[\![C_2]\!] \leq \bigwedge_{D\in\Delta}[\![D]\!] \supset [\![B]\!]$, thus $[\![C_1]\!] \vee [\![C_2]\!] = [\![C_1 \vee C_2]\!] \leq \bigwedge_{D\in\Delta}[\![D]\!] \supset [\![B]\!]$. Hence by definition of $\supset$, $[\![C_1 \vee C_2]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$.
  Since by induction hypothesis $\bigwedge_{D\in\Delta}[\![D]\!] \leq [\![C_1 \vee C_2]\!]$, by definition of $\wedge$, $[\![C_1 \vee C_2]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] = \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$. ↪

Observe how this case and the following ones use the properties of Heyting algebras. Also, note that by its definition $\neg A \equiv A \supset \bot$ in a Heyting algebra.

Soundness

↪ Proof. (iv)

- if π is an instance of $\supset$-introduction, $B \equiv B_1 \supset B_2$ and by induction hypothesis $[\![B_1]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B_2]\!]$. So by definition of $\supset$, $\bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B_1]\!] \supset [\![B_2]\!] = [\![B_1 \supset B_2]\!] = [\![B]\!]$.
- if π is an instance of $\supset$-elimination by induction hypothesis $\bigwedge_{D\in\Delta}[\![D]\!] \leq [\![C \supset B]\!] = [\![C]\!] \supset [\![B]\!]$ thus by definition of $\supset$, $[\![C]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$. Since by induction hypothesis $\bigwedge_{D\in\Delta}[\![D]\!] \leq [\![C]\!]$, by definition of $\wedge$, $[\![C]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] = \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$.
- if π is an instance of $\neg$-introduction, $B \equiv \neg C$ and by induction hypothesis $[\![C]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![\bot]\!] = \bot$. So by definition of $\neg$, $\bigwedge_{D\in\Delta}[\![D]\!] \leq \neg[\![C]\!] = [\![\neg C]\!] = [\![B]\!]$.
- if π is an instance of $\neg$-elimination, by induction hypothesis $\bigwedge_{D\in\Delta}[\![D]\!] \leq [\![\neg C]\!] = \neg[\![C]\!]$ thus by definition of $\neg$, $[\![C]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![\bot]\!]$. Since by induction hypothesis $\bigwedge_{D\in\Delta}[\![D]\!] \leq [\![C]\!]$, by definition of $\wedge$, $[\![C]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] = \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![\bot]\!] = \bot \leq [\![B]\!]$ by definition of $\bot$. ↪

Soundness

↪ Proof. (v)

Now consider $\pi : \Gamma \vdash A$ as in the statement of the theorem: since the proof π uses just a finite number of assumptions $\Gamma_0 \subseteq \Gamma$, by the induction above $\bigwedge_{G\in\Gamma_0}[\![G]\!] \leq [\![A]\!]$. But for each $G \in \Gamma$, $[\![G]\!] = \top$ by hypothesis, thus $\bigwedge_{G\in\Gamma_0}[\![G]\!] = \top \leq [\![A]\!] \leq \top$ by definition of $\top$. So by anti-symmetry $[\![A]\!] = \top$. □

Completeness

We will show a simplified completeness result. A more general result can be easily obtained by extending the presented core along the guidelines we followed in the classical case.

Theorem 23.9 (Completeness)

If the propositional formula A is valid in any Heyting model $(\mathscr{H}; v)$ then A is provable in the propositional natural deduction calculus for intuitionistic logic.

Proof. (i)

Let F be the collection of all formulæ. We define $A \sim B$ if and only $\vdash A = B$. Evidently, $\sim$ is an equivalence relation over F:

- $A \sim A$ since $\vdash A \supset A$;
- if $A \sim B$ then $\vdash A \supset B$ and $\vdash B \supset A$, so $B \sim A$;
- if $A \sim B$ and $B \sim C$ then $\vdash A \supset B$ and $\vdash B \supset C$, thus $\vdash A \supset C$ but also $\vdash C \supset B$ and $\vdash B \supset A$, so $\vdash C \supset A$ thus $A \sim C$. ↪

Completeness

↪ Proof. (ii)

Let $H = F/{\sim}$ and let $[A]_\sim \leq [B]_\sim$ exactly when $A \vdash B$.
Then $\langle H; \leq \rangle$ is an order since

- if $[A]_\sim = [A']_\sim$, $[B]_\sim = [B']_\sim$, and $[A]_\sim \leq [B]_\sim$, then $[A']_\sim \leq [B']_\sim$ because $A \sim A'$ and $B \sim B'$, thus $A' \vdash A$, $B \vdash B'$, and from $[A]_\sim \leq [B]_\sim$, $A \vdash B$, hence $A' \vdash B'$, i.e., $[A']_\sim \leq [B']_\sim$;
- $[A]_\sim \leq [A]_\sim$ because $A \vdash A$;
- if $[A]_\sim \leq [B]_\sim$ and $[B]_\sim \leq [A]_\sim$ then $A \vdash B$ and $B \vdash A$, so $\vdash A = B$, that is $A \sim B$, i.e., $[A]_\sim = [B]_\sim$;
- if $[A]_\sim \leq [B]_\sim$ and $[B]_\sim \leq [C]_\sim$ then $A \vdash B$ and $B \vdash C$, so $A \vdash C$, that is, $[A]_\sim \leq [C]_\sim$.

Also, $\langle H; \leq \rangle$ is bounded:

- $\bot = [\bot]_\sim$, indeed $\bot \vdash A$ for any formula A by $\bot$E, so $[\bot]_\sim \leq [A]_\sim$;
- $\top = [\top]_\sim$, indeed $A \vdash \top$ for any formula A by $\top$I, so $[A]_\sim \leq [\top]_\sim$. ↪

Completeness

↪ Proof. (iii)

Moreover, $\langle H; \leq \rangle$ is a lattice:

- $[A]_\sim \wedge [B]_\sim = [A \wedge B]_\sim$, indeed $A \wedge B \vdash A$ and $A \wedge B \vdash B$ by $\wedge$E, so $[A \wedge B]_\sim \leq [A]_\sim$ and $[A \wedge B]_\sim \leq [B]_\sim$; if $[C]_\sim \leq [A]_\sim$ and $[C]_\sim \leq [B]_\sim$ then $C \vdash A$ and $C \vdash B$, so $C \vdash A \wedge B$ by $\wedge$I, that is $[C]_\sim \leq [A \wedge B]_\sim$;
- $[A]_\sim \vee [B]_\sim = [A \vee B]_\sim$, indeed $A \vdash A \vee B$ and $B \vdash A \vee B$ by $\vee$I, so $[A]_\sim \leq [A \vee B]_\sim$ and $[B]_\sim \leq [A \vee B]_\sim$; if $[A]_\sim \leq [C]_\sim$ and $[B]_\sim \leq [C]_\sim$ then $A \vdash C$ and $B \vdash C$, so $A \vee B \vdash C$ by $\vee$E, that is $[A \vee B]_\sim \leq [C]_\sim$.

Observe how up to now, the proof is identical to the one for constructing the canonical Boolean algebra.

Finally $\langle H; \leq \rangle$ is a Heyting algebra: $[A]_\sim \supset [B]_\sim = [A \supset B]_\sim$, indeed $A \wedge (A \supset B) \vdash B$, so $[A \wedge (A \supset B)]_\sim = [A]_\sim \wedge [A \supset B]_\sim \leq [B]_\sim$; when $[A]_\sim \wedge [C]_\sim = [A \wedge C]_\sim \leq [B]_\sim$, $A \wedge C \vdash B$, so $C \vdash A \supset B$, that is $[C]_\sim \leq [A \supset B]_\sim$.
It is worth noting that $\neg [A]_\sim = [\neg A]_\sim$ since $\vdash \neg A = (A \supset \bot)$. ↪

Completeness

↪ Proof. (iv)

Let $v \colon V \to H$ be $v(x) = [x]_\sim$ for any variable x.

By induction on the structure of A we prove that $[\![A]\!] = [A]_\sim$ in $((H; \leq), v)$:

- if $A \equiv x$, a variable, by definition $[\![A]\!] = v(x) = [x]_\sim = [A]_\sim$;
- if $A \equiv \top$, $[\![A]\!] = \top = [\top]_\sim$;
- if $A \equiv \bot$, $[\![A]\!] = \bot = [\bot]_\sim$;
- if $A \equiv B \wedge C$, by induction hypothesis $[\![A]\!] = [\![B]\!] \wedge [\![C]\!] = [B]_\sim \wedge [C]_\sim = [B \wedge C]_\sim = [A]_\sim$;
- if $A \equiv B \vee C$, by induction hypothesis $[\![A]\!] = [\![B]\!] \vee [\![C]\!] = [B]_\sim \vee [C]_\sim = [B \vee C]_\sim = [A]_\sim$;
- if $A \equiv B \supset C$, by induction hypothesis $[\![A]\!] = [\![B]\!] \supset [\![C]\!] = [B]_\sim \supset [C]_\sim = [B \supset C]_\sim = [A]_\sim$;
- if $A \equiv \neg B$, by induction hypothesis $[\![A]\!] = \neg [\![B]\!] = \neg [B]_\sim = [\neg B]_\sim = [A]_\sim$. ↪

Completeness

↪ Proof. (v)

By hypothesis of the theorem, A is valid in any model that is $[\![A]\!] = \top$ in any model, so in particular $[\![A]\!] = \top$ in $((H; \leq), v)$.
But in $((H; \leq), v)$, $[A]_\sim = [\![A]\!] = \top = [\top]_\sim$, thus $A \sim \top$, from which $\top \vdash A$.
By $\top$I and $\top \vdash A$ we get that $\vdash A$. □

Observe how the proof in the last two slides follows the same line as the one on Boolean algebras.

References

Heyting algebras have been introduced by Arend Heyting in 1930 to formalise intuitionistic logic. An algebraic introduction to Heyting algebras is in *George Grätzer*, General Lattice Theory, second edition, Birkhäuser, (1996).

The soundness theorem as presented is folklore: the actual presentation derives from the generalised result on the internal logic of topos theory, which is based on the fact that the lattice of subobjects of the terminal object in a topos forms a Heyting algebra. The details can be found in *Robert Goldblatt*, Topoi: The Categorical Analysis of Logic, Dover Publishing, (2006).

The proof of the completeness theorem has been adapted from the categorical version in *Peter Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002).

Mathematical Logic

Lecture 24

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Constructive mathematics:

- Propositions as types
- Proofs, computationally
- Computations, logically
- Normalisation

Propositions as types

If we put side by side propositional logical formulæ and types in the simple theory of types, we get:

types	formulæ
variable	variable
0	$\bot$
1	$\top$
$\alpha \times \beta$	$\alpha \wedge \beta$
$\alpha + \beta$	$\alpha \vee \beta$
$\alpha \to \beta$	$\alpha \supset \beta$

This correspondence shows that we can translate any logical formula in a type and any type in a formula by a one-to-one map.

Propositions as types

If we put side by side the inference rules in the intuitionistic natural deduction system, and the term constructors in the simple theory of types, we get:

proof	assumption	$\top\mathsf{I}$	$\bot\mathsf{E}$	$\wedge\mathsf{I}$	$\wedge\mathsf{E}_{1,2}$	$\vee\mathsf{I}_{1,2}$	$\vee\mathsf{E}$	$\supset\mathsf{I}$	$\supset\mathsf{E}$
term	variable	$*$	$\Box_\alpha$	$\langle \_, \_ \rangle$	π_1, π_2	i_1^α, i_2^α	δ	λ	$\cdot$

There is an evident one-to-one correspondence which perfectly matches the one on types.

Propositions as types

Example 24.1
If $A\colon \alpha$ and $B\colon \beta$ are terms then $\langle A, B\rangle\colon \alpha \times \beta$ becomes

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ A & B \\ \alpha & \beta\end{array}}{\alpha \wedge \beta}\,\wedge\mathsf{I}$$

Example 24.2
If $A\colon \beta$ is a term and $x\colon \alpha$ a variable then $\lambda x\colon \alpha.A\colon \alpha \to \beta$ becomes

$$\dfrac{\begin{array}{c}[\alpha]^* \\ \vdots \\ A \\ \beta\end{array}}{\alpha \supset \beta}\,\supset\mathsf{I}^*$$

where the label $*$ stands for x.

Propositions as types

Example 24.3
Consider the proof:

$$\dfrac{\dfrac{\dfrac{[\neg A]^1 \quad [A]^2}{\bot}\,\neg\mathsf{E}}{\neg\neg A}\,\neg\mathsf{I}^1}{A \supset \neg\neg A}\,\supset\mathsf{I}^2$$

It gets translated in the typed term:

$$\lambda x_2\colon A, x_1\colon A \to 0.\, x_1 \cdot x_2\colon A \to ((A \to 0) \to 0)\ .$$

Propositions as types

The correspondence illustrated so far is known as the *propositions-as-types interpretation* and also as the *Curry-Howard isomorphism*.

At a first glance the simple theory of types is just a way to write proofs and formulæ as linear expressions instead of adopting the tree-like syntax of natural deduction.

However the logical syntax is coupled with a semantics, and the type theory with a computational meaning given by the reduction rules.

Computations, logically

Since every formal proof in intuitionistic logic corresponds to a typed term which are also λ-terms, each proof is a program which computes something.

It is possible to associate to each proof an object, which is an *evidence* of its type, or its conclusion if you prefer. So the evidence of $A \wedge B$ is a pair of evidences for A and B; the evidence of $A \vee B$ is a pair (w, e) with $w \in \{1,2\}$ telling us which disjunct holds and e an evidence for it; the evidence of $A \supset B$ is a function mapping any evidence of A into an evidence of B.

These evidences are the intermediate results of the computation performed by the λ-term associated to the proof. So in a constructive system proving a statement is essentially equivalent to write a computer program satisfying a specification given by the conclusion.

Proofs, computationally

Since typed terms are proofs under the correspondence, we can reduce them to a normal form. Formalising this process leads to state that every proof possesses a normal form.

Thus, considering any proof $\pi \colon \vdash A \vee B$ it can be reduced to a proof $\pi' \colon \vdash A \vee B$ in normal form whose last step is either an instance of $\vee \mathrm{I}_1$ or $\vee \mathrm{I}_2$. Hence the conclusion of the last but one step would be either A or B.

Similarly, considering any proof $\pi \colon \vdash \exists x \colon s.A$ it can be reduced to a proof $\pi' \colon \vdash \exists x \colon s.A$ in normal form whose last step is an instance of $\exists \mathrm{I}$. Hence the conclusion of the last but one step would be $A[t/x]$ for some term t, providing a witness to the existential statement.

Normalisation

We want to discuss the normalisation process, which has been sketched before, in the case of intuitionistic propositional logic.

The objective of normalisation is to eliminate the redundant steps in a proof and to give it a standard format, *minimal* in a sense.
Here, minimal means "most direct".

A natural requirement for a proof in natural deduction is that no conclusion of an introduction rule must be the major premise of an elimination rule. The major premise is the formula containing as principal connective the one which is eliminated by an elimination rule.

Also another natural requirement is that discharged assumptions should be used in disjunction elimination, while the false elimination rule has to derive a conclusion which is not $\bot$.

Finally, although the previous requirements seem evident they can be hidden, because of multiple subsequent elimination rules which can be permuted.

Normalisation

The *detour conversions* are deputed to eliminate detours, i.e., redundant elementary steps in a proof given by an introduction rule in the major premise of an elimination rule:

- $\wedge$ rules:

$$\dfrac{\dfrac{\begin{array}{cc}\vdots\,p_1 & \vdots\,p_2\\ A & B\end{array}}{A \wedge B}\wedge\mathrm{I}}{A}\wedge\mathrm{E}_1 \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_1\\ A\end{array} \qquad \dfrac{\dfrac{\begin{array}{cc}\vdots\,p_1 & \vdots\,p_2\\ A & B\end{array}}{A \wedge B}\wedge\mathrm{I}}{B}\wedge\mathrm{E}_2 \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_2\\ B\end{array}$$

- $\supset$ rules:

$$\dfrac{\dfrac{\begin{array}{c}[A]^1\\ \vdots\,p_1\\ B\end{array}}{A \supset B}\supset\mathrm{I}^1 \quad \begin{array}{c}\vdots\,p_2\\ A\end{array}}{B}\supset\mathrm{E} \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_2\\ A\\ \vdots\,p_1\\ B\end{array}$$

Normalisation

- $\vee$ rules:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,p_1\\ A\end{array}}{A\vee B}\vee\mathrm{I}_1 \quad \begin{array}{c}[A]^1\\ \vdots\,p_2\\ C\end{array} \quad \begin{array}{c}[B]^1\\ \vdots\,p_3\\ C\end{array}}{C}\vee\mathrm{E}^1 \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_1\\ A\\ \vdots\,p_2\\ C\end{array}$$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,p_1\\ B\end{array}}{A\vee B}\vee\mathrm{I}_2 \quad \begin{array}{c}[A]^1\\ \vdots\,p_2\\ C\end{array} \quad \begin{array}{c}[B]^1\\ \vdots\,p_3\\ C\end{array}}{C}\vee\mathrm{E}^1 \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_1\\ B\\ \vdots\,p_3\\ C\end{array}$$

Normalisation

Since $\neg A \equiv A \supset \bot$ we do not need detour conversions for $\neg$ rules as soon as we rewrite them as instances of the $\supset$ rules. The conversions for $\supset$ and $\vee$ are justified by Proposition 6.2, which allows to join proofs.

There are no detour conversions for $\bot$ and $\top$ since these connectives lack an introduction and elimination rule, respectively.

It is instructive to see these conversions through the propositions-as-types correspondence:

- $\wedge$ rules: $\pi_1\langle p_1, p_2\rangle \rightsquigarrow p_1$ and $\pi_2\langle p_1, p_2\rangle \rightsquigarrow p_2$;
- $\supset$ rules: $(\lambda x_1 : A.\, p_1)\cdot p_2 \rightsquigarrow p_1[p_2/x_1]$;
- $\vee$ rules: $\delta(i_1\, p_1, p_2, p_3) \rightsquigarrow p_2[p_1/x_1]$ and $\delta(i_2\, p_1, p_2, p_3) \rightsquigarrow p_3[p_1/x_1]$.

This observation shows that the conversion rules are precisely the reduction rules of the simple theory of types.

Normalisation

Detour conversions eliminate obviously redundant steps in a proof. However, there are instances of the disjunction elimination rule that are indeed redundant, those in which one of the discharged assumptions is not used. This fact leads to define the following *simplification conversions*: if in

$$\dfrac{\begin{array}{ccc}\vdots\,p_1 & [A]^1 & [B]^1\\ & \vdots\,p_2 & \vdots\,p_3\\ A\vee B & C & C\end{array}}{C}\vee\mathrm{E}^1$$

either the assumption A in p_2 is not used or the assumption B in p_3 is not used then we can use p_2 or p_3, respectively to prove the conclusion.

Normalisation

$$\dfrac{\begin{array}{ccc}\vdots\,p_1 & \vdots\,p_2 & [B]^1\\ & & \vdots\,p_3\\ A\vee B & C & C\end{array}}{C}\vee\mathrm{E}^1 \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_2\\ C\end{array}$$

$$\dfrac{\begin{array}{ccc}\vdots\,p_1 & [A]^1 & \\ & \vdots\,p_2 & \vdots\,p_3\\ A\vee B & C & C\end{array}}{C}\vee\mathrm{E}^1 \quad\rightsquigarrow\quad \begin{array}{c}\vdots\,p_3\\ C\end{array}$$

Normalisation

Moreover, the instances of the $\bot$ elimination rule in which the conclusion is $\bot$ are obviously redundant and we can apply another *simplification conversion* to eliminate them.

$$\dfrac{\begin{array}{c}\vdots\\ \bot\end{array}}{\bot}\,\bot\mathrm{E} \quad\rightsquigarrow\quad \begin{array}{c}\vdots\\ \bot\end{array}$$

In the Curry-Howard isomorphism, these conversions map to the admissible reductions:

- $\delta(p_1, K\,p_2, p_3) \rightsquigarrow p_2$;
- $\delta(p_1, p_2, K\,p_3) \rightsquigarrow p_3$;
- $\Box_0 p \rightsquigarrow p$.

Normalisation

Sometimes detours and simplifications cannot be directly applied because they are hidden inside a proof. This happens when we apply an elimination rule whose major premise is an application of the disjunction elimination rule.

In those cases, we can move the disjunction elimination downwards, eventually revealing hidden detours and simplifications. The rules to do so are called *permutation conversions*.

Normalisation

- $\wedge$ elimination:

$$\dfrac{\dfrac{\begin{array}{ccc}\vdots & [A]^1 & [B]^1\\ p_1 & \vdots & \vdots\\ A\vee B & C\wedge D & C\wedge D\end{array}}{C\wedge D}\,\vee\mathrm{E}^1}{C}\,\wedge\mathrm{E}_1 \quad\rightsquigarrow\quad \dfrac{\begin{array}{ccc}\vdots & [A]^1 & [B]^1\\ p_1 & \vdots & \vdots\\ A\vee B & C\wedge D & C\wedge D\end{array}}{\dfrac{C}{C}\,\wedge\mathrm{E}_1 \quad \dfrac{C}{C}\,\wedge\mathrm{E}_1}\,\vee\mathrm{E}^1$$

$$\dfrac{\dfrac{\begin{array}{ccc}\vdots & [A]^1 & [B]^1\\ p_1 & \vdots & \vdots\\ A\vee B & C\wedge D & C\wedge D\end{array}}{C\wedge D}\,\vee\mathrm{E}^1}{D}\,\wedge\mathrm{E}_2 \quad\rightsquigarrow\quad \dfrac{\begin{array}{ccc}\vdots & [A]^1 & [B]^1\\ p_1 & \vdots & \vdots\\ A\vee B & C\wedge D & C\wedge D\end{array}}{\dfrac{D}{D}\,\wedge\mathrm{E}_2 \quad \dfrac{D}{D}\,\wedge\mathrm{E}_2}\,\vee\mathrm{E}^1$$

Normalisation

- $\bot$ elimination:

$$\dfrac{\dfrac{\begin{array}{ccc}\vdots & [A]^1 & [B]^1\\ p_1 & \vdots & \vdots\\ A\vee B & \bot & \bot\end{array}}{\bot}\,\vee\mathrm{E}^1}{C}\,\bot\mathrm{E} \quad\rightsquigarrow\quad \dfrac{\begin{array}{ccc}\vdots & [A]^1 & [B]^1\\ p_1 & \vdots & \vdots\\ A\vee B & \dfrac{\bot}{C}\,\bot\mathrm{E} & \dfrac{\bot}{C}\,\bot\mathrm{E}\end{array}}{C}\,\vee\mathrm{E}^1$$

Normalisation

- $\supset$ elimination:

$$\dfrac{\dfrac{A \vee B \quad \overset{[A]^1}{\vdots\, C \supset D} \quad \overset{[B]^1}{\vdots\, C \supset D}}{C \supset D}\vee E^1 \quad \begin{matrix}\vdots\, p_1 \\ \vdots\, p_2 \\ \vdots\, p_3\end{matrix} \qquad \begin{matrix}\vdots\, p_4 \\ C\end{matrix}}{D}\supset E \quad \rightsquigarrow$$

$$\rightsquigarrow \quad \dfrac{\begin{matrix}\vdots\, p_1 \\ A \vee B\end{matrix} \quad \dfrac{\overset{[A]^1}{\vdots\, p_2} \quad \begin{matrix}\vdots\, p_4 \\ C\end{matrix}}{D}\supset E \quad \dfrac{\overset{[B]^1}{\vdots\, p_3} \quad \begin{matrix}\vdots\, p_4 \\ C\end{matrix}}{D}\supset E}{D}\vee E^1$$

Normalisation

- $\vee$ elimination:

$$\dfrac{\dfrac{A \vee B \quad \overset{[A]^1}{\vdots\, C \vee D} \quad \overset{[B]^1}{\vdots\, C \vee D}}{C \vee D}\vee E^1 \quad \begin{matrix}\vdots\, p_1 \\ \vdots\, p_2 \\ \vdots\, p_3\end{matrix} \qquad \begin{matrix}\overset{[C]^2}{\vdots\, p_4} \\ E\end{matrix} \quad \begin{matrix}\overset{[D]^2}{\vdots\, p_5} \\ E\end{matrix}}{E}\vee E^2 \quad \rightsquigarrow$$

$$\rightsquigarrow \quad \dfrac{\begin{matrix}\vdots\, p_1 \\ A \vee B\end{matrix} \quad \dfrac{\overset{[A]^1}{\vdots\, p_2} \quad \overset{[C]^2}{\vdots\, p_4} \quad \overset{[D]^2}{\vdots\, p_5}}{E}\vee E^2 \quad \dfrac{\overset{[B]^1}{\vdots\, p_3} \quad \overset{[C]^3}{\vdots\, p_4} \quad \overset{[D]^3}{\vdots\, p_5}}{E}\vee E^3}{E}\vee E^1$$

Normalisation

By applying all these conversions, mimicking the reduction process of the simple theory of types, we get the following result

Theorem 24.4 (Normalisation)

Each derivation in intuitionistic natural deduction reduces to a normal derivation, in which none of the detour, simplification, and permutation conversions can be applied.

Although we are not going the see the proof since it relies on a complex induction, we are able to derive a few consequences which are relevant.

Theorem 24.5 (Subformula property)

Let $\pi \colon \Gamma \vdash A$ be a normal derivation in intuitionistic propositional logic. Then each formula in π is a subformula of some formula in $\Gamma \cup \{A\}$.

Normalisation

By looking at the proof of the Normalisation Theorem,

Corollary 24.6

Let $\pi \colon \Gamma \vdash A$ be a normal derivation in intuitionistic propositional logic. If A is not atomic or $\bot$ then the last step is an introduction rule.

An immediate consequence is that disjunction is decidable.

Corollary 24.7 (Disjunction property)

Let $\pi \colon \Gamma \vdash A \vee B$ be a normal derivation in intuitionistic propositional logic. Then there is a subproof π' of π whose conclusion is either A or B.

Similar results hold for intuitionistic first order logic, and in particular

Corollary 24.8 (Explicit definability)

Let $\pi \colon \Gamma \vdash \exists x.A$ be a normal derivation in intuitionistic first order logic. Then there is a subproof π' of π whose conclusion is $A[t/x]$ for some term t.

Normalisation

It is important to remark that we have proved these results about normalisation in the natural deduction system for pure propositional logic. Choosing a different deductive system although sound and complete, does not necessarily lead to the same result.

Also adding a theory and thus instances of the axiom rule may lead to alternative normalisation procedures, or to systems in which normalisation cannot be obtained.

In these cases the constructive nature of intuitionistic logic, stemming from Corollaries 24.7 and 24.8 is not automatically achieved.
As an obvious counterexample consider that classical logic is just intuitionistic logic plus the theory $\{A \vee \neg A\colon A \text{ formula}\}$.

Normalisation

An extremely important consequence of normalisation is that intuitionistic propositional and first order logics are consistent.

Suppose $\vdash \bot$. Then there is a normal proof $\pi\colon \vdash \bot$ by Theorem 24.4.

Moreover, by Theorem 24.5, every formula occurring in π must be a subformula of $\bot$.

Also, by Corollary 24.6, the last step of π must not be an introduction rule.

Hence, the last step of π is necessarily an instance of $\bot$E from the premise $\bot$.

Which is impossible, being π in normal form.

Observe that, if $\vdash \bot$ in classical propositional logic or in classical first-order logic, then $\vdash (\bot)^N$ in the corresponding intuitionistic logic via the Gödel-Gentzen translation. But this is impossible since $(\bot)^N = \bot$.

Hence, the classical systems are consistent.

References

The propositions-as-types interpretation and the normalisation theorem are illustrated in many textbooks: the lesson has been adapted from *Anne Sjerp Troelstra* and *Helmut Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43, Cambridge University Press, (1996). An analysis of normalisation can be found in *Sara Negri* and *Jan Von Plato*, Structural Proof Theory, Cambridge University Press (2001).

A more computer science oriented text is *Benjamin C. Pierce*, Types and Programming Languages, The MIT Press, (2002).

Mathematical Logic

Lecture 25

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Constructive mathematics:

- Semantics of first-order intuitionistic logic
- Heyting categories
- Logical categories
- Kripke semantics
- Realizability

Semantics

The algebraic semantics based on Heyting algebras can be generalised to provide a meaning to first-order intuitionistic logic.

There are many ways to achieve this result, obtaining a soundness and completeness theorem:

- Heyting categories;
- Logical categories;
- Kripke semantics.

Heyting categories

Heyting categories are categories with a somewhat involved structure such that the class of sub-objects of any object form a Heyting algebra, ordered by the factorisation of morphisms.

Although it is beyond the scope of these lessons to provide a formal account, the idea is that quantifiers get a meaning by considering the maximal and the minimal element in a Heyting algebra which is related to the algebra used to interpret the quantified formula, so these extreme elements are generated by the relation of algebras, modelling the elimination of the quantified variable.

Toposes

Since any topos is also a Heyting category, one can limit the class of models to toposes. It turns out that it suffices to prove a completeness result.

Moreover, a further limitation to Grothendieck toposes suffices too. This becomes interesting because a topos of sheaves, the prototypical Grothendieck topos, provides a model which is composed by a collection of almost classical models a la Tarski, but in the internal set theory of the topos linked together by a relation modelling the growth of knowledge implicit in the constructive nature of intuitionistic first-order logic.

These models suffice to prove a completeness result, too.

Toposes

Models based on Heyting categories, toposes, or Grothendieck toposes are extremely useful and interesting. However, their study lies far beyond the scope of this course.

Interested students may find more about this fascinating topic in the courses about Topos Theory and Categorical Logic.

Logical categories

On a different line, by using categories naturally equipped with a Heyting algebra and a sort of topological structure modelling the link between a quantified formula and its instances through the introduction and elimination inference rules, one obtains another sound and complete semantics.

These categories are known as *logical categories*. They are simpler than Heyting categories, and allow to prove a completeness result in which there is a simple classifying model.

All these semantics are strictly related one to the other, emphasising some aspects of the deep nature of constructive logical systems, and this is the reason why all of them have been developed.

Kripke's semantics

The semantics based on Grothendieck toposes can be further specialised to the category of sets and functions. Again, it is possible to prove a soundness and completeness result.

This semantics is closer to Tarski's semantics for first-order classical logic, and it is called *Kripke's semantics*.

Its precise definition and its properties will be sketched in the next slides: the definition is quite technical, and difficult to justify (indeed, it is much easier looking at the categorical description). The soundness theorem poses no difficulties: it is proved, as usual, by an induction on the structure of proofs to show that the inference rules preserve validity.

The completeness theorem is similar to the one for classical logic, with a more involved saturation (the analogous of the construction of a Henkin set given a consistent set). In the end, using sets, the saturation is just more complex than the one for Tarski's semantics, but with no conceptual novelties.

Kripke's semantics

Definition 25.1 (Kripke's structure)

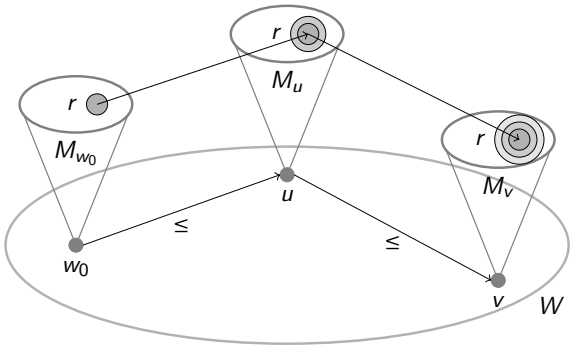
Let Σ be a first-order signature with just one sort s. A *Kripke's structure* on Σ is $\mathscr{K} = \langle W, \leq, \{M_w\}_{w \in W} \rangle$ where $\langle W, \leq \rangle$ is a preorder with a minimum $w_0 \in W$ and M_w is a Σ-structure for every $w \in W$ such that

- $[\![s]\!]_{M_u} \subseteq [\![s]\!]_{M_v}$ when $u \leq v$;
- for every function symbol f in Σ, if $u \leq v$ then $[\![f]\!]_{M_u} = [\![f]\!]_{M_v} \restriction [\![s]\!]_{M_u}$, with $\restriction$ the restriction of the function on the left to the domain on the right;
- for every relation symbol r in Σ, if $u \leq v$ then $[\![r]\!]_{M_u} = [\![r]\!]_{M_v} \restriction [\![s]\!]_{M_u}$, with $\restriction$ the restriction of the relation on the left to the domain on the right.

Intuitively, a Kripke's structure is a collection of classical Σ-structures, extending each other in accord with the indexing preorder.

Kripke's semantics

In a figure:



Kripke's semantics

Definition 25.2 (Forcing)

Fixed a Kripke's structure $\langle W, \leq, \{M_w\}_{w \in W} \rangle$, a *world* $w \in W$, and an evaluation of variables e in M_w, the *forcing* $\Vdash$ of a formula in w under e is inductively defined as:

- $w \Vdash_e \top$;
- $w \not\Vdash_e \bot$;
- $w \Vdash_e r(t_1, \ldots, t_n)$, with r a relation symbol, if and only if $(\llbracket t_1 \rrbracket_{w,e}, \cdots, \llbracket t_n \rrbracket_{w,e}) \in \llbracket r \rrbracket_w$;
- $w \Vdash_e A \wedge B$ if and only if $w \Vdash_e A$ and $w \Vdash_e B$;
- $w \Vdash_e A \vee B$ if and only if $w \Vdash_e A$ or $w \Vdash_e B$;
- $w \Vdash_e A \supset B$ if and only if, for every $u \geq w$, $u \not\Vdash_e A$ or $u \Vdash_e B$;
- $w \Vdash_e \exists x.A$ if and only if there is $a \in M_w$ such that $w \Vdash_{e[a/x]} A$;
- $w \Vdash_e \forall x.A$ if and only if, for every $u \geq w$ and for every $a \in M_u$, it holds $u \Vdash_{e[a/x]} A$.

Kripke's semantics

Observe how the forcing relation, when one does not take into account the worlds beyond w, reduces to the usual semantics for first order classical logic.

In other terms, Kripke's semantics extends the classical semantics by considering multiple models, each one connected to the others through the $\leq$ relation. The interesting aspect is that the ordering of the worlds induces a coherent extension of domains, and thus of interpretations.

So, we can recover the classical semantics by considering Kripke's semantics on the trivial preorder with just one element.

Kripke's semantics

Fixed a Kripke's structure, a formula A is *valid* in the w world under the evaluation of variables e on M_w when $w \Vdash_e A$.

In turn, fixed a Kripke's structure and an evaluation of variables e on w_0, the initial world, a formula is valid in the structure under e when it is valid in w_0.

A Kripke's model for a theory T is a pair composed by a Kripke's structure and an evaluation of variables on w_0, making all the formulæ of T valid.

Theorem 25.3 (Soundness and Completeness)

Fixed a theory T and a formula A:

- *if $\vdash_T A$ in intuitionistic first order logic, then A is valid in every Kripke's model for T;*
- *if A is valid in every Kripke's model for T, then $\vdash_T A$.*

Realizability

On a different line, some intuitionistic theories admit a very interesting semantics that links them to computability.

The prominent one is Kleene's *realizability*, which interprets the (intuitionistic) truth of arithmetical statements on natural numbers.

To understand the key idea, let's fix a good enumeration of all the partial recursive functions $\{\phi_i\}_{i\in\omega}$. Observing that $\top$ can be defined as $0=0$, we can decide (*realise*) whether a formula F with no free variables is true by providing just a number.

Indeed, any atomic formula has the form $t=s$ which is decidable, so no additional information is needed to realise it; $A\wedge B$ is realised when we have two numbers a and b realising A and B, so the pair $\langle a,b\rangle$ realises $A\wedge B$; to realise $A\vee B$ it suffices to have a number n which tells which disjunct holds, and another number to realise that disjunct; to realise $A\supset B$ it suffices to have the index i of a function ϕ_i mapping realizers of A into realizers of B.

Realizability

Definition 25.4 (Kleene's realizability)

Let E be a sentence in the language of arithmetic. Fix a good enumeration $\{\phi_i\}_{i\in\omega}$ of all the partial recursive functions.
Then $e\in\mathbb{N}$ *realises* E, notation $e\Vdash E$, when

- if E is atomic, i.e., $E\equiv(s=t)$ then $e\Vdash E$ exactly when E is valid;
- $e\nVdash\bot$;
- $e\Vdash A\wedge B$ when there are $a,b\in\mathbb{N}$ such that $e=\langle a,b\rangle$, $a\Vdash A$ and $b\Vdash B$;
- $e\Vdash A\vee B$ when there are $c,d\in\mathbb{N}$ such that $e=\langle c,d\rangle$, and $d\Vdash A$ when $c=0$, and $d\Vdash B$ when $c\neq 0$;
- $e\Vdash A\supset B$ when, for every $a\in\mathbb{N}$ such that $a\Vdash A$, $\phi_e(a)$ is defined and $\phi_e(a)\Vdash B$;
- $e\Vdash\forall x.A$ when, for every $n\in\mathbb{N}$, $\phi_e(n)$ is defined and $\phi_e(n)\Vdash A[S^n(0)/x]$;
- $e\Vdash\exists x.A$ when $e=\langle n,a\rangle$ and $a\Vdash A[S^n(0)/x]$.

For a formula A, $e\Vdash A$ when $e\Vdash\forall x_1,\ldots,x_n.A$ with $\mathrm{FV}(A)=\{x_1,\ldots,x_n\}$.

Realizability

The fundamental result is soundness:

Theorem 25.5 (Nelson)

If $\Gamma\vdash A$ in Heyting arithmetic, and every hypothesis in Γ is realisable, so is the conclusion.

Heyting arithmetic is the standard intuitionistic formal theory of naturals: it will be introduced in the next lecture.

There is not a corresponding completeness theorem because some principles which cannot be derived in Heyting arithmetic, are still realised. For example, the already cited formal Church Thesis.

However, the Law of Excluded Middle is easily shown to be non-realisable.

References

Heyting categories are defined in *Peter Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002).

The internal logic of a topos has been introduced by William Lawvere and an approachable text is *Robert Goldblatt*, Topoi: The Categorical Analysis of Logic, Dover Publishing, (2006).

Logical categories have been introduced by Marco Benini and a quite technical survey can be found in *Marco Benini*, Proof-Oriented Categorical Semantics, in D. Probst, P. Schuster eds., Concepts of Proof in Mathematics, Philosophy, and Computer Science, Ontos Mathematical Logic 6, De Gruyter, pp. 41-68 (2016).

References

The presentation of Kripke's semantics derives from *Saul A. Kripke* Semantical analysis of intuitionistic logic, pages 92–130, in *J. Crossley* and M.A.E. Dummett editors, Formal Systems and Recursive Functions, North-Holland Publishing (1965).

Realizability as presented here has been introduced in *Sthephen C. Kleene* On the interpretation of intuitionistic number theory, Journal of Symbolic Logic 10, 109–124 (1945).

Mathematical Logic

Lecture 26

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Limiting results:

- Peano arithmetic
- Standard and non-standard models
- Representable entities

Peano arithmetic

Peano arithmetic is the standard formal theory describing natural numbers and their properties.

It is composed by a series of axioms divided into groups, and it is interpreted in classical first order logic.

The very same theory interpreted in intuitionistic first order logic is called Heyting arithmetic. Despite they are syntactically identical, their interpretations are quite different. For example, in Peano arithmetic it is possible to show that there are functions which cannot be computed, while every function which can be proved to exist in Heyting arithmetic, is computable because of the constructive nature of the logic.

Peano arithmetic

Peano arithmetic is based on the language generated by the the signature

$$\langle \{\mathbb{N}\}; \{0 : \mathbb{N}, S : \mathbb{N} \to \mathbb{N}, +, \cdot : \mathbb{N} \times \mathbb{N} \to \mathbb{N}\}; \{= : \mathbb{N} \times \mathbb{N}\} \rangle \ .$$

The first group of axioms defines what is a natural number:

$$\forall x, y. S x = S y \supset x = y \ ; \tag{1}$$

$$\forall x. S x \neq 0 \ . \tag{2}$$

The idea is that natural numbers are the elements of the free algebra generated by 0 and S. The successor function S given a number x calculates the next number, $x+1$. So natural numbers are written in the unary representation and they are naturally equipped with a total order structure with minimum.

Peano arithmetic

The second group of axioms defines addition and multiplication:

$$\forall x. 0 + x = x \ ; \tag{3}$$

$$\forall x, y. S x + y = S(x + y) \ ; \tag{4}$$

$$\forall x. 0 \cdot x = 0 \ ; \tag{5}$$

$$\forall x, y. S x \cdot y = x \cdot y + y \ . \tag{6}$$

It is worth remarking the inductive nature of these definitions.

Peano arithmetic

The third and last group of axioms is a schema: for any formula A

$$A[0/x] \wedge (\forall x. A \supset A[S x/x]) \supset \forall x. A \tag{7}$$

This schema formalises induction on the structure of natural numbers:

- if A holds on 0
- and assuming that A holds on x we can show that it holds on $S x$,
- then A holds for every $x \in \mathbb{N}$.

Standard model

The intended model for Peano arithmetic is the structure which interprets the signature as

- the unique sort into the set of natural numbers denoted by $\mathbb{N}$;
- the function symbols into the zero number, the successor function, and the usual addition and multiplication, respectively.

Any model, i.e., any pair $(\mathscr{M}, \sigma)$ is said to be *standard* when $\mathscr{M}$ is the structure above while no restriction is posed on the evaluation σ of variables.
Although it may be apparently confusing we adopt the notation which uses the same symbols to denote the formal elements of the syntax and their intended interpretation.
In any standard model this convention makes no difference.

Since the purpose of the theory of arithmetic is to characterise the class of standard models, it would be nice if these were the only models of the theory. Unfortunately this is not the case.

Non-standard models

Definition 26.1 (Non-standard model)

Any structure $\mathcal{N}$ on the language of Peano arithmetic which is not isomorphic to the standard model $\mathcal{M}$ but for some evaluation σ of variables is a model $(\mathcal{N},\sigma)$ of Peano arithmetic, is called a *non-standard model*.

In the definition above an isomorphism between structures $f\colon \mathcal{N}\to\mathcal{M}$ is

- an invertible function between the universes;
- for each term t, $f([\![t]\!]_{\mathcal{N}}) = [\![t]\!]_{\mathcal{M}}$.

If a non-standard model exists it means that there is a structure $\mathcal{N}$ which makes Peano arithmetic true but interprets some term into an element e in the universe which cannot be mapped in some natural number.

Non-standard models

Proposition 26.2

There is a non-standard model for Peano arithmetic.

Proof.

Define $S^0(0)=0$ and $S^{i+1}(0)=S\,S^i(0)$. Let $\mathfrak{M}=\langle\mathcal{M},\theta\rangle$ be a standard model of Peano arithmetic, and fix a variable x. Finally, let $\Sigma=\{x\neq S^i(0)\colon n\in\mathbb{N}\}$ and consider the theory T which extends Peano arithmetic with Σ.

If $\Xi\subseteq T$ is finite, $m=\max(\{0\}\cup\{n\colon (x\neq S^n(0))\in\Xi\})$ is defined. Posing $\theta'(x)=m+1$ and $\theta'(y)=\theta(x)$ for all the variables $y\neq x$, it is clear that $\langle\mathcal{M},\theta'\rangle$ is model for Ξ.

Then T has a model $\mathfrak{N}$ by the Compactness Theorem 12.1, and $\mathfrak{N}$ is also a model for Peano arithmetic. Suppose there is an isomorphism $\tau\colon\mathfrak{M}\to\mathfrak{N}$. Since $[\![x]\!]_{\mathfrak{M}}=n$ for some $n\in\mathbb{N}$, then

$$[\![S^n(0)]\!]_{\mathfrak{N}}=\tau([\![S^n(0)]\!]_{\mathfrak{M}})=\tau([\![x]\!]_{\mathfrak{M}})=[\![x]\!]_{\mathfrak{N}}\ ,$$

thus $[\![x=S^n(0)]\!]_{\mathfrak{N}}=1$ which is impossible. □

Discussion

The existence of a non-standard model for Peano arithmetic shows that this theory does not describe **exactly** the natural numbers and their properties which can be expressed in the language. Here, not exactly means not only.

The first thought is to try to *complete* Peano arithmetic to prevent the construction of a model like the $\langle\mathcal{N},\sigma\rangle$ above. Clearly, the shape of the proof using the Compactness Theorem, does not allow to obtain this result in a direct way.

However, it is not evident whether the existence of a non-standard model is disturbing: we cannot use the proof of Proposition 26.2 to write a formula which holds in the non-standard model while it does not in any standard model. Indeed we used this property to synthesise the non-standard model from the standard one.

Discussion

Of course, we can use a theory to separate the non-standard model from any standard one: this is exactly the purpose of the Σ theory in Proposition 26.2.

But still it is not clear whether there is *closed* formula, i.e., a formula with no free variables, allowing to separate standard models from non-standard ones.

This would be crucial since such a sentence ϕ does not depend on the evaluation of variables, thus its truth variable would be defined by the structure of the model only. Thus ϕ, if it exists, cannot be provable even if it is true in any standard model because it would be false in some non-standard model, so by the Soundness Theorem it cannot be proved.

If such a ϕ exists, it means that we have a way to separate models **within** the theory of Peano arithmetic just by adding an axiom ϕ, or its complement $\neg\phi$.

References

Peano arithmetic is illustrated in most textbooks about logic.

The existence of non-standard models can be shown in many different ways. Proposition 26.2 is adapted from *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977).

Mathematical Logic

Lecture 27

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Limiting results:

- Representable entities

Representable entities

Definition 27.1 (Numerals)

Given $n \in \mathbb{N}$ the *numeral* $\overline{n}$ representing n is defined as $\overline{0} \equiv 0$ and $\overline{n+1} \equiv S\ \overline{n}$.

Definition 27.2 (Representation)

A relation $R \subseteq \mathbb{N}^k$ is *representable* in Peano arithmetic if and only if there is a formula ϕ such that

- if $(n_1, \ldots, n_k) \in R$ then $\vdash_{\mathsf{PA}} \phi(\overline{n_1}, \ldots, \overline{n_k})$;
- if $(n_1, \ldots, n_k) \notin R$ then $\vdash_{\mathsf{PA}} \neg\phi(\overline{n_1}, \ldots, \overline{n_k})$;

where $\vdash_{\mathsf{PA}}$ means 'provable in Peano arithmetic'.
A function $f : \mathbb{N}^k \to \mathbb{N}$ is *representable* in Peano arithmetic if the relation $R = \{(n_1, \ldots, n_k, m) : m = f(n_1, \ldots, n_k)\}$ is representable.
A set $S \subset \mathbb{N}$ is *representable* in Peano arithmetic if it is so when interpreted as a unary relation.

Representable entities

Proposition 27.3

If the relation $P, Q \subseteq \mathbb{N}^k$ are representable in Peano arithmetic, so are $\neg P$, $P \wedge Q$, and $P \vee Q$.

Proof.

Since P and Q are representable there are ϕ_P and ϕ_Q as in Definition 27.2. So $(n_1,\ldots,n_k) \in \neg P$ if and only if $(n_1,\ldots,n_k) \notin P$. Thus $\neg\phi_P$ represents $\neg P$ because $\neg\neg\phi_P(n_1,\ldots,n_k) = \phi_P(n_1,\ldots,n_k)$.
Also $(n_1,\ldots,n_k) \in P \wedge Q$ if and only if $(n_1,\ldots,n_k) \in P$ and $(n_1,\ldots,n_k) \in Q$. Thus $\phi_{P\wedge Q} = \phi_P \wedge \phi_Q$. Similarly $\phi_{P\vee Q} = \phi_P \vee \phi_Q$. □

Representable entities

Proposition 27.4

The $\underline{0}$ function is representable.

Proof.

Since $\underline{0}\colon \mathbb{N} \to \mathbb{N}$ we have to find a formula representing $Z = \{(n,m)\colon m = \underline{0}(n)\}$. Consider $\phi_{\underline{0}}(x,y) \equiv (y = 0)$.

- If $(n,m) \in Z$ then $m = \underline{0}(n)$, so $m = 0$. Thus $\phi_{\underline{0}}(\overline{n},\overline{m}) \equiv (\overline{m} = \overline{0}) \equiv (\overline{0} = \overline{0})$, so $\vdash_{\mathrm{PA}} \phi_{\underline{0}}(\overline{n},\overline{m})$ by reflexivity.
- If $(n,m) \notin Z$ then $m \neq \underline{0}(n)$, so $m \neq 0$. Thus $\overline{m} \equiv S\ \overline{m'}$ and $\phi_{\underline{0}}(\overline{n},\overline{m}) \equiv (\overline{m} = \overline{0}) \equiv (S\ \overline{m'} = \overline{0})$, so $\vdash_{\mathrm{PA}} \neg\phi_{\underline{0}}(\overline{n},\overline{m})$ from the axioms. □

Representable entities

Proposition 27.5

The successor function is representable.

Proof.

The formula $y = x + 1$ represents the successor function $S(x) = y$. □

Proposition 27.6

The projection functions are representable.

Proof.

The formula $y = x_i$ represents the projection $U^k_i(x_1,\ldots,x_k) = y$. □

Representable entities

Proposition 27.7

If g and $h_0,\ldots,h_k$ are representable, so is f obtained by substitution:

$$f(x_1\ldots,x_m) = g\,(h_0(x_1\ldots,x_m),\ldots,h_k(x_1\ldots,x_m))\ .$$

Proof.

Let ϕ_g and $\phi_{h_0},\ldots,\phi_{h_k}$ be the formulæ representing $g(z_1,\ldots,z_k) = y$ and $h_0(x_1,\ldots,x_m) = y,\ldots,h_k(x_1,\ldots,x_m) = y$ respectively, making explicit the link between variables, arguments and results. Then the formula

$$\begin{aligned}\phi_f \equiv \exists z_0,\ldots,z_k.\quad & \phi_{h_0}[z_0/y]\\ & \wedge\cdots\\ & \wedge\phi_{h_k}[z_k/y]\\ & \wedge\phi_g\end{aligned}$$

represents $f(x_1,\ldots,x_m) = y$. □

Representable entities

Proposition 27.8

If g is representable, so is f obtained by minimalisation:

$$f(x_1,\ldots,x_k) = \mu m.\, g(x_1,\ldots,x_k,m) = 0 \ .$$

Proof.

Let ϕ_g be the formula representing $g(x_1,\ldots,x_k,m) = z$.
Then $f(x_1,\ldots,x_k) = y$ is represented by

$$\phi_g[y/m,0/z] \wedge \forall m.\, m < y \supset \neg\phi_g[0/z] \ . \qquad \square$$

Proposition 27.9

Addition, multiplication and the relation equal to 0 are all representable.

Proof.

Clearly $x+y=z$ and $xy=z$ represent addition and multiplication.
Also $x=0$ represents the relation equal to 0. $\square$

Representable entities

Fact 27.10

The pairing function $\langle x,y\rangle = (x+y)(x+y+1)/2+x$ is representable. So are its projections π_1 and π_2.

Proof.

The pairing function is represented by $2z = (x+y)(x+y+1)+2x$; $\pi_1(z) = x$ is represented by $\exists y.\, 2z = (x+y)(x+y+1)+2x$ and $\pi_2(z) = y$ is represented by $\exists x.\, 2z = (x+y)(x+y+1)+2x$. All these functions are primitive recursive. $\square$

Fact 27.11

The function $\mathrm{rem}(x,y) = z$ with z the remainder of y/x is representable.

Proof.

It is represented by $\exists d.\, y = dx+z \wedge \exists e.\, e \neq 0 \wedge x = z+e$. $\square$

Representable entities

To show that a function constructed by primitive recursion is representable we need a few preliminary results.

Proposition 27.12 (Bézout identity)

Let $a,b \in \mathbb{N}$, be positive.
If $\gcd(a,b) = 1$ then there are $x,y \in \mathbb{N}$ such that $ax = by+1$.

Proof. (i)

Let $S = \{ax+by : x,y \in \mathbb{Z} \wedge ax+by > 0\}$. Observe that $a \in S$ for $x=1, y=0$, and $b \in S$ for $x=0, y=1$, and each element of S lies in $\mathbb{N}$.

Since $\emptyset \subset S \subseteq \mathbb{N}$, there is $d = \min S$, thus $d \in S$ so $d = au+bv$.

Since $d \leq a$, $a = dq+r$ for some $q,r \in \mathbb{N}$ with $0 \leq r < d$.
Thus $r = a - dq = a(1-qu)+b(-qv)$.
Suppose $r > 0$: then $r \in S$ so $d \leq r$, impossible. Hence $r = 0$, i.e., d divides a.
Analogously, d divides b.

Hence $0 < d \leq \gcd(a,b) = 1$ and $d \in \mathbb{N}$, so $d = 1$. $\hookrightarrow$

Representable entities

$\hookrightarrow$ **Proof. (ii)**

Then $au+bv = 1$, that is $au = b(-v)+1$ with $u,z \in \mathbb{Z}$.

Let $u = bp+s$ for some $p,s \in \mathbb{Z}$ with $0 \leq s < b$.

Pose $x = u-(p-1)b$ and $y = -v-(p-1)a$. Clearly $x,y \in \mathbb{Z}$.

Then $ax = au-(p-1)ab = b(-v)+1-(p-1)ab = by+1$.

Also, $x = bp+s-bp+b = b+s > 0$ so $x \in \mathbb{N}$.
Moreover, $ax = by+1 > 0$ thus $by \geq 0$, but $b > 0$ so $y \geq 0$, i.e., $y \in \mathbb{N}$. $\square$

Representable entities

Theorem 27.13 (Chinese remainder)

Let $x_0, \ldots, x_k \in \mathbb{N}$ be positive and pairwise coprime, i.e., $\gcd(x_i, x_j) = 1$ when $i \neq j$. Then for every $a_0, \ldots, a_k \in \mathbb{N}$ there is $d_0 \in \mathbb{N}$ such that $d_0 \equiv a_i \pmod{x_i}$ for every $0 \leq i \leq k$.

Proof.

Let $N = \prod_{i=0}^{k} x_i$ and $N_i = N/x_i$, $0 \leq i \leq k$, so $N, N_i \in \mathbb{N}$.
Observe that $\gcd(N_i, x_i) = 1$ so there are $p_i, q_i \in \mathbb{N}$ such that $q_i N_i = p_i x_i + 1$ by Bézout identity.

Define $e_i = q_i N_i$. Then $e_i \equiv 1 \pmod{x_i}$ and $e_i \equiv 0 \pmod{x_j}$ for $i \neq j$.

Let $d_0 = \sum_{i=0}^{k} e_i a_i$. Hence

$$d_0 = \left(e_i a_i + \sum_{j=0, i \neq j}^{k} e_j a_j \right) \equiv \left(1 a_i + \sum_{j=0, i \neq j}^{k} 0 a_j \right) = a_i \pmod{x_i} \ . \qquad \square$$

Representable entities

Fix $a_0, \ldots, a_k$. Let $m = \max\{k, a_0, \ldots, a_k\} + 1$ and $x_i = 1 + (1+i)m!$, $0 \leq i \leq k$.

Proposition 27.14

$x_0, \ldots, x_k$ are pairwise coprime.

Proof.

Suppose $p > 0$ divides x_i and x_j, $i \neq j$. Then p divides $x_i - x_j$.
Unfolding the definition of x_i, x_j, p has to divide $(i-j)m!$.
Since p divides x_i, it holds $(1+i)m! \equiv -1 \pmod{p}$, thus p does not divide $m!$, which implies $p > m$.
Hence p has to divide $i - j$, that is $i \equiv j \pmod{p}$. However, $0 \leq i \leq k < m < p$ and $0 \leq j \leq k < m < p$, hence $i = j$, contradiction. $\square$

Fact 27.15

For each $0 \leq i \leq k$, $a_i < x_i$.

Proof.

$a_i < m \leq m! < x_i$. $\square$

Representable entities

Definition 27.16 (β function)

Define $\beta^*(d_0, d_1, i) = \mathrm{rem}(1 + (i+1)d_1, d_0)$ and $\beta(d, i) = \beta^*(\pi_1(d), \pi_2(d), i)$.

Fact 27.17

The functions β^* and β are both representable and primitive recursive.

Representable entities

Proposition 27.18

For every sequence $a_0, \ldots, a_k \in \mathbb{N}$ there is a value d such that $\beta(d, i) = a_i$ for each $0 \leq i \leq k$.

Proof.

As before let $m = \max\{k, a_0, \ldots, a_k\} + 1$ and $x_i = 1 + (1+i)m!$, $0 \leq i \leq k$.
Then $x_0, \ldots, x_k$ are pairwise coprime and all bigger than $a_0, \ldots, a_k$.

Hence by Theorem 27.13 there is $d_0 \in \mathbb{N}$ such that $d_0 \equiv a_i \pmod{x_i}$ for every $0 \leq i \leq k$. Since $x_i > a_i$, $\mathrm{rem}(x_i, a_i) = a_i = \mathrm{rem}(x_i, d_0)$.

Pose $d = \langle d_0, m! \rangle$.
Then $\beta(d, i) = \beta^*(d_0, m!, i) = \mathrm{rem}(1 + (i+1)m!, d_0) = \mathrm{rem}(x_i, d_0) = a_i$. $\square$

Representable entities

Proposition 27.19

If g and h are representable, so is f constructed by primitive recursion:
$f(x_1,\ldots,x_k,0) = g(x_1,\ldots,x_k)$ *and*
$f(x_1,\ldots,x_k,m+1) = h(x_1,\ldots,x_k,m,f(x_1,\ldots,x_k,m))$.

Proof.

If ϕ_g represents $g(x_1,\ldots,x_n) = y$ and ϕ_h represents $h(x_1,\ldots,x_n,m,q) = y$ then we can easily represent $f(x_1,\ldots,x_n,y) = z$ if we suppose there is a sequence $p_0,\ldots,p_y$ such that

$$p_y = z \wedge \phi_g[p_0/y] \wedge \forall i.0 \le i < y \supset \phi_h[i/m, p_i/q, p_{i+1}/y] \ .$$

This requirement, which is **not** a representation formula by itself, can be expressed as a proper formula in Peano arithmetic using the β function:

$$\begin{aligned} \exists p.\quad & \beta(p,y) = z \\ & \wedge \phi_g[\beta(p,0)/y] \\ & \wedge \forall i.0 \le i < y \supset \phi_h[i/m, \beta(p,i)/q, \beta(p,i+1)/y] \ . \end{aligned}$$

□

Representable entities

Theorem 27.20

All recursive functions are representable in Peano arithmetic.

Proof.

Immediate consequence of Propositions 27.4, 27.5, 27.6, 27.7, 27.8, and 27.19. □

Corollary 27.21

All recursive sets and relations are representable in Peano arithmetic.

Note that it is a constructive proof: given a partial recursive function f it provides an effective method to build a formula representing f.

References

The representation of relations, sets, and functions is taken from *Barry Cooper*, Computability Theory, Chapman & Hall/CRC Mathematics, (2004).

The proof of Theorem 27.20 has been adapted from *Elliott Mendelson*, Introduction to Mathematical Logic, CRC Press.

Mathematical Logic

Lecture 28

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Limiting results:

- Gödel's First Incompleteness Theorem
- The idea behind the proof
- Coding terms
- Coding formulæ

Incompleteness theorem

Theorem 28.1 (Gödel's Incompleteness Theorem)

Let T be an effective theory which is consistent, and able to represent all the recursive functions. Then there is a closed formula G such that

$$T \nvdash G \text{ and } T \nvdash \neg G \ .$$

A theory is said to be *effective* when the set of axioms is recursive, that is applying a *coding* to its axioms so that they become a set of numbers, this set is recursive.

A coding of Peano arithmetic is a total map g from the expressions of the syntax (terms, formulæ, proofs) to $\mathbb{N}$ such that

- g is injective;
- g is recursive;
- g^{-1} on the image of g is recursive too.

Strategy

The proof of the incompleteness theorem is complex. It has a difficult part, the fixed point lemma and a lot of technicalities.

The strategy is to consider the sentence "this sentence is not provable".

- we will show that there is a coding function that maps terms, formulæ, and proofs into natural numbers;
- hence it is possible to write a formula which says "there is a number p which is the code of a proof of the sentence x";
- negating that formula we can express the fact that x is not provable;
- we will show a fixed point theorem saying that there exists a fixed point of the transformation which maps each sentence x to the code of the sentence expressing that x is not provable;
- thus the sentence G becomes the formula stating that x is not provable with x substituted with the fixed point;
- the meaning of G is that G is not provable;
- but G must be true in the standard model otherwise the theory would be contradictory, so the result follows.

Coding terms

In the following for the sake of simplicity, we will assume the set of variables in the language of Peano arithmetic to be $V = \{x_i : i \in \mathbb{N}\}$.

Definition 28.2 (Coding terms)

The *Gödel's coding function* g on terms is inductively defined as follows:

- $g(0) = 2 \cdot 3$;
- $g(x_i) = 2 \cdot 3^2 \cdot 5^{i+1}$;
- $g(S\,t) = 2 \cdot 3^3 \cdot 5^{g(t)}$;
- $g(t+s) = 2 \cdot 3^4 \cdot 5^{g(t)} \cdot 7^{g(s)}$;
- $g(t \cdot s) = 2 \cdot 3^5 \cdot 5^{g(t)} \cdot 7^{g(s)}$.

Thanks to the theorem saying that natural numbers admit a unique factorisation in primes, g is computable, injective, and g^{-1} is computable.

Coding terms

A few remarks are needed:

- each code for a term is of the form $2 \cdot n$ with n odd;
- the exponent of the factor 3 tells whether the term is 0, a variable, a successor, a sum, or a multiplication;
- the parameters of a term, i.e., the index of the variable, or the arguments of the successor, of the sum, or the multiplication, are the exponents of the factors 5 and 7, in that order.

Hence, it is possible to write a formula in Peano arithmetic that tells whether its argument is a code of a term by Proposition 27.21. Observe how, by the same result, one can write representations of the functions extracting the various pieces of information about a coded term.

Coding formulæ

Definition 28.3 (Coding formulæ)

The *Gödel's coding function* g on formulæ extends the coding of terms and it is inductively defined as follows:

- $g(\top) = 2^2 \cdot 3$;
- $g(\bot) = 2^2 \cdot 3^2$;
- $g(t = s) = 2^2 \cdot 3^3 \cdot 5^{g(t)} \cdot 7^{g(s)}$;
- $g(\neg A) = 2^2 \cdot 3^4 \cdot 5^{g(A)}$;
- $g(A \wedge B) = 2^2 \cdot 3^5 \cdot 5^{g(A)} \cdot 7^{g(B)}$;
- $g(A \vee B) = 2^2 \cdot 3^6 \cdot 5^{g(A)} \cdot 7^{g(B)}$;
- $g(A \supset B) = 2^2 \cdot 3^7 \cdot 5^{g(A)} \cdot 7^{g(B)}$;
- $g(\forall x.A) = 2^2 \cdot 3^8 \cdot 5^{g(A)} \cdot 7^{g(x)}$;
- $g(\exists x.A) = 2^2 \cdot 3^9 \cdot 5^{g(A)} \cdot 7^{g(x)}$.

Again the coding g is computable, injective, and g^{-1} is computable too.

Coding formulæ

A few remarks are needed:

- each code for a formula is of the form $2^2 \cdot n$ with n odd, so we can separate the codes of terms from the ones of formulæ just looking the exponent of the factor 2;
- the exponent of the factor 3 tells which kind of formula the code represents;
- the parameters of a formula are the exponents of the factors 5 and 7, in that order.

Hence, it is possible to write a formula in Peano arithmetic that tells whether its argument is a code of a formula, and formulæ to tell the various pieces of information about a given coded formula.

Coding sequences

Definition 28.4 (Coding finite sequences)

The *Gödel's coding function* g of a finite sequence $n_1, \ldots, n_k$ of natural numbers is $g(n_1, \ldots, n_k) = 2^3 \cdot \prod_{1 \leq i \leq k} p_{i+1}^{n_i+1}$ with p_j the j-th prime number.

It is clear that the coding function is injective, computable, and its inverse is computable too. Also the codes for sequences can be separated by the codes of terms and formulæ, and the set of codes for sequences can be represented in the sense of Proposition 27.21 by some formula of Peano arithmetic.

Coding proofs

Definition 28.5 (Coding proofs)

The *Gödel's coding function* g on proofs extends the previous coding g and it is inductively defined as:

- if A is a proof by assumption $g(A) = 2^4 \cdot 3^1 \cdot \cdot 13^{g(A)} \cdot 19^{g((A))}$ with $g((A))$ the code of the sequence of one element, A;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash A \quad \pi_2 : \Gamma \vdash B}{A \wedge B}\,\wedge\mathsf{I}\right) = 2^4 \cdot 3^2 \cdot 5^{g(\pi_1)} \cdot 7^{g(\pi_2)} \cdot 13^{g(A \wedge B)} \cdot 19^{g(\Gamma)}$ with $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ and $g(\Gamma) = g(g(\gamma_1), \ldots, g(\gamma_n))$;
- $g\left(\dfrac{\pi : \Gamma \vdash A \wedge B}{A}\,\wedge\mathsf{E}_1\right) = 2^4 \cdot 3^3 \cdot 5^{g(\pi)} \cdot 13^{g(A)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash A \wedge B}{B}\,\wedge\mathsf{E}_2\right) = 2^4 \cdot 3^4 \cdot 5^{g(\pi)} \cdot 13^{g(B)} \cdot 19^{g(\Gamma)}$; ↪

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{\pi : \Gamma \vdash A}{A \vee B}\,\vee\mathsf{I}_1\right) = 2^4 \cdot 3^5 \cdot 5^{g(\pi)} \cdot 13^{g(A \vee B)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash B}{A \vee B}\,\vee\mathsf{I}_2\right) = 2^4 \cdot 3^6 \cdot 5^{g(\pi)} \cdot 13^{g(A \vee B)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash A \vee B \quad \pi_2 : \Gamma, A \vdash C \quad \pi_3 : \Gamma, B \vdash C}{C}\,\vee\mathsf{E}\right) =$ $2^4 \cdot 3^7 \cdot 5^{g(\pi_1)} \cdot 7^{g(\pi_2)} \cdot 11^{g(\pi_3)} \cdot 13^{g(C)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi : \Gamma, A \vdash B}{A \supset B}\,\supset\mathsf{I}\right) = 2^4 \cdot 3^8 \cdot 5^{g(\pi)} \cdot 13^{g(A \supset B)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash A \supset B \quad \pi_2 : \Gamma \vdash A}{B}\,\supset\mathsf{E}\right) = 2^4 \cdot 3^9 \cdot 5^{g(\pi_1)} \cdot 7^{g(\pi_2)} \cdot 13^{g(B)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi : \Gamma, A \vdash \bot}{\neg A}\,\neg\mathsf{I}\right) = 2^4 \cdot 3^{10} \cdot 5^{g(\pi)} \cdot 13^{g(\neg A)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash \neg A \quad \pi_2 : \Gamma \vdash A}{\bot}\,\neg\mathsf{E}\right) = 2^4 \cdot 3^{11} \cdot 5^{g(\pi_1)} \cdot 7^{g(\pi_2)} \cdot 13^{g(\bot)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{}{\top}\,\top\mathsf{I}\right) = 2^4 \cdot 3^{12} \cdot 13^{g(\top)} \cdot 19^8$; ↪

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{\pi : \Gamma \vdash \bot}{A}\,\bot\mathsf{E}\right) = 2^4 \cdot 3^{13} \cdot 5^{g(\pi)} \cdot 13^{g(A)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{}{A \vee \neg A}\,\mathsf{lem}\right) = 2^4 \cdot 3^{14} \cdot 13^{g(A \vee \neg A)} \cdot 19^8$;
- $g\left(\dfrac{\pi : \Gamma \vdash A}{\forall x.A}\,\forall\mathsf{I}\right) = 2^4 \cdot 3^{15} \cdot 5^{g(\pi)} \cdot 13^{g(\forall x.A)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash \forall x.A}{A[t/x]}\,\forall\mathsf{E}\right) = 2^4 \cdot 3^{16} \cdot 5^{g(\pi)} \cdot 13^{g(A[t/x])} \cdot 17^{g(t)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash A[t/x]}{\exists x.A}\,\exists\mathsf{I}\right) = 2^4 \cdot 3^{17} \cdot 5^{g(\pi)} \cdot 13^{g(\exists x.A)} \cdot 17^{g(t)} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash \exists x.A \quad \pi_2 : \Gamma, A \vdash B}{B}\,\exists\mathsf{E}\right) =$ $2^4 \cdot 3^{18} \cdot 5^{g(\pi_1)} \cdot 7^{g(\pi_2)} \cdot 13^{g(B)} \cdot 19^{g(\Gamma)}$; ↪

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{}{\forall x.x = x}\,\mathsf{ax}\right) = 2^4 \cdot 3^{19} \cdot 13^{g(\forall x.x=x)} \cdot 19^8$;
- $g\left(\dfrac{}{\forall x,y.x = y \supset y = x}\,\mathsf{ax}\right) = 2^4 \cdot 3^{19} \cdot 13^{g(\forall x,y.x=y \supset y=x)} \cdot 19^8$;
- $g\left(\dfrac{}{\forall x,y,z.x = y \wedge y = z \supset x = z}\,\mathsf{ax}\right) = 2^4 \cdot 3^{19} \cdot 13^{g(\forall x,y,z.x=y \wedge y=z \supset x=z)} \cdot 19^8$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash A[t/x] \quad \pi_2 : \Gamma \vdash t = r}{A[r/x]}\,\mathsf{ax}\right) =$ $2^4 \cdot 3^{19} \cdot 5^{g(\pi_1)} \cdot 7^{g(\pi_2)} \cdot 13^{g(A[r/x])} \cdot 19^{g(\Gamma)}$;
- $g\left(\dfrac{}{\forall x_1, \ldots, x_n.\exists! z.z = f(x_1, \ldots, x_n)}\,\mathsf{ax}\right) =$ $2^4 \cdot 3^{19} \cdot 13^{g(\forall x_1, \ldots, x_n.\exists! z.z = f(x_1, \ldots, x_n))} \cdot 19^8$;
- $g\left(\dfrac{}{\forall x.Sx \neq 0}\,\mathsf{ax}\right) = 2^4 \cdot 3^{20} \cdot 13^{g(\forall x.Sx \neq 0)} \cdot 19^8$; ↪

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{}{\forall x,y.Sx=Sy\supset x=y}\,\text{ax}\right)=2^4\cdot 3^{20}\cdot 13^{g(\forall x,y.Sx=Sy\supset x=y)}\cdot 19^8$;
- $g\left(\dfrac{}{\forall x.0+x=x}\,\text{ax}\right)=2^4\cdot 3^{20}\cdot 13^{g(\forall x.0+x=x)}\cdot 19^8$;
- $g\left(\dfrac{}{\forall x,y.Sx+y=S(x+y)}\,\text{ax}\right)=2^4\cdot 3^{20}\cdot 13^{g(\forall x,y.Sx+y=S(x+y))}\cdot 19^8$;
- $g\left(\dfrac{}{\forall x.0\cdot x=0}\,\text{ax}\right)=2^4\cdot 3^{20}\cdot 13^{g(\forall x.0\cdot x=0)}\cdot 19^8$;
- $g\left(\dfrac{}{\forall x,y.Sx\cdot y=x\cdot y+y)}\,\text{ax}\right)=2^4\cdot 3^{20}\cdot 13^{g(\forall x,y.Sx\cdot y=x\cdot y+y)}\cdot 19^8$;
- $g\left(\dfrac{}{A[0/x]\wedge(\forall x.A\supset A[Sx/x])\supset\forall x.A}\,\text{ax}\right)=$
  $2^4\cdot 3^{20}\cdot 13^{g(A[0/x]\wedge(\forall x.A\supset A[Sx/x])\supset\forall x.A)}\cdot 19^8$.

Coding proofs

Although it is long and tedious to verify, g is injective, computable, and g^{-1} is recursive. Also, the coding function is written down to make easy to tell pieces apart. For example, the code of the conclusion is always the exponent of the 13 factor.

As before, all the function telling apart the pieces of information about a coded proof can be represented in Peano arithmetic, as well as the fact that a number is the code of a proof.

Numeral

Definition 28.6 (Numeral)

The *numeral* $\ulcorner A\urcorner$ of a formula A is defined as $\ulcorner A\urcorner = S^{g(A)}(0)$, that is the code of A written in the syntax of Peano arithmetic.

Similarly, the numeral of a term t is $\ulcorner t\urcorner = S^{g(t)}(0)$;
the numeral of a proof π is $\ulcorner \pi\urcorner = S^{g(\pi)}(0)$;
and the numeral of a sequence is $\ulcorner e_1,\dots,e_n\urcorner = S^{g(e_1,\dots,e_n)}(0)$.

Numerals allow to *internalise* the codes: we can indirectly speak of a formula (term, proof, sequence) by stating a property of its code. As soon as the property does not rely on the value but on the "meaning" of the code, this is a perfectly reasonable way to proceed.

Fixed point lemma

Lemma 28.7 (Fixed point)

Let Ξ be a theory in which every (primitive) recursive function is representable and let A be a formula such that $\mathrm{FV}(A)=\{y\}$.
Then there is a formula δ_A such that $\mathrm{FV}(\delta_A)=\emptyset$ *and* $\vdash_\Xi \delta_A = A[\ulcorner\delta_A\urcorner/y]$.

Proof. (i)

First it is provable in pure logic that

$$\vdash B[k/z]=(\exists z.z=k\wedge B)$$

for every formula B and for every term t of the same sort as z.

Let $\Delta_{\mathscr{F}}$ be the map from formulæ to formulæ defined by

$$\Delta_{\mathscr{F}}(B)\equiv\exists z.z=\ulcorner B\urcorner\wedge B\ .$$

Evidently this function is primitive recursive. ↪

Fixed point lemma

↪ Proof. (ii)

Thus the map $\Delta_{\mathbb{N}}$ defined by

$$\Delta_{\mathbb{N}}(g(B)) = g(\Delta_{\mathscr{F}}(B))$$

is total on the image of g and (primitive) recursive.

By hypothesis there is a formula Δ with $\mathrm{FV}(\Delta) = \{x, y\}$ such that Δ represents the function $\Delta_{\mathbb{N}}$. In particular it is provable that

$$\vdash_{\Xi} (y = \ulcorner \Delta_{\mathbb{N}}(g(B)) \urcorner) = \Delta[\ulcorner B \urcorner / x] \ .$$

With no loss of generality we may define

$$\delta_A \equiv \Delta_{\mathscr{F}}(F)$$

for some formula F to be determined. ↪

Fixed point lemma

↪ Proof. (iii)

$$\begin{aligned}
& A[\ulcorner \delta_A \urcorner / y] \\
\equiv\ & A[\ulcorner \Delta_{\mathscr{F}}(F) \urcorner / y] && \text{(definition of } \delta_A) \\
\equiv\ & A[\ulcorner \Delta_{\mathbb{N}}(g(F)) \urcorner / y] && \text{(definition of } \Delta_{\mathbb{N}}) \\
=\ & \exists y.\, y = \ulcorner \Delta_{\mathbb{N}}(g(F)) \urcorner \wedge A && \text{(avoiding substitution)} \\
=\ & \exists y.\, \Delta[\ulcorner F \urcorner / x] \wedge A && \text{(definition of } \Delta) \\
=\ & \exists x.\, x = \ulcorner F \urcorner \wedge \exists y.\, \Delta \wedge A && \text{(avoiding substitution)}
\end{aligned}$$

Hence posing $F \equiv \exists y.\, \Delta \wedge A$,

$$\begin{aligned}
\equiv\ & \exists x.\, x = \ulcorner F \urcorner \wedge F && \text{(definition of } F) \\
\equiv\ & \Delta_{\mathscr{F}}(F) && \text{(definition of } \Delta_{\mathscr{F}}) \\
\equiv\ & \delta_A && \text{(definition of } \delta_A)
\end{aligned}$$ □

Provability predicate

Definition 28.8 (Provability predicate)

The formula $\mathscr{D}$ with $\mathrm{FV}(\mathscr{D}) = \{x, y\}$ is defined as

$$\mathscr{D} \equiv \mathrm{isConclusion}(y, x) \wedge \mathrm{isHypotheses}(\ulcorner () \urcorner, x) \wedge \mathrm{isProof}(x) \wedge \mathrm{isFormula}(y) \ .$$

The *provability predicate* T is the formula $\exists x.\mathscr{D}$ having $\mathrm{FV}(\mathrm{T}) = \{y\}$.

Clearly $\mathscr{D}[\ulcorner \pi \urcorner / x, \ulcorner A \urcorner / y]$ holds exactly when A is the conclusion of the proof $\pi \colon \vdash A$. Consequently $\mathrm{T}[\ulcorner A \urcorner / y]$ holds when A is provable.

The formulæ isConclusion(x, y), isHypotheses$(\ulcorner () \urcorner, x)$ (with $\ulcorner () \urcorner$ the numeral of the empty sequence), isProof(x), and isFormula(y) in the definition of $\mathscr{D}$ have not been made explicit. Their definitions come from the fact that the collections of proofs and formulæ are recursive, and the functions to tell pieces apart are computable, as already remarked.

Incompleteness theorem

Theorem 28.9 (Gödel's Incompleteness Theorem)

Let T be an effective theory which is consistent and able to represent all the recursive functions. Then there is a closed formula G such that

$$T \nvdash G \text{ and } T \nvdash \neg G \ .$$

Proof.

Consider the formula $\neg \mathrm{T}[x/y]$: applying the fixed point lemma there is G such that $\mathrm{FV}(G) = \emptyset$ and $\vdash G = \neg \mathrm{T}[\ulcorner G \urcorner / y]$.

Assume there is $\pi \colon \vdash G$. Then $\vdash \neg \mathrm{T}[\ulcorner G \urcorner / y]$. But because $\pi \colon \vdash G$ it holds that $\vdash \mathscr{D}[\ulcorner \pi \urcorner / x, \ulcorner G \urcorner / y]$, and thus $\vdash \exists x.\mathscr{D}[\ulcorner G \urcorner / y]$, that is $\vdash \mathrm{T}[\ulcorner G \urcorner / y]$ making the theory non consistent. Hence $\nvdash G$.

Oppositely suppose there is $\pi \colon \vdash \neg G$. Then $\vdash \mathrm{T}[\ulcorner G \urcorner / y]$ by definition of G, so $\vdash \exists x.\mathscr{D}[\ulcorner G \urcorner / y]$. But this means that there exists $\theta \colon \vdash G$ with $x = \ulcorner \theta \urcorner$ since x is interpreted in some number in a standard model. Thus again we get a contradiction. Hence $\nvdash \neg G$. □

References

The original proof of the first incompleteness theorem can be found in *Kurt Gödel*, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I, Monatshefte für Mathematik und Physik 38, 173–198, (1931).

The proof has been generalised and polished by Rosser and we have shown a slightly reworked version of Rosser's result. The reference is *John Barkley Rosser*, Extensions of some theorems of Gödel and Church, Journal of Symbolic Logic 1, 87–91 (1936).

An account can be found in *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977). Nevertheless the lecture has been prepared roughly following some unpublished notes from the course held by Silvio Valentini in 1991.

Mathematical Logic

Lecture 29

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Limiting results:

- Gödel's Second Incompleteness Theorem
- Meaning and consequences

Properties of provability

Proposition 29.1

In Peano arithmetic, $\vdash A$ *if and only if* $\vdash \mathsf{T}[\ulcorner A \urcorner / y]$.

Proof.
Let $\pi \colon \vdash A$. Then $\vdash \mathscr{D}[\ulcorner \pi \urcorner / x, \ulcorner A \urcorner / y]$ by Definition 28.8, thus $\vdash \mathsf{T}[\ulcorner A \urcorner / y]$. Conversely, if $\vdash \mathsf{T}[\ulcorner A \urcorner / y]$ then in the standard model there is number which is the code of a proof $\vdash A$ by Definition 28.8. □

Proposition 29.2

In Peano arithmetic, if $\vdash \mathsf{T}[\ulcorner A \urcorner / y]$ *then* $\vdash \mathsf{T}[\ulcorner \mathsf{T}[\ulcorner A \urcorner / y] \urcorner / y]$.

Proof.
This is just Proposition 29.1 with $A \equiv \mathsf{T}[\ulcorner A \urcorner / y]$. □

Properties of provability

Proposition 29.3

In Peano arithmetic, if $\vdash \mathsf{T}[\ulcorner A \supset B \urcorner / y]$ *then* $\vdash \mathsf{T}[\ulcorner A \urcorner / y] \supset \mathsf{T}[\ulcorner B \urcorner / y]$.

Proof.
Let $\vdash \mathsf{T}[\ulcorner A \supset B \urcorner / y]$, then by Proposition 29.1 $\vdash A \supset B$.
Assume A. Then $A \vdash B$ by implication elimination and $A \vdash \mathsf{T}[\ulcorner B \urcorner / y]$ by Proposition 29.1.
Since $\mathsf{T}[\ulcorner A \urcorner / y]$ is equivalent to A by Proposition 29.1,
$\vdash \mathsf{T}[\ulcorner A \urcorner / y] \supset \mathsf{T}[\ulcorner B \urcorner / y]$ by implication introduction. □

Löb theorem

Theorem 29.4 (Löb)

In Peano arithmetic let θ *be a closed formula.*
Then $\vdash \mathsf{T}[\ulcorner \theta \urcorner / y] \supset \theta$ *if and only if* $\vdash \theta$.

Proof. (i)
If $\vdash \theta$ then $\vdash \mathsf{T}[\ulcorner \theta \urcorner / y] \supset \theta$ obviously.
Conversely assume $\vdash \mathsf{T}[\ulcorner \theta \urcorner / y] \supset \theta$.
By Lemma 28.7 there is a sentence ϕ such that $\vdash \phi = (\mathsf{T}[\ulcorner \phi \urcorner / y] \supset \theta)$. Then

$$\begin{aligned}
&\vdash \phi \supset (\mathsf{T}[\ulcorner \phi \urcorner / y] \supset \theta) \\
\Rightarrow\ &\vdash \mathsf{T}[\ulcorner \phi \supset (\mathsf{T}[\ulcorner \phi \urcorner / y] \supset \theta) \urcorner / y] && \text{(by Proposition 29.1)} \\
\Rightarrow\ &\vdash \mathsf{T}[\ulcorner \phi \urcorner / y] \supset \mathsf{T}[\ulcorner \mathsf{T}[\ulcorner \phi \urcorner / y] \supset \theta \urcorner / y] && \text{(by Proposition 29.3)} \\
\Rightarrow\ &\vdash \mathsf{T}[\ulcorner \phi \urcorner / y] \supset (\mathsf{T}[\ulcorner \mathsf{T}[\ulcorner \phi \urcorner / y] \urcorner / y] \supset \mathsf{T}[\ulcorner \theta \urcorner / y]) && \text{(by Proposition 29.3)}
\end{aligned}$$

By Proposition 29.2 $\vdash \mathsf{T}[\ulcorner \phi \urcorner / y] \supset \mathsf{T}[\ulcorner T[\ulcorner \phi \urcorner / y] \urcorner / y]$. ↪

Löb theorem

↪ **Proof. (ii)**
So

$$\begin{aligned}
&\vdash \phi \supset (\mathsf{T}[\ulcorner \phi \urcorner / y] \supset \theta) \\
\Rightarrow\ &\vdash \mathsf{T}[\ulcorner \phi \urcorner / y] \supset \mathsf{T}[\ulcorner \theta \urcorner / y] \\
\Rightarrow\ &\vdash \mathsf{T}[\ulcorner \phi \urcorner / y] \supset \theta && \text{(by hypothesis)} \\
=\ &\phi && \text{(by definition of } \phi) \\
\Rightarrow\ &\vdash \mathsf{T}[\ulcorner \phi \urcorner / y] && \text{(by Proposition 29.1)}
\end{aligned}$$

Hence $\vdash \theta$ by definition of ϕ. □

Second incompleteness theorem

Theorem 29.5 (Gödel's second incompleteness theorem)

There is no provable formula C *in Peano arithmetic which codes the consistency of the theory, i.e., such that* $\vdash C \supset \neg \mathsf{T}[\ulcorner \bot \urcorner / y]$.

Proof.
Since Peano arithmetic is consistent $\nvdash \bot$.
Then by Theorem 29.4 $\nvdash \mathsf{T}[\ulcorner \bot \urcorner / y] \supset \bot$, i.e. $\nvdash \neg \mathsf{T}[\ulcorner \bot \urcorner / y]$.
If $\vdash C$ then $\vdash \neg \mathsf{T}[\ulcorner \bot \urcorner / y]$, obtaining a contradiction. □

It is important to remark that Löb's theorem and Gödel's second incompleteness theorem can be immediately extended to all the consistent theories able to represent all the computable functions with a provability predicate T for which Propositions 29.1, 29.2, and 29.3 hold.

Mathematical meaning

The incompleteness theorems closes the quest for a universal, self-contained foundation of Mathematics which is able to prove its own consistency. Simply, such a system cannot exist.

Nevertheless these theorems opened the way to many developments, and to some of the fundamental results in XX$^{\text{th}}$ century:

- the effective construction of non-computable functions
- the idea of coding lead to reason "modulo a coding function", which has been greatly influential in algebra, algebraic geometry, algebraic topology, number theory, ...
- examples of independent statements arose in many fields and they shed lights to a variety of hidden aspects of apparently clean notions, like for example the assumptions behind cardinality in set theory.

Foundational consequences

Having a mathematical theory T which is powerful enough to represent Peano arithmetic has the consequence that we cannot prove its consistency within T. We need a theory T' containing T, and more powerful.

This fact led to the development of many hierarchies of formal systems to classify the power of mathematical theories: we scratched just the surface by showing that the consistency of Peano arithmetic can be proved in a stronger system. But how much stronger? Since the proof of Gödel's results much deeper analyses have been conducted, and nowadays this part of Logic is a complex, intricate, difficult field on its own.

In constructive mathematics the same fact led to doubt that "truth" is the right concept to analyse, and there are approaches favouring the notion of provability as the real foundation of Mathematics. This has a number of consequences, which we do not want to discuss here.

Understanding

For a very long time mathematicians regarded the incompleteness theorems as strange beasts: something which is important, but essentially with no influence in the mathematical practise.

For example the textbook of Bell and Machover we referred to many times explicitly says that the sentences which are not provable in Peano arithmetic are not important in arithmetic because they have no "arithmetical" content, but just a logical one. This is true for the sentence G and for most other sentences we can construct within the logical analysis.

Unfortunately there are purely arithmetical properties of genuine interest for mathematicians not working in logic which are independent from Peano arithmetic. And the same holds in other mathematical theories.

References

The original proof of the second incompleteness theorem can be found in *Kurt Gödel*, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I, Monatshefte für Mathematik und Physik 38, 173–198, (1931).

The proof we have shown uses Löb theorem. An account can be found in *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977). Nevertheless the lecture has been prepared roughly following some unpublished notes by Michael Rathjen.

Mathematical Logic

Lecture 30

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Limiting results:

- Kolmogorov complexity
- Chaitin's Incompleteness Theorem

A different incompleteness result

Consider the so-called *Berry's paradox*:

The smallest positive integer not definable in under sixty letters.

Fix a language, say English. The set of English sentences with length at most 59 letters is finite. Thus the sentences defining a number in less than 60 letters are finite.

The set of positive integer is infinite. Thus necessarily there is number which cannot be defined by a short sentence. Since positive integers are well-ordered, there is a minimal one.

Hence, the sentence of the paradox defines exactly that number, which is, by definition undefinable.

The Chaitin's incompleteness theorem formalises this paradox, showing that a sentence like the one of the paradox is non provable.

Kolmogorov complexity

Consider all the finite strings on the $\{0,1\}$ alphabet.

Fix a partial recursive function f, seen as going from $\{0,1\}^*$ to $\{0,1\}^*$.

Finally, fix a string $\sigma \in \{0,1\}^*$: we say that f *generates* σ if there is $\tau \in \{0,1\}^*$ such that $f(\tau) = \sigma$.

Definition 30.1 (relative Kolmogorov complexity)

The *Kolmogorov complexity* of $\sigma \in \{0,1\}^*$ relative to f, a partial recursive function, is

$$\mathcal{K}_f(\sigma) = \min\{|\tau| \colon f(\tau) = \sigma\} \ ,$$

where $|\tau|$ is the length of the string τ, and $\mathcal{K}_f(\sigma) = \infty$ if σ is not in the image of f.

Kolmogorov complexity

If we imagine $f(\tau)$ as a description of σ through f, the Kolmogorov complexity measures the length of the minimal description of σ which f makes available.

We would like to measure the length of the minimal string τ which could generate σ independently of f.
Of course, this concept does not make sense, since the constant function $g(\tau) = \sigma$ clearly generates σ when τ is the empty string.

But it makes sense to ask the minimal *size* of a pair (f, τ) such that $f(\tau) = \sigma$.

Kolmogorov complexity

Fix an acceptable enumeration of all the partial recursive function $\{\phi_i\}_{i\in\omega}$ with a distinguished universal function $U(i,x) = \phi_i(x)$.

Definition 30.2 (Kolmogorov complexity)

The *Kolmogorov complexity* of $\sigma \in \{0,1\}^*$ is

$$\mathscr{K}(\sigma) = \min\{|\tau| \colon U(\tau) = \sigma\} \ .$$

Observe how $\mathscr{K}(\sigma) = \mathscr{K}_U(\sigma)$.
Also, think to τ as a sequence representing a pair (i,x).

Kolmogorov complexity

Theorem 30.3 (Optimality)

For every partial recursive function f there is $c \in \omega$ such that, for every $\sigma \in \{0,1\}^*$, $\mathscr{K}(\sigma) \leq \mathscr{K}_f(\sigma) + c$.

The proof amounts to observe that there is $i \in \omega$ such that $f = \phi_i$, so $U(i,x) = f(x)$. The constant c is then constructed by choosing the i of minimal length for which this happens.

Theorem 30.4 (Invariance)

If U_1 and U_2 are universal functions, there is $c \in \omega$ such that, for every $\sigma \in \{0,1\}^*$, $|\mathscr{K}_{U_1}(\sigma) - \mathscr{K}_{U_2}(\sigma)| \leq c$.

The proof is immediate from Theorem 30.3.

Hence, up to constants, the choice of the universal function does not matter.

Incompressible strings

Note that $\mathscr{K}(\sigma) \leq |\sigma| + c$ for some c independent from σ because the identity function is recursive.

Observe how there are 2^n strings of length n, while there are

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1$$

strings of length less than n.

Hence, for each $n \in \omega$, there is at least one string σ of length n such that

$$\mathscr{K}(\sigma) \geq |\sigma| \ .$$

These strings are called *incompressible*.

Incompressible strings

More in general, fix $k \in \omega$: if $\sigma \in \{0,1\}^*$ satisfies $\mathscr{K}(\sigma) \geq |\sigma| - k$, the σ string is called *k-incompressible*.

Clearly, for each $n \in \omega$, there are at least $2^n - (2^{n-k} - 1)$ k-incompressible strings of length n. Therefore, at least

$$1 - \frac{1}{2^k}$$

strings of length n are k-incompressible.

Observe how the k-incompressible strings of length n *depend* on the particular choice of universal function in the definition of Kolmogorov complexity.

Chaitin's theorem

Let T be an effective theory of arithmetic which represents all the partial recursive functions. Fix a universal function U.
Then there is a formula ψ such that

$$T \vdash \psi(\ulcorner \sigma \urcorner, \ulcorner \tau \urcorner) \text{ if and only if } U(\sigma) = \tau \ .$$

Hence there is a formula $\phi(x, y)$ such that, for any σ and n,

$$\mathscr{K}(\sigma) \geq n \text{ if and only if } \phi(\ulcorner \sigma \urcorner, n) \text{ is true on } \mathbb{N} \ .$$

A theory T as above is said to be *$\mathscr{K}$-sound* if, for every σ and n,

$$T \vdash \phi(\ulcorner \sigma \urcorner, n) \text{ implies } \mathscr{K}(\sigma) \geq n \ .$$

Observe how Peano arithmetic is $\mathscr{K}$-sound.

Chaitin's theorem

Theorem 30.5 (Chaitin's incompleteness)
Let T be an effective $\mathscr{K}$-sound theory representing all the partial recursive functions. Then, there is $N \in \omega$ such that, for every $\sigma \in \{0,1\}^*$,

$$T \nvdash \phi(\ulcorner \sigma \urcorner, N) \ .$$

Proof. (i)
By contradiction, suppose that for every $N \in \omega$ there is $\sigma \in \{0,1\}^*$ such that

$$T \vdash \phi(\ulcorner \sigma \urcorner, N) \ .$$

Since T is effective, there is a recursive function e enumerating all its theorems. And this function is representable by hypothesis.

Observe that the length of a string is a computable function, thus it is representable by hypothesis. ↪

Chaitin's theorem

↪ *Proof.* (ii)
Define the function f on the input τ which takes the first pair $\langle \sigma, k \rangle$ in the enumeration e such that $T \vdash \phi(\ulcorner \sigma \urcorner, k)$ and $k > 2|\tau|$, and outputs σ.

As observed, $k > 2|\tau|$ is computable, and thus enumerating all the theorems of T by e, it suffices to find the minimal index for which $T \vdash \phi(\ulcorner \sigma \urcorner, k)$ and $k > 2|\tau|$, for some $\langle \sigma, k \rangle$.

So, f is computable. But by the hypothesis to contradict, f is also total. Hence, its Kolmogorov complexity is defined.

Let $d \in \omega$ be such that, for every $\sigma \in \{0,1\}^*$,

$$\mathscr{K}(\sigma) \leq \mathscr{K}_f(\sigma) + d \ ,$$

as for Theorem 30.3. ↪

Chaitin's theorem

↪ Proof. (iii)

Fix $\delta \in \{0,1\}^*$ of length d. Clearly, $\mathscr{K}_f(\delta) \leq |\delta| = d$, by definition.

Also, let σ be such that $f(\delta) = \sigma$. Thus, by definition of f, $T \vdash \phi(\ulcorner\sigma\urcorner, k)$ for some $k > 2|\delta| = 2d$.

Hence, by $\mathscr{K}$-soundness, $\mathscr{K}(\sigma) \geq k$.

Putting all together,

$$2d < k \leq \mathscr{K}(\sigma) \leq \mathscr{K}_f(\sigma) + d \leq d + d = 2d \ ,$$

an evident contradiction. □

Discussion

The Chaitin's incompleteness theorem tells that there is barrier N such that all the sufficiently incompressible strings, i.e., those whose Kolmogorov complexity is at least N, cannot be proved to be so much incompressible.

The interesting aspects of the Theorem are:

- It is based on a different paradox, which is not of a logical nature, but rather of an *information-theoretic* nature.
- Incompressible strings are *random* strings in a quite strict sense. This incompleteness result tells that randomness is not a concept that can be fully formalised.
- There is a link between information theory, an essentially probabilistic theory, and limiting results in Logic, a quite unexpected and surprising fact.

References

The original development of Chaitin's incompleteness result can be found in *Gregory Chaitin*, Information-Theoretic Limitations of Formal Systems, Journal of the ACM 21(3), 403–424 (1974).

The background on information theory can be found in *Claude Shannon* and *Warren Weaver*, Mathematical Theory of Communication, University of Illinois Press (1963).

Mathematical Logic

Lecture 31

Dr Marco Benini

marco.benini@uninsubria.it

Dipartimento di Scienza e Alta Tecnologia
Università degli Studi dell'Insubria

a.a. 2023/24

Syllabus

Limiting results:

- Incompleteness and computability
- Natural incompleteness
- Incompleteness in set theory
- Ordinal analysis

Incompleteness and computability

The proof that there are non-computable functions, based on a counting argument, is unsatisfactory. It is correct, but it does not show an example of a non-computable function.

The example we want to show is the so-called *halting problem*: given the code of a program and a value as an input, we ask whether we could always establish that such a program computing on the given value terminates.

Let $\{\phi_i\}_{i\in\mathbb{N}}$ be a reasonable enumeration of all the partial recursive functions.

Consider the function

$$h(i,x) = \begin{cases} 1 & \text{if } \phi_i(x) \text{ is defined} \\ 0 & \text{otherwise} \end{cases}$$

Incompleteness and computability

Let $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ be a total recursive function.
Define

$$g(i) = \begin{cases} 0 & \text{if } f(i,i) = 0 \\ \bot & \text{otherwise} \end{cases}$$

Clearly, g is partial recursive. Thus, there is an index $e \in \mathbb{N}$ such that $\phi_e = g$.

Consider $f(e,e)$:

- if $f(e,e) = 0$, then $g(e) = \phi_e(e) = 0$, thus $h(e,e) = 1$;
- if $f(e,e) \neq 0$, then $g(e) = \phi_e(e) = \bot$, thus $h(e,e) = 0$.

Observe that h is total.

Suppose h is recursive.
Hence, posing $f = h$, we have $h(e,e) = 1$ if and only if $h(e,e) = 0$, getting a contradiction. Therefore h is non-computable.

Incompleteness and computability

Theorem 31.1 (Weak incompleteness)

No theory of arithmetic is consistent, able to prove all the true sentences in the standard model, and such that that there is a recursive function $\mathscr{E} : \mathbb{N} \to \mathbb{N}$ for which $\mathscr{E}(\mathbb{N}) = \{g(A) : T \vdash A \wedge \mathrm{FV}(A) = \emptyset\}$ with g Gödel's coding.

Proof. (i)

Suppose there is a theory T as in the statement.

Let $\{\phi_i\}_{i\in\mathbb{N}}$ a good enumeration of all the partial recursive functions, and let U be universal function in it: $\mathrm{U}(i,x) = \phi_i(x)$. Also let $\mathscr{T}$ be the set of all the true sentences in the standard model.

Since $\mathscr{C} = \{A : T \vdash A \wedge \mathrm{FV}(A) = \emptyset\}$ contains $\mathscr{T}$ by hypothesis, $\mathscr{C} = \mathscr{T}$. Indeed, if $B \in \mathscr{C}$ but $B \notin \mathscr{T}$, then $\neg B \in \mathscr{T}$ because every sentence is either true or false in a model, so $\neg B \in \mathscr{C}$, thus T is not consistent, contradiction. ↪

Incompleteness and computability

↪ Proof. (ii)

Consider the universal function U. It is representable in Peano arithmetic, thus there is formula F with $\mathrm{FV}(F) = \{i, x, y\}$ such that

- if $\mathrm{U}(m,n) = k$ then $\vdash_{\mathrm{PA}} F\left[S^m(0)/i, S^n(0)/x, S^k(0)/y\right]$. Thus by ∃I, $\vdash_{\mathrm{PA}} \exists y.F\left[S^m(0)/i, S^n(0)/x\right]$, and this sentence is true in every model by the Soundness Theorem, in particular in the standard model, thus $(\exists y.F\left[S^m(0)/i, S^n(0)/x\right]) \in \mathscr{T} = \mathscr{C}$. Hence, $T \vdash \exists y.F\left[S^m(0)/i, S^n(0)/x\right]$.
- if $\mathrm{U}(m,n) \neq k$ then $\vdash_{\mathrm{PA}} \neg F\left[S^m(0)/i, S^n(0)/x, S^k(0)/y\right]$. By the Soundness Theorem, this sentence in true in every model, in particular in the standard one, so $(\neg F\left[S^m(0)/i, S^n(0)/x, S^k(0)/y\right]) \in \mathscr{T} = \mathscr{C}$. Hence $T \vdash \neg F\left[S^m(0)/i, S^n(0)/x, S^k(0)/y\right]$.
  When $\mathrm{U}(m,n)$ is undefined then $(\neg F\left[S^m(0)/i, S^n(0)/x, S^k(0)/y\right]) \in \mathscr{T}$ for every $k \in \mathbb{N}$, thus $(\neg\exists y.F\left[S^m(0)/i, S^n(0)/x\right]) \in \mathscr{T} = \mathscr{C}$ by definition of semantics. Hence $T \vdash \neg\exists y.F\left[S^m(0)/i, S^n(0)/x\right]$ by definition of $\mathscr{C}$. ↪

Incompleteness and computability

↪ Proof. (iii)

Hence, when $\phi_m(n)$ is defined, $T \vdash \exists y.F\left[S^m(0)/i, S^n(0)/x\right]$, while when $\phi_m(n)$ is undefined, $T \vdash \neg\exists y.F\left[S^m(0)/i, S^n(0)/x\right]$.
Also exactly one of these sentences lies in $\mathscr{C} = \mathscr{T}$. Thus

$$f(m,n) = \mu k. \quad \begin{array}{l} \mathscr{E}(k) = \ulcorner \exists y.F\left[S^m(0)/i, S^n(0)/x\right] \urcorner \\ \vee\, \mathscr{E}(k) = \ulcorner \neg\exists y.F\left[S^m(0)/i, S^n(0)/x\right] \urcorner \end{array}$$

is recursive and total.

Define

$$h(m,n) = \mathrm{sg}\left(\left|f(m,n) - \ulcorner \neg\exists y.F\left[S^m(0)/i, S^n(0)/x\right] \urcorner\right|\right) \ .$$

Then $h(m,n) = 1$ if $\phi_m(n)$ is defined, while $h(m,n) = 0$ when $\phi_m(n)$ is undefined, and it is recursive.
Thus it solves the Halting Problem, impossible. □

Natural incompleteness

Historically, the first theorem which states a result beyond Logic that is true but non-provable in Peano arithmetic is:

Theorem 31.2 (Paris, Harrington)

For all $e, r, k \in \mathbb{N}$ there is $M \in \mathbb{N}$ such that for every $f : \{F \subseteq \{0,\dots,M\} : |F| = e\} \to \{0,\dots,r\}$ there is $H \subseteq \{0,\dots,M\}$ such that

- $|H| \geq \max\{k, \min H\}$ *and*
- *exists $v \leq r$ such that for all $F \subseteq H$ with $|F| = e$, $f(x) = v$ for each $x \in F$.*

By using the Infinite Ramsey's Theorem it is not too difficult to derive a value $M \in \mathbb{N}$ which makes the statement true on naturals. This proof is carried out either in second-order arithmetic with the full induction principle, or in a suitable set theory, e.g., **ZFC**.

Natural incompleteness

Nevertheless, it is possible to show, *within Peano arithmetic*, that the combinatorial principle in Theorem 31.2 implies the consistency of Peano arithmetic, thus it is impossible to prove in that theory according to Gödel's second incompleteness theorem.

This theorem is *natural* in the sense that changing the first condition in Theorem 31.2 to $|H| \geq k$, we get the Finite Ramsey's Theorem, which is provable inside Peano arithmetic, and which is the starting point for a large branch of Combinatorics.

Natural incompleteness

By the way, the cited Ramseys' Theorem are:

Theorem 31.3 (Ramsey, finite)

Let $c \in \mathbb{N}, c > 1$. Let $n_1, \ldots, n_c \in \mathbb{N}$. Then there is $N \in \mathbb{N}$ such that if the edges of a complete graph $\mathscr{C}$ of order N are coloured with c different colours, then for some $1 \leq i \leq c$, $\mathscr{C}$ must contain a complete subgraph of order n_i whose edges have all colour i.

Theorem 31.4 (Ramsey, infinite)

Let X be an infinite set and colour the subsets of X of size n in c different colours. Then there exists some infinite subset $M \subseteq X$ such that the subsets of size n in M all have the same colour.

Natural incompleteness

Another important theorem from a different branch of combinatorics is independent from Peano arithmetic: it holds in the standard model but we cannot prove it in the theory. This is the famous Kruskal's theorem on trees. A simplified version suffices to yield the independence result.

Theorem 31.5

There is some $n \in \mathbb{N}$ such that if $T_1, \ldots, T_n$ is a finite sequence of trees where T_k has $k + n$ vertices, then for some $i < j$ there is an injective map $f : T_i \to T_j$ between the vertices of the trees which preserves paths.

The independence proof for this theorem follows a different pattern: it is possible to show that any function which provably exists in Peano arithmetic cannot grow too fast, but the above theorem allows to construct a function which grows even faster. And this suffices to establish the fact that the theorem is unprovable in Peano arithmetic.

Natural incompleteness

Definition 31.6 (Well quasi order)

A *quasi order* is a structure $\langle \mathscr{O}; \leq \rangle$ such that $\leq$ is a reflexive and transitive relation over $\mathscr{O}$.

A *well quasi order* is a quasi order such that

- every proper descending chain is finite: a *proper descending chain* is a sequence $\{e_i\}_i$ in $\mathscr{O}$ such that $e_i < e_j$ when $j < i$;
- every antichain is finite: an *antichain* is a subset $A \subseteq \mathscr{O}$ such that, if $a, b \in A$ and $a \neq b$, then $a \not\leq b$ and $b \not\leq a$.

Natural incompleteness

Kruskal's Theorem admits a simple and useful generalisation:

Theorem 31.7 (Kruskal)

The set of all finite trees with the embedding relation is a well quasi order.

The embedding relation is defined as: $T \leq S$ if and only if there is an injective function f from the nodes of T to the nodes of S which preserves paths, that is, if there is a path from a to b in T, then there is a path from $f(a)$ to $f(b)$ in the S tree.

Natural incompleteness

Definition 31.8 (Graph minor)

Let $\mathscr{G}$ and $\mathscr{H}$ be two finite undirected graphs. Then $\mathscr{H}$ is a *minor* of $\mathscr{G}$ if and only if there is an equivalence relation $\sim$ on the nodes of $\mathscr{G}$ and an injective function from the nodes of $\mathscr{H}$ to the nodes of $\mathscr{G}$ such that

- if $a \sim b$ then there is a path from a to b;
- if (a, b) is an arc in $\mathscr{H}$, then there are two nodes c and d in $\mathscr{G}$ such that $c \not\sim d$, $f(a) \sim c$, $f(b) \sim d$ and (c, d) is an arc in $\mathscr{G}$.

The idea behind the definition is that we can partition the nodes of $\mathscr{G}$ in connected subsets, and from these subsets we can construct a quotient graph $\mathscr{G}/\sim$ whose nodes are the subsets, and whose edges are the arcs between nodes in distinct subsets. Hence, $\mathscr{H} \leq \mathscr{G}$ when $\mathscr{H}$ is a subgraph of $\mathscr{G}/\sim$.

Natural incompleteness

Theorem 31.9 (Graph Minor)

The set of finite undirected graphs together with the graph minor relation forms a well quasi order.

This theorem whose proof is one of the major achievements in the XX$^{\text{th}}$ century Mathematics, is easily shown to be unprovable in Peano Arithmetic since it allows to derive Kruskal's Theorem.

Incompleteness in set theory

We have already discussed how the Axiom of Choice, the Continuum Hypothesis, and the Generalised Continuum Hypothesis are independent from **ZF**. All these statements are "natural" as they state properties of sets which are inherently of interest, either because of their consequences, or because they impose a regular structure over the objects we want to study.

Indeed the independence results in set theory and in Peano arithmetic are related. For example Theorem 31.2 is a restriction to the finite case of the proof of independence about the existence of large cardinals.

Ordinal analysis

There is a branch of proof theory, called *ordinal analysis*, devoted to study the "power" of deductive systems showing which is the minimal ordinal to which transfinite induction can be relativised to prove a consistency statement.

This is a deep, delicate, and difficult part of logic, still in development: it is sometimes referred to as "reverse mathematics" when the goal is to find the minimal theory in which a given theorem can be shown to hold.

References

The discussion is general, and there is no specific reference for it. Some ideas could be found in *Jon Barwise*, Handbook of Mathematical Logic, Studies in Logic and the Foundations of Mathematics 90, North-Holland, (1977).

As an example of a (very) popular book which deals with incompleteness, we signal *Dougls Hofstadter*, Gödel, Escher, Bach: an Eternal Golden Braid, Basic books, (1979). It is an enjoyable account for non-specialists but it also contains many debatable points and opinions. Nevertheless, the mathematical content is, essentially, precise—and the author won the Pulitzer prize for non-fiction.

References

A dated, but still valid reference for Ramsey theory is *Ronald L. Graham, Bruce L. Rothschild, Joel H. Spencer*, Ramsey Theory, 2nd edition, John Wiley and Sons, (1990).

The original paper *Joseph Bernard Kruskal*, Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture, Transactions of the American Mathematical Society 95(2), pp. 210–225, American Mathematical Society, (1960) is an inspiring introduction to the theorem and its motivation.

Although there are many texts providing a general overview of combinatorics, my preferred one is *Miklós Bóna*, A Walk Through Combinatorics, 2nd edition, World Scientific, (2006).

References

The link between Kruskal's theorem and logic is analysed in depth in *Jean Henri Gallier*, What's so special about Kruskal's theorem and the ordinal Γ_0? A survey of some results in proof theory, Annals of Pure and Applied Logic 53(3), pp. 199-260, (1991).

The original publication about the Paris-Harrington theorem can be found in *Jon Barwise*, Handbook of Mathematical Logic, Studies in Logic and the Foundations of Mathematics 90, North-Holland, (1977).

Finally a fine introduction to ordinal analysis can be found in *Michael Rathjen*, The art of ordinal analysis, Proceedings of the International Congress of Mathematicians, volume 2, pp. 45–70, (2006), written by a master of the field.

The end

©Marco Benini, Patio in the forest, Seoul