Mathematical Logic

Lecture 1

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Bureaucracy:

- Introduction
- Program
- Examination
- Timing
- Questions

Mathematical logic:

- Classical logic
- Propositional logic

Introduction

Mathematical logic is a subfield of Mathematics which studies the deduction process, and the foundations of the whole discipline.

This course will introduce mathematical logic from the very beginning, assuming a minimal knowledge of elementary mathematics.

Also, the material of the course is, more or less, standard, and most introductory textbooks will cover it. For the purposes of this course, slides and lecture notes will be made available to students after every lesson.

The course is in English.

Program

The course takes 64 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;
- *Set theory*: fundamental axioms, ordinals, cardinals, transfinite induction, axiom of choice, continuum hypothesis;
- *Constructive mathematics*: intuitionistic logic, computable functions, λ calculi, propositions as types;
- *Limiting results*: Peano arithmetic, Gödel's incompleteness theorems, natural incompleteness results, incompleteness and computability.

Texts

Since this is the first year this course is taught, there is no textbook available in advance. A draft of the textbook is available on the course website and, from time to time, it will grow as the lessons will proceed.

All the slides, along with the lecture notes will become available roughly after each lesson at the course website:

`http://marcobenini.wordpress.com/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 available. While the content of slides is *mandatory*, looking at references is *optional*. Also, the lecture notes will provide the same material as the slides, eventually complemented with exercises: while it is not mandatory to study on the lecture notes, they could be a big aid.

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 dates scheduled in the official calendar: 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 beginning of the course.

As usual, independently from the results, repeating an examination cancels the previous ones.

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, so that students can move between classrooms.

There are no pauses during the lessons.

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 the ask questions by email: in case write at the address

`marco.benini@uninsubria.it`

specifying your name, the course, and the question. If possible, try to 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, when I am not teaching, it is often the case that I am not in Italy, so, please, use this as your last resource.

Mathematical Logic

Logic is formal

Consider arithmetic as a guiding example. When expressing this theory in logical terms, you will have three main levels to look at it:

- syntax
- semantics
- intended interpretation

Logic keeps the intended interpretation in the background, and it focuses on the study of syntax and semantics.

Also, the syntax and the semantics are *formal*: although this could be boring, and, in some cases, a burden to get to results, it is also the fundamental tool of logic. If you don't like it, well, you are in the wrong place!

Syntax

The *syntax* is the way we write down things.

For example, $1 + 2$ is an *expression*, and $1 + 2 = 5$ is a formula. The *language* of a theory, e.g., arithmetic, is the collection of rules allowing to write all the possible expressions and formulae.

Also, since we are interested in proving theorems, which are formulae, eventually depending on other formulae, the hypotheses, we need a way to write proofs. The way to construct proofs is, again, formal, and it is described by a *deductive system*, a collection of axioms and rules.

Together, the language and the deductive system form the *syntax* of a theory. Syntactical reasoning is **the** way to think inside a logical theory, the only one which can be studied.

Intended interpretation

The intended interpretation of a theory is the informal, intuitive way to understand a (logical) theory.

For example, when we say that “arithmetic studies the properties of integer numbers”, we should read this sentence as “the formal theory of arithmetic, that is, its syntax, has the properties of integers as its intended interpretation”, and we assume to know what does it mean to be a property, and what is the shape of integers.

In mathematical logic, we keep the intended interpretation in the background: we are interested in a syntax which allows to express what we intend, e.g., by “property” or by “integer”, and we are interested in a formal way to say when a formula is true, which should correspond to a property being valid in the intended interpretation.

Semantics

The semantics is the formal way to attribute a meaning to a given syntax.

We are interested in semantic systems which, in some sense, capture the intended meaning of our theories. For example, in arithmetic, we would like a semantics that says that $1+2=3$ is a true formula, while $1+2=5$ is a false formula.

Usually, a semantics, defines a universe, where expressions are interpreted, and a notion of truth and falsity, which are used to distinguish between valid and invalid properties.

We will see many examples of semantics, in this course, and we will see that a *good* agreement between the syntax and the semantics is what we will constantly search for.

Syntactical classifications

Logical theories may be classified in many ways. One common criterion is to use the syntax. Specifically, we require a number of logical connectives and quantifiers to be present in the language.

The fundamental connectives of logic are $\wedge$ (and), $\vee$ (or), $\supset$ (if...then...), $\neg$ (not), $\top$ (true), and $\bot$ (false). The fundamental quantifiers are $\forall$ (for all), and $\exists$ (exists).

There are logics allowing for other connectives and quantifiers: for example, *modal* logics have the connectives $\Box$ (necessity) and $\Diamond$ (possibility).

We will not study logics using other connectives than the fundamental ones in this course.

Syntactical classifications

A logical system may deal with expressions, like arithmetic, or it may speak only of formulae. In the latter case, the system is said to be *propositional*.

On the contrary, we may imagine a system that speaks of elements of some universe. In particular, we allow to quantify only over elements: $\forall x. \mathrm{even}(x) \vee \mathrm{odd}(x)$ is a formula of arithmetic. When we do not allow quantifiers to range over collections of elements, we will say that the system is *first-order*.

On the contrary, when we allow to quantify over collections of elements, we will speak of *higher-order* systems. For example, the formula $\forall S. \exists n. \max S < n \wedge \min S > -n \supset 0 \in S$ is a second-order formula since we quantify over S, which stands for a set of integers.

In this course, we will study propositional and first-order systems.

A rule of thumb says that all the mathematics developed before the 20^{th} century could be expressed as a collection of first-order theories.

Classical logics

Another way to classify logical theories is by means of their deductive systems. In fact, the standard connectives and quantifiers should be coupled with axioms and rules so to deduce formulae from other formulae.

For example, $\forall x. x = x$ is an axiom stating that equality is reflexive. And

$$\frac{A \quad B}{A \wedge B}$$

is a rule saying that, from the formulae A and B, we can deduce $A \wedge B$.

A logic is said to be *classical* when it allows to deduce $A \vee \neg A$ for any formula A. This principle is called *tertium non datur* or, also, the *Law of Excluded Middle*. For many reasons, it is an important principle, although it is debatable.

In this course, we will limit our study to classical systems, with one big exception: intuitionistic logic, which is, in some sense, the logic of computable functions.

Foundational issues

One of the big motivations for studying mathematical logic lies in the foundational problem: is Mathematics coherent?

In fact, as we will see in the end of this course, there is no hope to answer such a question within mathematics. But, still, relative coherence is an important question and it can be answered: is it impossible to deduce a statement and it negation in a given logical system, assuming that another theory is coherent?

As we will see, this question can be addressed, and some of its consequences are surprising: these will be presented at the end of this course.

Foundational issues

As a matter of fact, most branches of Mathematics could be developed using set theory plus classical logic as a framework: for example, arithmetic can be derived by identifying natural numbers with some special sets, and arithmetical operations become specific functions.

Since we have not to add any axiom or rule, but just definitions, that is, we add names, shorthands if you prefer, to the language, we could say that set theory is expressive enough to model arithmetic.

The pursue for a universal theory, one allowing to model every mathematical theory, is impossible to achieve, as we will prove in this course, but, still, some theories, like set theory, are close enough to allow us to reason on almost the whole Mathematics. In this course, we will discuss set theory to some extent, although we will not study any other such "universal" theory.

Soundness and completeness

The first and fundamental intent of a logical system is to derive the true sentences. To this aim, a deductive system is provided by the syntax, and a notion of truth is provided by the semantics.

It is worth noticing that different semantics may provide different notions of truth, and, in fact, truth is not universal in logic: it strictly depends on the semantics we will adopt. And yes, the same theory may have different semantics, not necessarily compatible.

This raises two major questions:

- is it the case that every formula we may prove is true?
- is it the case that every formula which is true, admits a proof in the deductive system?

Soundness and completeness

The first property is called *soundness*: we are not interested in non-sound deductive systems. A fundamental requirement for a syntax is to forbid deriving false consequences from true hypotheses. But we must prove that a syntax is sound with respect to a given semantics.

The second property is *completeness*: a syntax is a perfect description of a semantics when it allows to prove every true statement and to show that every false statement has a counterexample. We will see that completeness, as stated, is a very strong property. More, we will show that the majority of naturally interesting theories cannot be complete in the above sense, a shocking fact that changed the history of Mathematics.

There are many other properties of interest in logic, and, from time to time we will mention them, as appropriate. But soundness and completeness are the most fundamental ones, and we will focus on them in this course.

Your teacher

I am a researcher in Mathematical Logic. This means that my main job is to think, and, sometimes, to prove 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. In fact, I investigate mainly constructive logical systems, which have nice computational properties, and my favourite playground, the "universe" I work within, is topos theory, a branch of category theory.

For more, please visit my web page:

`http://marcobenini.wordpress.com`

References

For those interested in the history of logic, and its relations to Mathematics, a nice, short book is *Piergiorgio Odifreddi*, La matematica del Novecento—Dagli insiemi alla complessità, Piccola Biblioteca Einaudi, Einaudi, (2000), ISBN 88-06-15153-3.

There are many introductory textbooks of mathematical logic and a few important reference books. 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), ISBN 0-444-863888-5.

I do not have a preferred textbook, but I suggest the following notes by Prof. Helmut Schwichtenberg:

`http://www.mathematik.uni-muenchen.de/~schwicht/lectures/logic/ws03/ml.pdf`

Mathematical Logic

Lecture 2

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Propositional logic:

- Language
- Deduction system
- Informal meaning
- Examples

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 where 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 a 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.

Definition 2.1 (Formula)

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

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

Notice 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, when not specified otherwise, stand for variables.
- uppercase letters, when not specified otherwise, stand for objects in the metalanguage.

An important point to remark is that the definition of formula is by induction. So, we can use this structure to define new notions or to prove properties of formulae.

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

Definition 2.2 (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 $A \equiv B \land C$, $A \equiv B \lor C$, 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 stands for some truth value.
- $\top$ denotes true.
- $\bot$ denotes false.
- $A \wedge B$ is true when both A and B are true, and false otherwise.
- $A \vee B$ is true when A is true, or B is true, or both are true, and 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 those formulae 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 so 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 2.3 (Theory)

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

Definition 2.4 (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 formulae, 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:

$$\frac{A \wedge B}{A}\ \wedge E_1 \qquad \frac{A \wedge B}{B}\ \wedge E_2 \qquad \frac{A \quad B}{A \wedge B}\ \wedge I$$

we have two elimination rules, and an introduction rule.

Those governing disjunctions are:

$$\frac{A}{A \vee B}\ \vee I_1 \qquad \frac{B}{A \vee B}\ \vee I_2 \qquad \frac{A \vee B \quad \begin{matrix}[A] \\ \vdots \\ C\end{matrix} \quad \begin{matrix}[B] \\ \vdots \\ C\end{matrix}}{C}\ \vee E$$

Natural deduction

Implication and negation are subject to the following rules:

$$\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \supset B}\,{\supset}\mathrm{I} \qquad \dfrac{A \supset B \quad A}{B}\,{\supset}\mathrm{E}$$

$$\dfrac{\begin{matrix}[A]\\ \vdots \\ \bot\end{matrix}}{\neg A}\,\neg\mathrm{I} \qquad \dfrac{\neg A \quad A}{\bot}\,\neg\mathrm{E}$$

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:

$$\frac{}{\top}\,\top\mathsf{I} \qquad \frac{\bot}{A}\,\bot\mathsf{E}$$

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

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

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

$$A$$

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

$$\frac{}{A \vee \neg A}\,\text{lem}$$

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.

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* of *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 anymore from A.
- discharging is optional: we must not discharge an assumption when a rule does not allow, but we may (or we may not) discharge an assumption if the rules allows to.

When we do not want to specify 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.

Example 2.5

The formula $(p \supset q) \wedge p \supset q$ is a theorem in the pure logic, i.e., in the empty theory. In fact, this is a proof:

$$\dfrac{\dfrac{\dfrac{[(p \supset q) \wedge p]^*}{p \supset q}\,\wedge E_1 \quad \dfrac{[(p \supset q) \wedge p]^*}{p}\,\wedge E_2}{q}\,\supset E}{(p \supset q) \wedge p \supset q}\,\supset I^*$$

Discharged assumptions are written in square brackets and the superscripts indicate which inference rule discharges them.

In order to say that such a formula is always true, we could write $\vdash (p \supset q) \wedge p \supset q$.

Natural deduction

Example 2.6

The *double negation* law says that p is equivalent to $\neg\neg p$:

$$\dfrac{\dfrac{\dfrac{[\neg p]^* \quad [p]^\dagger}{\bot}\,\neg\mathsf{E}}{\neg\neg p}\,\neg\mathsf{I}^*}{p \supset \neg\neg p}\,\supset\mathsf{I}^\dagger \qquad\qquad \dfrac{\dfrac{\dfrac{}{p\vee\neg p}\,\mathsf{lem} \quad [p]^\ddagger \quad \dfrac{\dfrac{[\neg\neg p]^\S \quad [\neg p]^\ddagger}{\bot}\,\neg\mathsf{E}}{p}\,\bot\mathsf{E}}{p}\,\vee\mathsf{E}^\ddagger}{\neg\neg p \supset p}\,\supset\mathsf{I}^\S$$

In general, we say that two formulae A and B are *equivalent* when we can deduce one from the other, or, which is the same, when $A \supset B$ and $B \supset A$.

References

This lesson corresponds to sections 2.1, 2.2, and 2.3 of the lecture notes.

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

For a comprehensive and deep treatment of natural deduction, see *A.S. Troelstra* and *H. Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43, Cambridge: 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. 2015/16

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 lesson.

Definition 3.1 (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 follows:

- 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

Example 3.2

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:

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

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. In fact, connectives are not independent, as they can be mutually defined.

Interdependence of connectives

Proposition 3.3

Negation can be defined using implication and falsity.

Proof.

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

Proposition 3.4

The set of connectives $\wedge$, $\vee$, and $\neg$ suffices 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 3.5

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

Proof.

Checking the proof table, 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 the De Morgan's Laws. Here, $A = B$ between two formulae A and B means that both $A \supset B$ and $B \supset A$ hold, i.e., A and B are equivalent.

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.

Theorem 3.6 (Soundness)

If Γ is a set of formulae, 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

$\hookrightarrow$ Proof. (ii)

- if π is an instance of the $\bot E$ rule, then, by induction hypothesis, $[\![\bot]\!] = 1$, but we know that $[\![\bot]\!] = 0$, thus $0 = 1$. Then, since $[\![A]\!] \in \{0,1\}$, it follows that $[\![A]\!] = 1$.
- if π is an instance of the Law of Excluded Middle, $A \equiv B \lor \neg B$. But $[\![B \lor \neg B]\!] = 1$, as it is immediate to check by the truth tables.
- if π is an instance of $\neg I$, then, by the induction hypothesis applied to $\pi' \colon \Gamma \cup \{A\} \vdash \bot$, 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 twice to both antecedents, we get that $[\![\neg A]\!] = 1$ and $[\![A]\!] = 1$. Thus, $0 = [\![A]\!] = 1$. Then $[\![\bot]\!] = 0 = 1$.

$\hookrightarrow$

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 $\wedge E_1$, 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 $\wedge E_2$, 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 $[\![A]\!] = 1$.

↪

↪ Proof. (iv)

- if π is an instance of $\vee I_1$ 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 $\vee I_2$ 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

$\hookrightarrow$ 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 that $[\![C]\!] = 1$. 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. $\square$

References

The truth table semantics is described in Section 2.4 of the lecture notes.

The soundness theorem is folklore. In fact, 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 4

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Propositional logic:

- Orders
- Lattices
- Boolean algebras
- Semantics
- Examples

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 4.1 (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 $x \leq z$.

Definition 4.2 (Least upper bound)

Fixed an order $\mathscr{O} = \langle S; \leq \rangle$ 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, for every $x \in U$, $x \leq w$, it holds that $m \leq w$.

Definition 4.3 (Greatest lower bound)

Fixed an order $\mathscr{O} = \langle S; \leq \rangle$ 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, for every $x \in U$, $w \leq x$, it holds that $w \leq m$.

Definition 4.4 (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. By convention, $\bigvee \emptyset$ is denoted by $\bot$, and $\bigwedge \emptyset$ is denoted by $\top$.

It is immediate to see that, in a bounded lattice, every element is greater that $\bot$ and less than $\top$.

Definition 4.5 (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.

Definition 4.6 (Distributive lattice)

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

Lattices

It is a basic fact of lattice theory that the condition

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

is equivalent to

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

The proof of this fact is simple, but not relevant to our purposes.

A few other facts of interest are

- for each x, y in a lattice, $x \wedge y = y \wedge x$, and $x \vee y = y \vee x$, by definition.
- for each x in a bounded lattice, $x = x \wedge \top$ and $x = x \vee \bot$, by definition.

Proposition 4.7

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

- $x \wedge y = \bot = x \wedge z$,
- $x \vee y = \top = x \vee z$.

Thus, $y = y \wedge \top = y \wedge (x \vee z) = (y \wedge x) \vee (y \wedge z) = \bot \vee (y \wedge z) = (z \wedge x) \vee (z \wedge y) = z \wedge (x \vee y) = z \wedge \top = z$. □

Boolean algebras

Definition 4.8 (Boolean algebra)

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

Example 4.9

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 4.10

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 4.11

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 4.12 (Semantics)

Fixed a Boolean algebra $\mathscr{O} = \langle O; \leq \rangle$, and $v \colon V \to O$ 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.

Examples

Example 4.13

Let us fix the Boolean algebra given by the powerset of $\mathbb{N}$, ordered by inclusion. For simplicity, the variables have the form x_n, with $n \in \mathbb{N}$, and $\nu(x_n) = \{n\}$. It is immediate to check that meets are unions, and joins are intersections. Also, $\bot = \emptyset$ and $\top = \mathbb{N}$.
Then, $[\![x_3 \vee \neg x_3]\!] = [\![x_3]\!] \cup (\mathbb{N} \setminus [\![x_3]\!]) = \{3\} \cup (\mathbb{N} \setminus \{3\}) = \mathbb{N}$.
Also, $[\![x_5 \wedge \neg x_5]\!] = [\![x_5]\!] \cap (\mathbb{N} \setminus [\![x_5]\!]) = \{5\} \cap (\mathbb{N} \setminus \{5\}) = \emptyset$.
Finally, $[\![x_3 \vee \neg x_5]\!] = [\![x_3]\!] \cup (\mathbb{N} \setminus [\![x_5]\!]) = \{3\} \cup (\mathbb{N} \setminus \{5\}) = \mathbb{N} \setminus \{5\}$.

Every "true" formula seems to be intepreted in the top element of the algebra; every "false" formula seems to be interpreted in the bottom element of the algebra.
But a formula, which, according to the truth table semantics, is sometimes "true" and sometimes "false", depending on the values of its variables, seems to be interpreted in a "truth-value" which is neither $\top$ nor $\bot$.

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), ISBN 978-1-59102-089-9.

Two excellent references for orders, lattices, and Boolean algebras are *B. A. Davey and H. A. Priestley*, Introduction to Lattices and Order, Cambridge University Press, (2002), ISBN 978-0-521-78451-1, and *George Grätzer*, General Lattice Theory, second edition, Birkhäuser, (1996), ISBN 978-3-7643-6996-5.

Section 2.5 of the Lecture Notes contains the text of this lesson.

Mathematical Logic

Lecture 5

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Propositional logic:

- Soundness

Soundness

This lesson will illustrate just one theorem: soundness.

Definition 5.1 (Validity)

A formula A is *valid* or *true* in a Boolean algebra $\mathscr{O} = \langle O; \leq \rangle$ together with an interpretation $v \colon V \to O$ of variables, when $[\![A]\!] = \top$.
A set of formulae is *valid* or *true* when each formula in the set is valid.

Theorem 5.2 (Soundness)

In any Boolean algebra $\mathscr{O} = \langle O; \leq \rangle$, for any interpretation $v \colon V \to O$ of variables, which makes true the theory T and the assumptions in the finite set Δ, if A is the conclusion of a proof π from Δ in T, then A is valid.

Soundness

Proof. (i)

The proof is by induction on the structure of the proof π: we prove 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

$\hookrightarrow$ 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 formulae 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 $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!]$.

$\hookrightarrow$

Soundness

$\hookrightarrow$ Proof. (iv)

- if π is an instance of $\supset$-introduction, then $A \equiv B \supset C$ for some formulae 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 $\neg$, $\bigwedge_{G\in\Gamma}[\![G]\!] \leq (\neg[\![B]\!] \wedge [\![B]\!]) \vee ([\![A]\!] \wedge [\![B]\!]) = \bot \vee ([\![A]\!] \wedge [\![B]\!]) = [\![A]\!] \wedge [\![B]\!] \leq [\![A]\!]$.

$\hookrightarrow$

Soundness

$\hookrightarrow$ 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 from Δ in the theory T, $\bigwedge_{G \in \Delta} [\![G]\!] \leq [\![A]\!]$. But, by hypothesis, for every $G \in \Delta$, $[\![G]\!] = \top$, so $\bigwedge_{G \in \Delta} [\![G]\!] = \top$, thus, by definition of $\top$, $\top \leq [\![A]\!] \leq \top$, that is, by anti-symmetry, $[\![A]\!] = \top$. $\square$

References

The idea of the proof of the Soundness Theorem is folklore: in fact, 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 *P. Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002), ISBN 978-0-19-853425-9 and 978-0-19-851598-2.

The general textbooks on lattice theory, see the previous lesson, could be useful to connect this result to its algebraic interpretation.

The content of this lesson can be found in Section 2.6 of the Lecture Notes.

Mathematical Logic

Lecture 6

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Propositional logic:

- Completeness

Completeness

This lesson will introduce the first part of the Completeness Theorem.

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

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

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

The proof is complex and subtle.

In the first place, it is worth noticing that, if $\pi \colon \Gamma \vdash_T A$, then there is a finite $\Delta \subseteq \Gamma$ such that $\pi \colon \Delta \vdash_T A$. In fact, 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.

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

Strategy

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;
- show that, for any other Boolean algebra $\mathbb{O}$, there is a function $e\colon \mathbb{B} \to \mathbb{O}$ which preserves the ordering relation;
- deduce that, up to isomorphisms, if $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!]$ in every Boolean algebra, then $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!]$ in $\mathbb{B}$, which is obvious, and vice versa, which is **not** obvious;
- prove that, for any finite set Γ of formulae and for any formula A, if $\bigwedge_{G \in \Gamma} [\![G]\!] \leq [\![A]\!]$ in $\mathbb{B}$, then there exists $\pi\colon \Gamma \vdash_T A$.

This strategy is general: most completeness results for most logical systems, follow this pattern.

Canonical model

Definition 6.1 (Canonical Boolean algebra)

Let T be a theory. Then the *canonical Boolean algebra* $\mathbb{B}(T)$ on T is the set $\{A\colon A \text{ is a formula in the language of } T\} / \sim$, where $A \sim B$ if and only if $A \vdash_T B$ and $B \vdash_T A$, together with the order defined by $[A]_\sim \leq_{\mathbb{B}(\mathbb{T})} [B]_\sim$ exactly when $A \vdash_T B$.

For the sake of simplicity, when it is clear from the context, we omit the subscripts.

An auxiliary result

Lemma 6.2

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

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 $\xi : \Gamma \vdash_T \bot$, so applying the $\bot$-elimination rule to ξ gives the required ν.

$\hookrightarrow$

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

↪

↪ 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. $\square$

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 $\sim$, in fact, if $[A]_\sim = [A']_\sim$ and $[B]_\sim = [B']_\sim$, 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, $[A']_\sim \leq [B']_\sim$.
- By the assumption rule, $A \vdash_T A$, so $[A]_\sim \leq [A]_\sim$, i.e., $\leq$ is reflexive.
- If $[A]_\sim \leq [B]_\sim$ and $[B]_\sim \leq [C]_\sim$, then $A \vdash_T B$ and $B \vdash_T C$, so, by Lemma 6.2, $A \vdash_T C$, that is, $[A]_\sim \leq [C]_\sim$, i.e., $\leq$ is transitive.
- If $[A]_\sim \leq [B]_\sim$ and $[B]_\sim \leq [A]_\sim$, then $A \vdash_T B$ and $B \vdash_T A$, so, by definition of $\sim$, $A \sim B$, that is, $[A]_\sim = [B]_\sim$, i.e., $\leq$ is anti-symmetric. □

Properties of the canonical model

Proposition 6.5

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

Proof.

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

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

References

The completeness proof can be found in Section 2.7 of the lecture notes.

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), ISBN 978-0198524960.

Mathematical Logic

Lecture 7

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Propositional logic:

- Completeness

Proposition 7.1

$\mathbb{B}(T)$ *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 $(A \lor B) \land (A \lor C) \vdash_T A \lor (B \land C)$ since

$$\dfrac{\dfrac{(A \lor B) \land (A \lor C)}{A \lor B}\,{\scriptstyle \land E_1} \qquad \dfrac{[A]^*}{A \lor (B \land C)}\,{\scriptstyle \lor I_1} \qquad \begin{array}{c}[B]^* \\ \vdots \\ A \lor (B \land C)\end{array}}{A \lor (B \land C)}\,{\scriptstyle \lor E^*}$$

where the third antecedent is

$$\dfrac{\dfrac{(A \lor B) \land (A \lor C)}{A \lor C}\,{\scriptstyle \land E_2} \qquad \dfrac{[A]^\dagger}{A \lor (B \land C)}\,{\scriptstyle \lor I_1} \qquad \dfrac{\dfrac{B \quad [C]^\dagger}{B \land C}\,{\scriptstyle \land I}}{A \lor (B \land C)}\,{\scriptstyle \lor I_2}}{A \lor (B \land C)}\,{\scriptstyle \lor E^\dagger}$$

Thus, $(A \lor B) \land (A \lor C) \sim A \lor (B \land C)$, and the conclusion follows. □

Properties of the canonical model

Proposition 7.2

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

Proof.

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

$$\dfrac{\dfrac{A \wedge \neg A}{A}\,\wedge E_1 \qquad \dfrac{A \wedge \neg A}{\neg A}\,\wedge E_2}{\bot}\,\neg E$$

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 7.3

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

Classifying models

Proposition 7.4

Fixed a theory T, let $\mathbb{O}$ be any Boolean algebra and let $v\colon V \to \mathbb{O}$ be any assignment of variables on it such that $[\![A]\!] = \top$ for any $A \in T$. 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 5.2, applied in the $\mathbb{O}$ Boolean algebra with the v assignment, $[\![B]\!]_\mathbb{O} \leq_\mathbb{O} [\![C]\!]_\mathbb{O}$. □

Definition 7.5 (Canonical map)

Fixed a theory T, let $\mathbb{O}$ be any Boolean algebra and let $v\colon V \to \mathbb{O}$ be any assignment of variables on it such that $[\![A]\!] = \top$ for any $A \in T$. 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 $\mathbb{B}$. In fact, if $[A] = [A']$, then, $[A] \leq [A']$ and $[A] \leq [A']$, so, by Proposition 7.4, $[\![A]\!] \leq [\![A']\!]$ 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 7.6 (Completeness)

Fixed a theory T, for any finite set Γ of formulae and for any formula A, if $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$ in any Boolean algebra and any assignment of variables which makes the theory T true, then there is a natural deduction proof $\pi : \Gamma \vdash_T A$.

Proof.

If $\bigwedge_{G\in\Gamma}[\![G]\!] \leq [\![A]\!]$, then $[\![\bigwedge_{G\in\Gamma} G]\!] \leq [\![A]\!]$.
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 $\pi : \bigwedge_{G\in\Gamma} G \vdash_T A$. Noticing 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 7.7

If $[\![A]\!] = \top$ *in every Boolean algebra and with any assignment of variables making the theory* T *true, then there is a proof* $\pi : \vdash_T A$.

Proof.

If $[\![A]\!] = \top$, then $\top = [\![\top]\!] \leq [\![A]\!]$, being $\leq$ reflexive. By the Completeness Theorem, 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.

References

The completeness proof can be found in Section 2.7 of the lecture notes.

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), ISBN 978-0198524960.

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 8

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

First order logic:

- Motivation
- Language
- Bounded variables
- Substitution

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 formulae 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.

Definition 8.1 (Signature)

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

- a 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 notation $f : s_1 \times \cdots \times s_n \to s_0 \in F$ and $r : s_1 \times \cdots \times s_n \in R$ means 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 language: sorts stands for collection of elements, functions are used to denote elements, while relations are used to form basic formulae.

Example 8.2
The signature

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

specifies the basic language for arithmetic. There is one sort, which, in the intended interpretation, stands for the collection of natural numbers. There is constant, 0, denoting the zero natural number, there is a function S, which stands for "successor", denoting the next natural number, so that, in the intended interpretation, $S(5) = 6$, 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.

Example 8.3

The signature $\mathscr{G} = \langle \{G\} ; \big\{1\colon G, \cdot\colon G \times G \to G, -^{-1}\colon G \to G\big\} ; \{=\colon G \times G\} \rangle$ describes the language of the theory of groups.

Example 8.4

The signature $\mathscr{O} = \langle \{O\} ; \emptyset ; \{\leq\colon O \times O\} \rangle$ describes the language of the theory of orders.

Example 8.5

The signature $\mathscr{L} = \langle \{E, L\} ; \{\mathrm{nil}\colon L, \mathrm{cons}\colon E \times L \to L\} ; \{=_E\colon E \times E, =_L\colon L \times L\} \rangle$ 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 8.6 (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 $x : s$. We require that there is an infinite amount of variables for each type $s \in S$. A *term*, along with the set of its *free variables*, is inductively defined as:

- if $x : s \in V$, then x is a term of type s, and $\mathrm{FV}(x) = \{x\}$;
- if $f : s_1 \times \cdots \times s_n \to s_0 \in F$ 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 $t : s$ to say that the term t has type s.

Example 8.7
Using the signature $\mathscr{N}$ 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 $\mathbb{N}$. Notice how $x+0$ and $x(1+2)$ are **not** terms.

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

Formulae

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

Definition 8.8 (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 formulae, and $\mathrm{FV}(\top) = \mathrm{FV}(\bot) = \emptyset$.
- if $r \colon s_1 \times \cdots \times s_n \in R$ is a relation symbol, and $t_1 \colon s_1, \dots, t_n \colon s_n$ are terms, then $r(t_1, \dots, t_n)$ is an *atomic* formula, and $\mathrm{FV}(r(t_1, \dots, t_n)) = \bigcup_{i=1}^{n} \mathrm{FV}(t_i)$.
- if A and B are formulae, 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 $x \colon s$ is a variable and A is a formula, so are $\forall x \colon s.\, A$ and $\exists x \colon s.\, A$, and $\mathrm{FV}(\forall x \colon s.\, A) = \mathrm{FV}(\exists x \colon s.\, A) = \mathrm{FV}(A) \setminus \{x\}$.

Formulae

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

- instead of propositional variables, we have atomic formulae, which link the formulae with terms by means of a relation;
- there are quantified formulae, 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. In fact, 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, the 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. In fact, from a formula A where the variable x appears as 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. And 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 8.9 (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 $t[r/x] = r$;
- if t is a variable, but $t \not\equiv x$, $t[r/x] = t$;
- if $t \equiv f(t_1, \dots, t_n)$, then $t[r/x] = f(t_1[r/x], \dots, t_n[r/x])$.

Notice that the substitution operation is defined only when t and x share the same type.

Substitution

Definition 8.10 (Substitution on formulae)

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

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

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 $\exists x : \mathbb{N}. x^2 = x^3$. In fact, the formula is true, because $1^2 = 1 = 1^3$, but, evidently, it happens just for **some** values of x, which 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, which does not appear in B and t.

An example may clarify why this must be done: let $A \equiv \exists x : \mathbb{N}. x + y = 2y$, and let $t \equiv 2x$. If we do not rename variables, $A[t/y]$ would give $\exists x : \mathbb{N}. x + 2x = 2(2x)$, that is, $\exists x : \mathbb{N}. 3x = 4x$. We notice the A holds whenever $x = y$, but $A[t/y]$ does not. The problem is that the x in t and the one in A should be kept distinct—and we do this by renaming before performing the substitution.

References

The content of this lesson is reported in Section 3.1 of the Lecture Notes.

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 pass 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), ISBN 0-7204-28440.

Mathematical Logic

Lecture 9

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

First order logic:

- Definitions
- Natural deduction
- Examples

Definitions

The language of first-order logic is cumbersome. Despite the fact that 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 distant 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, formulae, proofs) as our main proving instrument.

There is a way in between: we can construct a reasonable formal 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 construction 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.

Definition 9.1 (Function definition)

Fixed a first-order language with equality, let f be a new symbol. Whenever it holds that $\forall x_1 : s_1. \dots \forall x_n : s_n. \exists y : s_0. F \wedge \forall z : s_0. F[z/y] \supset z = y$, with $\mathrm{FV}(F) \subseteq \{x_1, \dots, x_n, y\}$, then $f : s_1 \times \dots \times s_n \to s_0$ can be used as an additional function symbol, since it can be removed from the language by the rule

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

for any formula A, and with the obvious extensions to the definition of substitution. As far as a different syntax is non-ambiguous, we allow it in place of the standard functional syntax.

Definitions

Definition 9.2 (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 $\mathrm{FV}(R) = \{x_1, \ldots, x_n\}$, since it can be removed by substituting $R(t_1, \ldots, t_n)$ whenever $r(t_1, \ldots, t_n)$ occurs in any formula A. Again, as far as the syntax is non-ambiguous, we allow fancy syntactical constructions.

Notice that there is no way to define new sorts. This happens because defining new sorts require sophisticated rules which cannot be easily managed by translating into the original language.

Example 9.3

Consider the language generated by the signature:

$$\langle\{\mathbb{N}\};\{0\colon \mathbb{N}, \mathrm{succ}\colon \mathbb{N}\to\mathbb{N}, \mathrm{add}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}, \mathrm{times}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}\};\{\mathrm{eq}\colon \mathbb{N}\times\mathbb{N}\}\rangle$$

Then, the syntax $x+y$ stands for $\mathrm{add}(x,y)$, xy stands for $\mathrm{times}(x,y)$, and $x=y$ stands for $\mathrm{eq}(x,y)$. The last definition is a relation symbol definition, while the first two definitions are function symbol definitions, corresponding to the formulae

$$\forall x\colon \mathbb{N}.\forall y\colon \mathbb{N}.\exists z\colon \mathbb{N}.\mathrm{eq}(\mathrm{add}(x,y),z)\land\forall w\colon \mathbb{N}.\mathrm{eq}(\mathrm{add}(x,y),w)\supset\mathrm{eq}(z,w)$$

and

$$\forall x\colon \mathbb{N}.\forall y\colon \mathbb{N}.\exists z\colon \mathbb{N}.\mathrm{eq}(\mathrm{times}(x,y),z)\land\forall w\colon \mathbb{N}.\mathrm{eq}(\mathrm{times}(x,y),w)\supset\mathrm{eq}(z,w)$$

that we must prove.

Example 9.4

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

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*.

Notice 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, $x\colon s \not\in \mathrm{FV}(G)$ and $x\colon s \not\in \mathrm{FV}\, A$. Again, $x\colon s$ is said to be an eigenvariable. Notice how this inference rule discharges the assumption B.

Notice 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:

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

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

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

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

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.

Examples

Example 9.5

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

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

Examples

Example 9.6

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

Putting $P \equiv \neg A$ and applying the double negation law, one gets that $\forall x\colon s.\,A = \neg\exists x\colon s.\,\neg A$.

Examples

Example 9.7

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

Example 9.8

$$
\dfrac{
  \dfrac{}{(\exists x : s.\, \neg P) \lor \neg(\exists x : s.\, \neg P)}\,\text{lem}
  \qquad
  [\exists x : s.\, \neg P]^1
  \qquad
  \dfrac{
    \dfrac{
      \begin{array}{c}[\neg \exists x : s.\, \neg P]^1 \\ \vdots \\ \forall x : s.\, P\end{array}
      \qquad
      [\neg \forall x : s.\, P]^2
    }{\bot}\,\neg\text{E}
  }{\exists x : s.\, \neg P}\,\bot\text{E}
}{
  \dfrac{\exists x : s.\, \neg P}{(\neg \forall x : s.\, P) \supset \exists x : s.\, \neg P}\,\supset\text{I}^2
}\,\lor\text{E}^1
$$

Examples

Example 9.9

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

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

The instance of the $\forall I$ rule on the top is invalid, since $x\colon s$ 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 that, since $P[0/x]$ is true, every natural number is even!

Example 9.10

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 that there is no even natural number!

References

This lesson covers the material in Section 3.2 of the Lecture Notes.

Our treatment of definitions follows *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), ISBN 0-7204-28440.

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

Mathematical Logic

Lecture 10

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

First order logic:

- Informal meaning
- Semantics
- Examples

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 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, formulae stands for a truth value, either true or false, as in the propositional case. And connectives have the intended propositional meaning, we already illustrated.

Atomic formulae, $r(t_1, \dots, t_n)$, are true when the argument $(t_1, \dots, t_n)$ is in the relation denoted by r.

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

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

Definition 10.1 (Σ-structure)

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

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

Semantics

To make clear the relation between a signature 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* in the Σ-structure.

Definition 10.2 (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 collection of functions $\nu_s \colon \{v \colon v \colon s \in V\} \to [\![s]\!]$, 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 10.3 (Interpretation of formulae)

Let $\Sigma = \langle S; F, R \rangle$ be a signature, let $\mathscr{M}$ be a Σ-structure, and let v 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 $A \equiv r(t_1, \dots, t_n)$, $[\![A]\!] = 1$ if $\left([\![t_1]\!], \dots, [\![t_n]\!]\right) \in [\![r]\!]$, 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 $A \equiv \forall x \colon s.\, B$ or $A \equiv \exists x \colon s.\, B$, let $\xi = \{\xi_s\}_{s \in S}$ be an evaluation of variables such that, $\xi_\alpha = v_\alpha$, for each $\alpha \neq s$, and $\xi_s(v) = v_s(v)$ for each $v \neq x$. Then, $[\![\forall x \colon s.\, B]\!] = 1$ if, for all the possible ξ, $[\![B]\!] = 1$, and $[\![\forall x \colon s.\, B]\!] = 0$ otherwise. Also, $[\![\exists x \colon s.\, B]\!] = 1$ if, there is a ξ such that $[\![B]\!] = 1$, and $[\![\exists x \colon s.\, B]\!] = 0$ 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* $\{(x, x) \colon x \in [\![s]\!]\}$.

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 $=$ should be read as a shorthand for $=_s$, which stands for the diagonal relation on the set denoted by the sort s.

Example 10.4

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 $2=2$.

Example 10.5

Fix the signature of arithmetic, and consider the standard model of natural numbers. Let 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.

Example 10.6

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.

Example 10.7

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 $[\![x=(SS0)y]\!]=0$ as the left side differs from the right.

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.

References

The interpretation of formulae, 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 *C.C. Chang* and *H.J. Keisler*, Model Theory, Studies in Logic and the Foundations of Mathematics, 3rd edition, Elsevier, (1990), ISBN 008088007X. Nevertheless, this text is quite dated, and an introduction to the basics of contemporary model theory can be found in *W. Hodges*, A Shorter Model Theory, Cambridge University Press, (1997), ISBN 0-521-58713-1.

We will not introduce model theory in this course, for reasons of time.

Mathematical Logic

Lecture 11

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

First order logic:

- Soundness

Soundness

This lesson will illustrate just one theorem: soundness.

Definition 11.1 (Validity)

A formula A is *valid* or *true* in a Σ-structure $\mathscr{M}$ together with an interpretation ν of variables, when $[\![A]\!] = 1$.
A set of formulae is *valid* or *true* when each formula in the set is valid.

Theorem 11.2 (Soundness)

In any Σ-structure $\mathscr{M}$, for any interpretation ν of variables, which makes true the theory T and the assumptions in the finite set Δ, if A is the conclusion of a proof π from Δ in T, then A is valid.

Proof. (i)

First, we observe that, by Definition 10.3, 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 $A \equiv \forall x : s. x = x$, so $[\![A]\!] = 1$ when $[\![x = x]\!] = 1$ for each possible evaluation of the variable x in $[\![s]\!]$. So, if x gets mapped to $e \in [\![s]\!]$, $(e,e) \in \left\{(z,z) : z \in [\![s]\!]\right\}$, so $[\![x = x]\!] = 1$ for any e.

$\hookrightarrow$

Soundness

$\hookrightarrow$ Proof. (ii)

- if π is an instance of sym, then $A \equiv \forall x: s. \forall y: s. x = y \supset y = x$, 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 $(e_x, e_y) \in \{(z,z): z \in [\![s]\!]\}$, then $e_x = e_y$, thus $(e_y, e_x) \in \{(z,z): z \in [\![s]\!]\}$, that is, $[\![x = y \supset y = x]\!] = 1$.
- if π is an instance of trans, then $A \equiv \forall x: s. \forall y: s. \forall z: s. x = y \wedge y = z \supset x = z$, 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 $(e_x, e_y) \in \{(z,z): z \in [\![s]\!]\}$ and $(e_y, e_z) \in \{(z,z): z \in [\![s]\!]\}$, then $e_x = e_y = e_z$, and thus $(e_x, e_z) \in \{(z,z): z \in [\![s]\!]\}$, that is, $[\![x = y \wedge y = z \supset x = z]\!] = 1$.

$\hookrightarrow$

Soundness

$\hookrightarrow$ Proof. (iii)

- if π is an instance of fun, then $A \equiv \forall x_1 \colon s_1. \dots \forall x_n \colon s_n. \exists! z \colon s_0. z = f(x_1, \dots, x_n)$, so $[\![A]\!] = 1$ exactly when z can be uniquely mapped into a value e_z in $[\![s_0]\!]$ so that $(e_z, [\![f]\!](e_{x_1}, \dots, e_{x_n})) \in \{(z, z) \colon z \in [\![s]\!]\}$, which is evidently true for $e_z = [\![f]\!](e_{x_1}, \dots, e_{x_n})$.
- 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.

$\hookrightarrow$

Soundness

$\hookrightarrow$ 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 formulae 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 $[\![A]\!] = 1$.

$\hookrightarrow$

↪ Proof. (v)

- if π is an instance of $\supset$-introduction, then $A \equiv B \supset C$ for some formulae 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 $A \equiv \forall x\colon s.\, B$, and, by induction hypothesis, $[\![B]\!] = 1$ for every evaluation of variables which makes the assumptions true. But, since $x\colon s$ 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, $[\![\forall x\colon s.\, B]\!] = 1$. So, in particular, when x evaluates to $[\![t]\!]$, $[\![A]\!] = [\![B[t/x]]\!] = 1$.

↪

↪ Proof. (vii)

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

References

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), ISBN 0-7204-28440.

It is worth comparing the proof in this lesson with the propositional proof using the truth-tables semantics.

Mathematical Logic

Lecture 12

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

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.
- We will show that any set of formulae 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 12.1 (Consistent set)

Fixed a first-order signature, a set of formulae 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 12.2 (Maximal consistent set)

Fixed a first-order signature, a set of formulae 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 12.3

For any set of formulae 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 2.6, it follows that $\Gamma \vdash A$.
Conversely, $\Gamma \cup \{\neg A\} \vdash A$ by hypothesis, and $\Gamma \cup \{\neg A\} \vdash \neg A$ by assumption, 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 part. □

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 12.3, 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 notice that any super set of a set of non consistent formulae 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.

Proposition 12.4

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 12.3, $\Gamma \vdash \neg A$ 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 12.5

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 12.6

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* $\forall x : s. A \in \Gamma$ *then* $A[t/x] \in \Gamma$ *for each term* $t : s$;
8. *if* $\neg(\exists x : s. A \in \Gamma)$ *then* $\neg A[t/x] \in \Gamma$ *for each term* $t : s$.

Proof. (i)

Since $\Gamma \vdash \top$ by truth introduction, $\top \in \Gamma$. Hence, since $\neg\top$ is equivalent to $\bot$, $\bot \notin \Gamma$. The condition on atomic formulae follows from Proposition 12.4. ↪

↪ Proof. (ii)

If $A \wedge B \in \Gamma$ then, by and elimination, $\Gamma \vdash A$ and $\Gamma \vdash B$. So, by Corollary 12.5, $A \in \Gamma$ and $B \in \Gamma$. Moreover, by the 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 the 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 12.7

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* $f : s_1 \times \cdots \times s_n \to s_0$ *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* $p : s_1 \times \cdots \times s_n$ *in the language.*

Proof.

Since all these equalities can be deduced from Γ applying the inference rules in an elementary way, by Corollary 12.5 the results follow. □

Closure of maximal consistent sets

Two evident conditions are lacking from Proposition 12.6:

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

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

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. Take Ψ as the collection of true formulae on the model U under the evaluation σ. Evidently, Ψ is consistent, since it has a model. 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 12.8 (Henkin set)

A set of formulae 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.

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), ISBN 0-7204-28440.

Gödel's proof was his doctoral dissertation, and it is based on a 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 13

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

First order logic:

- Completeness

Canonical model

Lemma 13.1

If Γ is a Henkin set, then there is a Σ-structure $\mathscr{M}$ together with an evaluation of variables σ which makes Γ true.

Proof. (i)

Let T be the set of terms in the language. Define $t \sim r$ when $t: s, r: s \in T$ and $t = r \in \Gamma$. By the properties of a Henkin set, see Proposition 12.7, $\sim$ is an equivalence relation. So, it induces a partition on T. Thus, we define $U = \{\{[t]_\sim : t: s \in T\}\}_{s \in S}$.

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

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

Notice how this definition is legitimate, since the class $[f(t_1, \ldots, t_n)]_\sim$ does not depend on the choice of the representatives $[t_1]_\sim$, ..., $[t_n]_\sim$, by a direct application of Proposition 12.7. ↪

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 $[t_1]_\sim$, ..., $[t_n]_\sim$ by Proposition 12.7.

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

Define σ as 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 formulae, 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 $A \equiv p(t_1, \dots, t_n)$, $[\![A]\!] = 1$ if and only if $([\![t_1]\!], \dots [\![t_n]\!]) \in [\![p]\!]$, that is, $([t_1]_\sim, \dots, [t_n]_\sim) \in [\![p]\!]$, and, by definition of the model, this happens exactly when $p(t_1, \dots, t_n) \in \Gamma$, 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 12.6, $B \in \Gamma$ and $C \in \Gamma$, thus $[\![A]\!] = 1$. On the contrary, when $\neg A \in \Gamma$, by Proposition 12.6, $\neg B \in \Gamma$ 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 12.6, $B \in \Gamma$ or $C \in \Gamma$, thus $[\![A]\!] = 1$. On the contrary, when $\neg A \in \Gamma$, by Proposition 12.6, $\neg B \in \Gamma$ and $\neg C \in \Gamma$, and, being Γ maximal consistent, $B \notin \Gamma$ and $C \notin \Gamma$. In both cases, $[\![A]\!] \neq 1$, 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 12.6, $\neg B \in \Gamma$ or $C \in \Gamma$, thus $[\![A]\!] = 1$. On the contrary, when $\neg A \in \Gamma$, by Proposition 12.6, $B \in \Gamma$ and $\neg C \in \Gamma$, and, being Γ maximal consistent, $B \in \Gamma$ and $C \notin \Gamma$. In both cases, $[\![A]\!] \neq 1$, so $[\![A]\!] = 0$.

↪

↪ 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 12.6, 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 12.6, there is a term $t\colon s$ such that $\neg B[t/x] \in \Gamma$, so, by induction hypothesis, $[\![B[t/x]]\!] = 0$, thus $[\![A]\!] = 0$.

↪

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 the σ evaluation. □

Corollary 13.2

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

References

The relevant references have been given in the previous lecture.

Mathematical Logic

Lecture 14

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

First order logic:

- Completeness
- Compactness

Existence of Henkin sets

Proposition 14.1

Let Γ be a consistent set of formulae on the signature Σ. Then, there is a set of formulae 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

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

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 formulae 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

$\hookrightarrow$ Proof. (ii)

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

1. $\Gamma_j \subseteq \Gamma_i$ for every $j < i$;
2. Γ_i is consistent;
3. no more that i constant 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.

If $i \leq \lambda$ is a limit infinite ordinal, we put $\Gamma_i = \bigcup_{j<i} \Gamma_j$. By definition, condition (1) holds. If $\Gamma_i \vdash A$ 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 assumption 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, proving (3) $\hookrightarrow$

↪ Proof. (iii)

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.
- 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.
  Suppose Γ_i to be 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 $S_k \equiv \exists x \colon s.A$, $B \equiv A$, thus $\Gamma_k \cup \{S_k\} \vdash \bot$ by exists-elimination. If $S_k \equiv \neg\forall x \colon s.A$, $B \equiv \neg A$, thus $\Gamma_k \cup \{S_k\} \vdash \bot$ 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

$\hookrightarrow$ Proof. (iv)

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

Theorem 14.2

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

Proof.

By Proposition 14.1, Γ can be extended to a Henkin set Δ. By Lemma 13.1, Δ, and thus Γ, has a model satisfying the cardinality constraints. □

Theorem 14.3 (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 14.2, $\Gamma \cup \{\neg A\}$ is non consistent.
Then, by Proposition 12.3, $\Gamma \vdash A$. □

Compactness

Theorem 14.4 (Compactness)

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

Proof.

By hypothesis, applying the Soundness Theorem 11.2, every finite subset of Γ is consistent.

Suppose Γ to be 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 14.2, Γ has a model. □

References

The relevant references have been given in the previous lectures.

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 formulae. This fact turns out to be critical in many situations. A good starting reference is *W. Hodges*, A Shorter Model Theory, Cambridge University Press, (1997), ISBN 0-521-58713-1.

Mathematical Logic

Lecture 15

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Set theory:

- Language
- Classes and sets
- Paradoxes
- Comparing sets

Language

The language of the theory of sets is the usual 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.

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.

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 $\forall x. x \subseteq y \supset A$;
- there is at most one x such that A, $\exists^* x. A$ abbreviates $\forall x. \forall y. A \wedge A[y/x] \supset x = y$ where $y \notin \mathrm{FV}(A)$.

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. It is important to distinguish a set by its extension, which is really the way it relates with other sets, which may be elements of it.

As we will see, sets will have properties not shared by classes, e.g., sets have a *cardinality*, while proper classes have not. These properties are what identifies 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 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 usually called the unrestricted *Comprehension Axiom* and it has been used to define sets by Gottlob Frege.

Unfortunately, the 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 set theory is non consistent.

Comparing sets

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

A few fact, which seem to be paradoxical at the first sight, are of common use.

Comparing two sets means to establish a correspondence between them. A function, mapping all the elements of one set in the element 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.

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 the 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 15.1 (Schröder-Bernstein)

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

Proof. (i)

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

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

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

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

We must show that $h(A) = B$. Firstly, 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, by induction on n, $g(z) \notin C_n$ for any n, 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.

Example 15.2
Let $P = \{2n\colon n \in \mathbb{N}\}$. Since $f\colon P \to \mathbb{N}$ such that $f(x) = x$ is injective, and $g\colon \mathbb{N} \to P$ such that $g(x) = 2x$ is injective, by Theorem 15.1 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.

Example 15.3
$\mathbb{N} \times \mathbb{N} \cong \mathbb{N}$

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 15.1 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 $k > 0$.

Example 15.4

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 $h(\{x_i\}_{1 \leq i \leq n}) = (n, g_n(x_1, \ldots, x_n))$. 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 15.1.

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 formulae 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.

Example 15.5

$\wp(\mathbb{N}) \not\cong \mathbb{N}$.

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 $\chi_A \colon \mathbb{N} \to \{0,1\}$. Suppose that all these functions are in bijection with $\mathbb{N}$: then, there is a bijective function e which enumerates them. So, we have a sequence $\wp(\mathbb{N}) \cong \{\chi_{A_i}\}_{i \in \mathbb{N}}$ such that the i-th function is given by $e(i)$. Define a function $\Delta \colon \mathbb{N} \to \{0,1\}$ 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, $\wp(\mathbb{N}) \not\cong \mathbb{N}$.

Again, this result can be generalised to any infinite set. As a side effect, since the functions $\mathbb{N} \to \{0,1\}$ are in evident 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) ISBN 0-387-90092-6. Reprinted by Martino Fine Books (2011), ISBN 978-1-61427-131-4.

The content of this lesson covers some classical proposition of set theory, due to Georg Cantor. The original articles can be found online, but a very nice translation in French can be found in *Georg Cantor*, Sur Les Fondements De La Théorie Des Ensembles Transfinis, Editions Jacques Gabay, (1989), ISBN 2-87647-062-4.

Mathematical Logic

Lecture 16

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Set theory:

- Axioms

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 16.1

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 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 16.1, 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 set.

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.

Notice 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.

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.

Not immediately, but it is possible to formally prove 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 in biijection with the set of natural numbers. We will denote this minimal set as ω.

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 $\{z : z \subseteq x\}$.

Working formally, 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, we want to introduce an axiom which prevents this phenomenon to happen. The consequence will be that $\wp(x) \neq x$ for every set x, thus proving the Cantor's Theorem we illustrated in the last lecture.

Axioms: regularity

Axiom (Regularity)

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

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 $x \in y \in x$, ...

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.

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 construction of subsets: $\{x \in A \colon P\}$ is exactly the result of applying separation to A with the property P. It follows that P must contain just one free variable, x.

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 XXth century.

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

References

The content of this lesson derives from the presentation in *Kenneth Kunen*, Set Theory: An Introduction to Independence Proofs, Studies in Logic and the Foundations of Mathematics 102, Elsevier, (1980), ISBN 0-444-86839-9.

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), ISBN 0-444-863888-5.

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.

Mathematical Logic

Lecture 17

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Set theory:

- Ordinals
- Induction
- Arithmetic

Well orders

Definition 17.1

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 17.2

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. Also, it is immediate to define an order relation on A which makes it a total order. But it is not clear whether it is possible to define an order relation which makes it a well order.

But 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 17.3

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

This definition is significant because it allows to prove

Proposition 17.4

Each 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. □

Proposition 17.5

If S is an ordinal and $x \in S$, then x is an ordinal.

Proof.

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

Proposition 17.6

If U is a set of ordinals, then U is well ordered by inclusion.

Proof.

Consider the structure $\langle U; \subseteq \rangle$. It is evident that it forms an order. If $S \subseteq U$ is non empty, consider $\bigcap S = \{x : \forall y \in S. x \in y\}$. So $\bigcap S \subseteq y$, for all $y \in S$. Thus $\bigcap S$ is totally well ordered by $\in \cup =$ and, for each $x \in \bigcap S$, $x \subseteq y$, that is, for all $z \in x$, $z \in y$, so $\forall z \in x. z \in \bigcap S$, i.e., $x \subseteq \bigcap S$. Thus, $\bigcap S$ is an ordinal. Suppose $\bigcap S \notin S$. Then, $\bigcap S \in y$ for all $y \in S$ since $\bigcap S \subseteq y$ and both are ordinals. Thus $\bigcap S$ lies in the intersection of all y, in symbols, $\bigcap S \in \bigcap S$, contradicting the axiom of regularity. Thus $\bigcap S \in S$. □

Proposition 17.7

For each ordinal x, $x = \bigcup_{y \in x} y$.

Proof.

Immediate by Proposition 17.5. □

Proposition 17.8

The collection of all the 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. □

Transfinite induction

Proposition 17.7 justifies

Principle 17.9 (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 then some fixed ordinal β, leading to

Principle 17.10 (Transfinite induction)

If P is a property, and, assuming that P holds for every ordinal less than $\alpha < \beta$, we can prove that P holds for α, then P holds for any ordinal less than β.

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 remarking that the set of natural numbers is in bijection with ω, the ordinal containing $\emptyset$ and closed under the successor operation.

Transfinite induction

Principle 17.11 (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 for first order logic.

Also, notice how the usual induction principle on natural numbers is equivalent to the transfinite induction principle relativised to ω.

Ordinal arithmetic

Definition 17.12 (Ordinal addition)

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, i.e., such that $x \leq y$ in S implies $h(x) \leq h(y)$ in $\alpha + \beta$, where $S = \langle A \sqcup B; \leq \rangle$, the disjoint union of α and β, and $x \leq y$ if and only if x and y are both in α or in β, or $x \in \alpha$ and $y \in \beta$.

Leaving the proof that addition is properly defined as an exercise, it is easy to see that ordinal addition is associative.

Also, 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.

Ordinal arithmetic

Definition 17.13 (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 $i < j$, $i, j \in \beta$ and $x \in \alpha_i$, $y \in \alpha_j$, or $x, y \in \alpha_i$ for some $i \in \beta$ and $x \leq y$ in α.

Again, it is easy to see that multiplication is associative.

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, $2\omega = \omega$ by choosing $h(x) = 2i + x$ when $x \in 2_i$. 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.

Distributivity holds in just one direction: $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$. But $(\beta + \gamma)\alpha \neq \beta\alpha + \gamma\alpha$, in fact, $(1 + 1)\omega = 2\omega = \omega \neq 1\omega + 1\omega = \omega + \omega$.

References

The content of this lesson derives from the presentation in *Kenneth Kunen*, Set Theory: An Introduction to Independence Proofs, Studies in Logic and the Foundations of Mathematics 102, Elsevier, (1980), ISBN 0-444-86839-9.

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 18

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Set theory:

- Cardinals
- Induction
- Arithmetic

Cardinals

Definition 18.1 (Set ordering)

A set A is less or equal than the set B when there is $f: A \to B$ injective. Equality is achieved when A is in bijection with B.

Although the notion holds for any set, it is particularly relevant when applied to ordinals. In that case, we will write $\alpha \leq \beta$ to say that the ordinal α is less or equal than the ordinal β.

It is essential to note that $\alpha \leq \beta$ is **not** the same ordering used to define ordinals, i.e., $\in \cup =$. For example, $\omega + 1 \not\in \omega$ and $\omega + 1 \neq \omega$ as ordinals, but $\omega + 1 \leq \omega$, and, in fact, $\omega + 1 \cong \omega$.

Definition 18.2 (Cardinal)

A *cardinal* α is an ordinal such that, for all $\beta \in \alpha$, $\beta < \alpha$ strictly.

Notice how any cardinal is necessarily a limit ordinal.

Definition 18.3

Let α be any cardinal, then 2^α is the cardinal such that $2^\alpha \cong \wp(\alpha)$.

Notice how, in the light of Cantor's Theorem, the definition is significant since $\alpha \ncong 2^\alpha$. An obvious consequence is that the collection of all cardinals is not a set.

Definition 18.4 (The aleph class function)

By transfinite induction, we define the map $\aleph$ from the class of ordinals to the class of cardinals:

$$\aleph_i = \begin{cases} \omega & \text{when } i = 0 \\ \min\left\{\beta \in 2^{\alpha+1}+1 : \aleph_\alpha < \beta\right\} & \text{when } i = \alpha+1 \\ \bigcup_{\gamma\in\lambda} \aleph_\gamma & \text{when } i = \lambda, \text{ a limit ordinal.} \end{cases}$$

Cardinals

Notice how we used almost all the properties and the definitions in the previous lesson to define the map $\aleph$.

In fact, using those results, it is immediate to show that $\aleph_i$ is a cardinal for each ordinal i.

In particular, the finite cardinals are just the natural numbers. Each infinite cardinal is in the range of the $\aleph$ map, as it is easy to show, since the ordering of cardinals is total.

So, it makes sense to extend on cardinals the successor function $f(x) = x + 1$ on finite cardinals, posing, on infinite cardinals, $\mathrm{succ}(\aleph_i) = \aleph_{i+1}$.
Notice how ordinal arithmetic appears in the index.

Naturally, a limit cardinal has the form $\aleph_\lambda$ where λ is a limit ordinal.

Induction

Evidently, applying ordinal transfinite induction through the $\aleph$ map, we get transfinite cardinal induction.

Nevertheless, the most useful form of induction in logic applies when we assume to work within a set theory with the Axiom of Choice. Since it is equivalent to requiring that every set can be well ordered, that is, if A is a set, then there is a map $A \mapsto |A|$ that uniquely associates A with a cardinal, we may apply transfinite induction to $|A|$ to deduce properties valid for every element in A.

It is worth to note how the cardinality of a set is a consequence of the existence of cardinals, which provide a *measure*, **and** the Axiom of Choice, which ensures that each set can be reduced to such a measure.

In fact, transfinite induction operates by saying that each element a of a set A is in bijection with an ordinal $\alpha \in |A|$, so we use the total well ordering of $|A|$ as the structure that preserves the validity of properties—which is exactly what the induction principle requires to show.

Cardinal arithmetic

Definition 18.5

Let X and Y be two sets, then

- $|X| + |Y| = |X \sqcup Y|$, the disjoint union of X and Y;
- $|X| \cdot |Y| = |X \times Y|$, the Cartesian product of X and Y;
- $|X|^{|Y|} = \left|X^Y\right|$, the function space $Y \to X$.

It is simple to prove that addition and multiplication are associative and commutative. Also, the product distributes over the sum.

When dealing with finite sets, these rules reduce to the arithmetic of natural numbers. Also, $|X| + 0 = |X|$ and $|X| \cdot 0 = 0$ for any set X. When both X and Y are non empty, and at least one of them is infinite

$$|X| + |Y| = |X| \cdot |Y| = \max\{|X|, |Y|\} \quad .$$

Cardinal arithmetic

When dealing with function spaces, i.e., with cardinal exponentiation, a few rules can be derived:

- $|X|^0 = 1$, since there is a unique function from $\emptyset$ to any other set X;
- $0^{|X|} = 0$ when $0 < |X|$, i.e., $X \neq \emptyset$;
- $1^{|X|} = 1$;
- if $|X| \leq |Y|$, then $|X|^{|Z|} \leq |X|^{|Z|}$;
- if $1 < |X| < \aleph_0$, $1 < |Y| < \aleph_0$, i.e., X and Y are finite and contain more than one element, and $|Z| \geq \aleph_0$, i.e., Z is infinite, then $|X|^{|Z|} = |X|^{|Z|}$;
- if $|X| \geq \aleph_0$, i.e., X is infinite, and $0 < |Y| < \aleph_0$, i.e., Y is finite and non empty, then $|X|^{|Y|} = |X|$.

These rules are tacitly applied whenever one has to calculate the cardinality of sets. We already used them in the proof of the completeness theorem for first order logic.

References

The content of this lesson derives from the presentation in *Kenneth Kunen*, Set Theory: An Introduction to Independence Proofs, Studies in Logic and the Foundations of Mathematics 102, Elsevier, (1980), ISBN 0-444-86839-9.

Another reference of interest is *N. Bourbaki*, Elements of Mathematics: Theory of Sets, Springer, (1968), ISBN 978-3-540-22525-6.

Mathematical Logic

Lecture 19

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

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 pick 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**. But, when I is infinite, this is not possible.

Some important results in Mathematics require the Axiom of Choice to be proved: as a small collection of examples, take

- 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 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
- the Hartog's Theorem
- the Cartesian product of a family $\{X_i\}_{i \in I}$ of non empty sets, is non empty.

Well ordering theorem

Theorem 19.1 (Well ordering)

For any set X, $X \cong |X|$.

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 $s(\alpha) = c\left(X \setminus \{s(\beta)\colon \beta \in \alpha\}\right)$. We note that the construction must eventually stop, otherwise X would be in bijection with a proper class, the collection of all ordinals. And, moreover, s is a bijection, as it is immediate to see. By definition, $|X|$ is the least ordinal which is in bijection with X, and we now that there is 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 $X \in \mathscr{F}$, $X \cong I_X$ for some ordinal I_X, that is, there is $g_X : I_X \to X$ bijective.
Then, we can define a choice function $f : \mathscr{F} \to \bigcup_{X \in \mathscr{F}} X$ as $f(X) = g_X(\emptyset)$.

Zorn lemma

Theorem 19.2 (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 19.3 (Hartog)

If X and Y are two sets, it holds that either $|A| \leq |B|$ or $|B| \leq |A|$.

Although we are not going to prove these results, they shed some light to the meaning of the Axiom of Choice: in fact, 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 lose 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 *P. Suppes*, Axiomatic Set Theory, Dover Publishing, (1972), ISBN 0-486-61630-4.

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.

The continuum hypothesis is the main subject of the essays in *Paul J. Cohen*, Set Theory and the Continuum Hypothesis, Dover Publishing, (2008), ISBN 0-486-46921-2. 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 20

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Constructive mathematics:

- Motivation
- Intuitionistic logic
- Syntax
- Expressive power

Motivation

Consider the following

Proposition 20.1

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.

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. In fact, if $\log_2 9 = m/n$ for some $m, n \in \mathbb{N}$, then, by the properties of logarithm, $2^m = 9^n$, 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 hold 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. In particular, there is an axiom in the classical system that evidently violates the requirement. In fact, 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. So, the Law of Excluded Middle must be rejected if we adopt a notion of *proof* as the one above.

Motivation

The Law of the Excluded Middle is essential in the first proof of Proposition 20.1: it avoids the need to decide whether $\sqrt{2}^{\sqrt{2}}$ is rational or not (in fact, it is not).

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!

Motivation

In fact, what we would like to have is a logical system which allows to *calculate* the objects or the choices we have to make. In a sense, we are interested in systems where proofs are a sort of algorithms 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 formal attempt to capture in a formal system the original idea of a constructive approach to Mathematics. Practically, it is the simplest, most studied, and, in some sense, 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, formulae are formed in exactly the same way. In the first-order case, terms and formulae 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

Evidently, by definition, every proof $\pi \colon \Gamma \vdash_T A$ performed in the intuitionistic logic, i.e., without the Law of Excluded Middle, is also a valid classical proof.

So, we may think that intuitionistic logic is less expressive than classical logic: possibly, there are statement which are provable in the classical system, which 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 a classical system, because each intuitionistic proof is also a classical proof 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 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 inconsistent.

Expressive power

From another point of view, in a sense, every theorem in classical logic can be proved in intuitionistic logic, modulo a translation. The precise statement is as follows:

Definition 20.2

The *Gödel-Gentzen translation* is a map of formulae to formulae inductively defined as:

- $(\top)^N = \top$, $(\bot)^N = \bot$;
- for any A atomic, $(A)^N = \neg\neg A$;
- $(\neg A)^N = \neg (A)^N$;
- $(A \wedge B)^N = (A)^N \wedge (B)^N$;
- $(A \vee B)^N = \neg\left(\neg (A)^N \wedge \neg (B)^N\right)$;
- $(A \supset B)^N = (A)^N \supset (B)^N$;
- $(\forall x \colon s.\, A)^N = \forall x \colon s.\, (A)^N$;
- $(\exists x \colon s.\, A)^N = \neg \forall x \colon s.\, \neg (A)^N$.

Expressive power

Proposition 20.3

In classical logic, for any formula A, there is $\pi : A = (A)^N$.

Proposition 20.4

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 these theorems: 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 formulae which are classically equivalent.

References

A good introduction to the constructive way of reasoning can be found in *A.S. Troelstra* and *D. van Dalen*, Constructivism in Mathematics, volume I, Studies in Logic and the Foundations of Mathematics 121, Elsevier, (1988), ISBN 0-444-70506-6.

There are many ways to translate intuitionistic logic into classical logic. A survey can be found in *A.S. Troelstra* and *H. Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43, Cambridge: Cambridge University Press, (1996).

Mathematical Logic

Lecture 21

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Constructive mathematics:

- Heything algebra
- Semantics
- Soundness

Heyting algebra

Definition 21.1 (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 $\perp$ 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

Fact 21.2

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 21.3

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. □

Fact 21.4

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 $1/2 \vee \neg 1/2 = 1/2 \vee 0 = 1/2 \neq 1 = \top$. □

Proposition 21.5

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, $x \wedge y \leq (x \wedge y) \vee (x \wedge z)$ and $x \wedge z \leq (x \wedge y) \vee (x \wedge z)$ by definition of $\vee$. So, $y \leq (x \supset (x \wedge y) \vee (x \wedge z))$ and $z \leq (x \supset (x \wedge y) \vee (x \wedge z))$ by definition of $\supset$, 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 21.6 (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 $\mathscr{H}$, the meaning $[\![A]\!]$ of a propositional formula A is a map from the set of formulae 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 21.7 (Soundness)

If $\pi : \Gamma \vdash A$ 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 $\pi : \Delta \vdash B$, 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 $\wedge$, $\bigwedge_{D \in \Delta} [\![D]\!] \leq [\![B_1]\!] \wedge [\![B_2]\!] = [\![B_1 \wedge B_2]\!] = [\![B]\!]$.

$\hookrightarrow$

Soundness

$\hookrightarrow$ 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.
- 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 $\wedge$, $[\![C_1 \vee C_2]\!] \wedge \bigwedge_{D\in\Delta}[\![D]\!] = \bigwedge_{D\in\Delta}[\![D]\!] \leq [\![B]\!]$. $\hookrightarrow$

Soundness

↪ Proof. (iii)

- 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. (iv)

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 of $\leq$, $[\![A]\!] = \top$. □

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), ISBN 978-3-7643-6996-5.

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. For those interested the details can be found in *R. Goldblatt*, Topoi: The Categorical Analysis of Logic, Dover Publishing, (2006), ISBN 0-486-45026-0.

Mathematical Logic

Lecture 22

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Constructive mathematics:

- Completeness
- Variations on the theme

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 22.1 (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 formulae. 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

$\hookrightarrow$ 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

- $[A]_\sim \leq [A]_\sim$ 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:

- $\bot = [\bot]_\sim$, in fact, $\bot \vdash A$ for any formula A by $\bot$-elimination, so $[\bot]_\sim \leq [A]_\sim$;
- $\top = [\top]_\sim$, in fact, $A \vdash \top$ for any formula A by $\top$-introduction, so $[A]_\sim \leq [\top]_\sim$.

$\hookrightarrow$

↪ Proof. (iii)

Moreover, $\langle H; \leq \rangle$ is a lattice:

- $[A]_\sim \wedge [B]_\sim = [A \wedge B]_\sim$, in fact, $A \wedge B \vdash A$ and $A \wedge B \vdash B$ by $\wedge$-elimination, 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 $\wedge$-introduction, that is, $[C]_\sim \leq [A \wedge B]_\sim$;
- $[A]_\sim \vee [B]_\sim = [A \vee B]_\sim$, in fact, $A \vdash A \vee B$ and $B \vdash A \vee B$ by $\vee$-introduction, 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 $\vee$-elimination, that is, $[A \vee B]_\sim \leq [C]_\sim$.

Finally, $\langle H; \leq \rangle$ is a Heyting algebra: $[A]_\sim \supset [B]_\sim = [A \supset B]_\sim$, in fact, $A \wedge (A \supset B) \vdash B$ by $\supset$-elimination, 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 $C \vdash A \supset B$ by $\supset$-introduction , that is $[C]_\sim \leq [A \supset B]_\sim$. It is worth noticing that $\neg[A]_\sim = [\neg A]_\sim$ since $\vdash \neg A = (A \supset \bot)$. ↪

Completeness

$\hookrightarrow$ Proof. (iv)

Let $\nu \colon V \to H$ be $\nu(x) = [x]_\sim$ for any variable x.

By induction on the structure of A, we prove that $\llbracket A \rrbracket = [A]_\sim$ in $((H; \leq), \nu)$:

- if $A \equiv x$, a variable, by definition $\llbracket A \rrbracket = \nu(x) = [x]_\sim = [A]_\sim$;
- if $A \equiv \top$, $\llbracket A \rrbracket = \top = [\top]_\sim$;
- if $A \equiv \bot$, $\llbracket A \rrbracket = \bot = [\bot]_\sim$;
- if $A \equiv B \wedge C$, by induction hypothesis,
  $\llbracket A \rrbracket = \llbracket B \rrbracket \wedge \llbracket C \llbracket = [B]_\sim \wedge [C]_\sim = [B \wedge C]_\sim = [A]_\sim$;
- if $A \equiv B \vee C$, by induction hypothesis,
  $\llbracket A \rrbracket = \llbracket B \rrbracket \vee \llbracket C \llbracket = [B]_\sim \vee [C]_\sim = [B \vee C]_\sim = [A]_\sim$;
- if $A \equiv B \supset C$, by induction hypothesis,
  $\llbracket A \rrbracket = \llbracket B \rrbracket \supset \llbracket C \llbracket = [B]_\sim \supset [C]_\sim = [B \supset C]_\sim = [A]_\sim$;
- if $A \equiv \neg B$, by induction hypothesis, $\llbracket A \rrbracket = \neg \llbracket B \rrbracket = \neg [B]_\sim = [\neg B]_\sim = [A]_\sim$. $\hookrightarrow$

Completeness

$\hookrightarrow$ 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 $A \sim \top$, that is $\vdash A \supset \top$ and $\vdash \top \supset A$. By $\top$-introduction and $\vdash \top \supset A$, we get that $\vdash A$. □

Variations on the theme

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;
- Kripke semantics;
- logical categories.

Variations on the theme

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 that these extreme elements are generated by the relation of algebras, which models the elimination of the quantified variable.

Variations on the theme

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 Grotendieck 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, and their classical set-theoretic version is known as *Kripke semantics*, and it is usually built up from the usual set theory.

Variations on the theme

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*.

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.

References

The proof of the completeness theorem has been adapted from the categorical version in *P. Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002), ISBN 978-0-19-853425-9 and 978-0-19-851598-2.

Heyting categories are defined in the same text. The internal logic of a topos has been introduced by W. Lawvere, and an approachable text is *R. Goldblatt*, Topoi: The Categorical Analysis of Logic, Dover Publishing, (2006), ISBN 0-486-45026-0.

On the contrary, logic categories have been introduced by M. Benini, and a quite technical survey can be found in *M. Benini*, Intuitionistic First-Order Logic: Categorical Semantics via the Curry-Howard Isomorphism, arXiv.org (2013), `http://arxiv.org/abs/1307.0108`.

Mathematical Logic

Lecture 23

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Constructive mathematics:

- Computable functions
- The simple theory of types
- λ calculus

Computable functions

A function is *computable* when there is a mechanical procedure which transforms the argument in the value.

A few remarks are due:

- we assume a sort of machine able to represent the argument, the result and the procedure;
- we assume that such a machine can execute the procedure, given the argument, without any external aid;
- we assume the machine could operate in the real world, at least in principle;
- as a consequence, we assume to have at disposal only a finite amount of resources, usually execution time and space, e.g., in the form of memory;
- we assume the argument, the result, and the procedure to be perfectly observable, that is, we can know their values with no degree of imprecision, uncertainty or error.

Computable functions

The previous assumptions are committing: dropping one of them, we get a different notion of computability.

These alternative notions have been studied, but they have no actual application, since no one has been able, up to now, to construct a physical machine outside the bounds we have sketched above. On the contrary, real computing machines of various sort have been realised that meet all those constraints.

It is evident that the constraints we posed on computability are not of a mathematical character. As performing extensive computation gained importance, mathematicians started to formalise system to capture the notion in a mathematical form.

Computable functions

Although the description of what is a computable function is informal, it still allows to conclude that there are functions which cannot be computed:

- since the representation must be finite, it can be reduced to an expression in a language over a denumerable alphabet.
- any function $\mathbb{R} \to \mathbb{R}$ cannot be computed. In fact, since the set of real numbers has a cardinality of $2^{\aleph_0} > \aleph_0$, the argument is not, in general, representable. The same holds for the result.
- the functions $\mathbb{N} \to \{0,1\}$ form a set whose cardinality is $2^{\aleph_0}$, so there is no way to have a language within the constraints that can represent a procedure to compute all of them, even if both the argument and the result are representable.

Computable functions

Despite the apparently informal description, the notion of computable function is surprisingly stable: there are numerous ways to formalise it, and all of them identify as computable the same class of functions $\mathbb{N} \to \mathbb{N}$.

In this sense, those formal systems are equivalent.

Among these formal systems, the λ-calculus has a special interest for logicians. In fact, coupling an extremely simple formal syntax with an elementary way to perform computations, i.e., the “machine” it uses to calculate is completely formal and really elementary, the λ-calculus is easier to analyse than many other systems.

Moreover, considering its typed version, which we will do in the following, it formally justifies the claim that intuitionistic logic is the logic of computation, since there is a strict correspondence between the logic and the type system.

Simple theory of types

Definition 23.1 (Type)

Fixed a denumerable set V_T of *type variables*, a *type* is inductively defined as follows:

- $x \in V_T$ 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 in the theory of types, the *term*.

Simple theory of types

Definition 23.2 (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, $\square_\alpha\colon 0 \to \alpha$ is a term and $FV(\square_\alpha\colon 0 \to \alpha) = \emptyset$;
- 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 $\mathrm{FV}(\pi_1 A\colon \alpha) = \mathrm{FV}(\pi_2 A\colon \beta) = \mathrm{FV}(A\colon \alpha \times B)$;

$\hookrightarrow$

Simple theory of types

$\hookrightarrow$ (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.

Simple theory of types

Terms can be *reduced* according to the following rules, where it is assumed that both sides of the equalities are correctly typed:

- $\pi_1\langle A, B\rangle = A$;
- $\pi_2\langle A, B\rangle = B$;
- $\langle \pi_1 A, \pi_2 A\rangle = A$;
- $(\lambda x : a. A) \cdot B = A[B/x]$, the act of substituting B for x;
- $\lambda x : \alpha. (A \cdot x) = A$, when $x : \alpha \not\in \mathrm{FV}(A : \alpha \to \beta)$;
- $\delta(i_1 C, A, B) = A \cdot C$;
- $\delta(i_2 C, A, B) = B \cdot C$.

It is clear that these rules satisfy the requirements on computable functions.

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 $\lambda x : \alpha . A : \alpha \rightarrow \beta$, called *abstractions*, $A \cdot B : \beta$, called *applications*, $\langle A, B \rangle : \alpha \times \beta$, called *pairs*, $\pi_1 A : \alpha$ and $\pi_2 A : \beta$, called *projections*, we get a subsystem of special interest.

In fact, if we interpret $\times$ as the Cartesian product, and $\rightarrow$ as the function space, we can easily derive 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.

In Computer Science, this subsystem is the core of most functional programming languages. Admitting recursive definitions, it becomes a system that is computationally complete, i.e., able to represent any computable function.

λ-calculus

In the subsystem of the previous slide, when we forget types and we allow to freely construct terms, we get another formal system: the λ-calculus.

This untyped system is extremely powerful: it allows to describe recursion via definable fixed-point operations, and, as in the typed system, so it is computationally complete.

Also, in the λ-calculus, we can simulate the terms we omitted in the typed subsystem: those related to $+$, informally standing for the disjoint union, 0, the *unit* type, informally the empty type, and 1, the *universal* type, informally standing for the entire universe.

It can be shown that the λ-calculus enjoys a number of very strong reduction properties, like every term has at most one *normal form*, i.e., a term that cannot be reduced any further. It can be shown that there is a reduction strategy that eventually reduces every term to its normal form, if there is one.

These properties ensure that the λ-calculus is a good computational system, and it can be thought of as a deterministic computational machine.

Although it is beyond the scope of this course to develop the branch of mathematical logic called Computability Theory, it is worth remarking that we may think of the simple theory of types, or any other type system, as a λ-calculus with no types, providing the 'computational engine', and an algebra of types, constraining the terms which are acceptable in the system, a sort of weak semantic decoration to discriminate well-behaved procedures from ill-formed ones.

Finally, it is possible to prove that each term in the simple theory of types eventually reduces to its normal form in a finite number of reduction steps in the λ-calculus, i.e., each computation terminates with a unique result.

This are good news since we cannot construct programs which are too ill-behaved in the simple theory of type, but they are also bad news since it clearly shows that the simple theory of types is unable to compute all the computable functions: in fact, some of them do not terminate on every input.

References

Computability theory, also known as recursion theory is a major branch of mathematical logic. A very nice introductory text is *S.B. Cooper*, Computability Theory, Chapman & Hall/CRC Mathematics, (2004), ISBN 1-58488-237-9.

A classical, and still excellent introduction to λ-calculus and the simple theory of types is *J.R. Hindley* and *J.P. Seldin*, Lambda-Calculus and Combinators, Cambridge University Press, (2008), ISBN 978-0-521-89885-0.

The link between logical systems, their semantics, and the simple theory of types is illustrated in *P. Johnstone*, Sketches of an Elephant: A Topos Theory Compendium, two volumes, Oxford University Press (2002), ISBN 978-0-19-853425-9 and 978-0-19-851598-2.

Mathematical Logic

Lecture 24

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Constructive mathematics:

- Propositions as types
- Proofs, computationally
- Computations, logically
- Variations on the theme

Propositions as types

If we put side by side propositional logical formulae and types in the simple theory of types, we get:

types	formulae
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 propositional proof in the intuitionistic natural deduction system, and terms in the simple theory of types, we get:

proof	assumption	$\top I$	$\bot E$	$\wedge I$	$\wedge E_{1,2}$	$\vee I_{1,2}$	$\vee E$	$\supset I$	$\supset E$
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

Let examine a couple of examples:

- if $A\colon \alpha$ and $B\colon \beta$ are terms, so is $\langle A, B\rangle\colon \alpha \times \beta$ becomes

$$\dfrac{\begin{array}{c}\vdots A \\ \alpha\end{array} \qquad \begin{array}{c}\vdots B \\ \beta\end{array}}{\alpha \wedge \beta}\,\wedge\mathrm{I}$$

- if $A\colon \beta$ is a term and $x\colon \alpha$ a variable, then $\lambda x\colon \alpha . A\colon \alpha \to \beta$ becomes

$$\dfrac{\begin{array}{c}[\alpha]^{*} \\ \vdots A \\ \beta\end{array}}{\alpha \supset \beta}\,\supset\mathrm{I}^{*}$$

where the label $*$ stands for x.

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 formulae as linear expressions instead of adopting the tree-like syntax of natural deduction.

But the logical syntax is coupled with a semantics, and the type theory with a computational meaning, given by the reduction rules.

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. Although it is too long and complex to precisely state and prove the result here, one consequence is that every proof in normal form consists of a sequence of elimination rules followed by a series of introduction rules.

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 $\vee I_1$ or $\vee I_2$. 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 $\exists I$. Hence, the conclusion of the last but one step would be either $A[t/x]$ for some term t, providing a witness to the existential statement.

Computations, logically

Since every formal proof in intuitionistic logic corresponds to a typed term, and typed terms 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.

Variations on the theme

The simple theory of types is just the simplest type theory: many other systems have been analysed, and many of them have a propositions-as-types interpretation, computationally characterising some logical system.

In some cases, like in the constructive type theory, the corresponding logical system is part of the type theory itself. This reflection allows to use such a system to describe mathematical theories, like set theory, inside the type system, becoming part of it. Thus, the type system acts as a *universal* theory, which contains the whole mathematics representable in its logical counterpart.

This way of proceeding has recently lead to a promising approach, which explains computation in terms of algebraic topology (and vice versa). It is called *homotopy type theory*, and it is part of the contemporary frontier of mathematical research. The basic idea is that, by adding a pair of axioms to constructive type theory, one can interpret a computation as a path in some homotopy space. It turns out that paths which are homotopy equivalent can be represented by the same term. Of course, behind this intuition the formal theory is somewhat involved, and not yet completely stable...

References

The propositions-as-types interpretation is illustrated in many textbooks: the lesson has been adapted from *A.S. Troelstra* and *H. Schwichtenberg*, Basic Proof Theory, Cambridge Tracts in Theoretical Computer Science 43, Cambridge: Cambridge University Press, (1996).

A more computer science oriented text is *B.C. Pierce*, Types and Programming Languages, The MIT Press, (2002), ISBN 978-0262-16209-8.

Homotopy Type Theory is introduced in *The Univalent Foundation Program*, Homotopy Type Theory, Institute of Advanced Studies, Princeton, (2013). The chapter on type theory is also a very nice introduction to the theory of dependent types, which extends the simple theory theories in a natural and constructive way.

Mathematical Logic

Lecture 25

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- Review of completeness
- Set theory
- Inconsistent theories
- Effective theories
- Expressive power

Review of completeness

The completeness result for first order classical logic says that, whenever a formula A is true in every model for a theory T and under any interpretation of variables, then it is provable from T.

This fact establishes a symmetry between provability and truth: every formula which is true *everywhere*, is provable and vice versa.

This opens the possibility to have formulae which are true in some model and false in other models, so they cannot be proved. And their negation would be true in some model and false in others, so they cannot be proved, too.

In particular, in some cases, we would like to have a theory which captures exactly the features of a *specific* model, for example, arithmetic should speak exactly of natural numbers. In that case, we would expect that each formula which is true on natural numbers to be provable from the theory.

Review of completeness

Consider two example: the theory of groups and arithmetic. Although it is not difficult to formalise these theories in the first order language, they are very different.

The theory of groups is deputed to describe all groups, while arithmetic is deputed to describe what are the true sentences on natural numbers. Both theories are interested in characterising the true sentences, but in one case, the intended interpretation is not fixed upon a single model, while in the other we would like to speak of a single, specific model.

We know, since the canonical model of the Completeness Theorem has the same cardinality of the language, that there is no hope the describe exactly a single model when its universe is big enough, for example, mathematical analysis cannot describe exactly the model of real numbers.

Review of completeness

The aim of this and the following lessons is to answer a simple question:

> *Is it possible to define a theory which captures exactly the true formulae in a fixed model?*

An obvious answer is positive: we may take as the theory T, the collection of all the true sentences in the model of interest. In that case, obviously, T is consistent and it allows to deduce all the true sentence in the model trivially, being them axioms. Since no other sentence are true in that model, by the Soundness Theorem, it follows that T does not allow to derive any other sentence.

Clearly, this is not a good answer: what we are really asking for, is a theory T which is consistent, *effective*, and sufficiently powerful.

Review of completeness

Actually, the proof of the Completeness Theorem 14.3, already contains a negative answer: since the canonical model has the same cardinality as the language, and it contains every formula or its negation, it follows that we cannot write a first order theory which captures exactly the truth content of a model whose cardinality is greater than the language.

For example, there is no hope to write, within a finite or denumerable language, a complete theory which defines exactly the theory of real numbers. In fact, such a theory would have, at least, a denumerable model, the canonical one, while the reals form a set whose cardinality is $2^{\aleph_0}$.

Review of completeness

On the other side, when we limit ourselves to finite models, we can describe exactly one single model, up to isomorphisms. We just need to write an axiom which lists all the element of the universe, and which says that no other element could exist.

In this case, our aim is to find an effective theory that, including this axiom, allows us to derive all the true sentences in that model. Since we can express the fact that a model must have a finite cardinality, all the models will be, in fact, isomorphic.

It is clear that there is a boundary: denumerable models. As a matter of fact, model theory already provides a negative answer: we cannot characterise a single model, since every theory which has an infinite model, must have models of any infinite cardinality. This result, we are not going to state precisely and to prove, is known as the *Löwenheim-Skolem theorem(s)*.

So, the question should not be intended as trying to find a theory which identifies an unique model, up to isomorphisms. Rather, we are searching for a consistent, effective theory which is able to prove all the true sentences in a model, thus, because the classical language has negation, and because of the Law of Excluded Middle, we will know all the false sentences, too.

In other words, we are searching for a theory which proves every sentence or its negation. And we require this theory to be consistent, which means the previous “or” to be exclusive, and we require this theory to be effective, which means we can really write it down, decide which formulae are axioms, and we can really construct proofs.

We already encountered a negative example of this phenomenon: **ZFC** is an effective theory, in the sense that we can write down its axioms, and, given a formula, we may decide whether it is an instance of an axiom or not. Also, it is based on a finite signature, so, by the Completeness Theorem, it has a denumerable model.

We have said that there are principles, like the Continuum Hypothesis, which cannot be proved or disproved in **ZFC**. This lead to a variety of notions of sets, each one corresponding to a different intended model.

In some cases, this are good news: the mathematical universe is richer, and we can choose, from time to time, which notion of set better suits our needs.

Effective theories

An essential requirement in the development of a mathematical theory, is that it must be *effective*.

Informally, it means that we must be able to write down formulae, and to carry on deductions.

So, we need a finite or denumerable signature, since it is evident that we cannot really write more than $\aleph_0$ symbols!

Also, we need an algorithm that, given a formula, tells us whether it is an axiom. In technical terms, a theory must be *decidable*, or *recursive*. Coupled with the natural deduction system, this fact allows to really write the formal proofs, thus enabling the theory to be used.

A theory meeting these criteria is called *effective*. In logic, we may be interested also in non effective theories, like the collection of all the true formulae in a model.

Expressive power

Since effective theories are a small part of all the possible theories, it becomes significant to ask whether an effective theory has enough expressive power to deduce all the truths in a domain of interest.

This is the real question behind the results we will discuss in the following lessons. . .

References

This lesson encompasses different results from many areas of logic: the incompleteness of set theory has already been discussed when dealing with the Axiom of Choice, for example.

The limit induced by the proof of the Completeness Theorem are folklore. The notion of effective theory, and its relation to computability was, more or less, understood by Gödel, but it required some time to get polished.

A discussion of all these aspect can be found in *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), ISBN 0-7204-28440. However, one should keep in mind (and it appears in the text) that no *natural* example of undecidable proposition in arithmetic was known at the time that book was written. Nowadays, there are many.

Mathematical Logic

Lecture 26

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- Peano arithmetic
- Induction
- Standard and non-standard models

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

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

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 define addition and multiplication:

$$\forall x.\, 0 + x = x \quad ; \tag{3}$$

$$\forall x, y.\, S\,x + y = S(x + y) \quad ; \tag{4}$$

$$\forall x.\, 0 \cdot x = 0 \quad ; \tag{5}$$

$$\forall x, y.\, S\,x \cdot y = x \cdot y + y \quad . \tag{6}$$

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 $x \in \mathbb{N}$.

Induction

There is a link between induction and recursion: an inductive definition induces a recursive procedure that allows to calculate/generate the defining objects, and vice versa, a recursive procedure induces an inductive definition of its results.

Example 26.1

The axioms (3) and (4) provide a recursive schema that allows to calculate the addition:

$$x + y = \text{if } x = 0 \text{ then } y \text{ else let } x = S\,z \text{ in } S(z + y) \;\; ;$$

Conversely, we may say that the result of the sum is identified by induction of the first summand.

Standard model

The standard 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 confusing, we adopt the standard 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 26.2 (Non-standard model)

Any structure $\mathscr{N}$ on the language of Peano arithmetic which is not isomorphic to the standard model $\mathscr{M}$ but, for any evaluation σ of variables is a model $(\mathscr{N},\sigma)$ of Peano arithmetic, is called a *non-standard model*.

In the definition above, an isomorphism between structures $f:\mathscr{N}\to\mathscr{M}$ is

- an invertible function between the universes;
- for each term t, $f(\llbracket t\rrbracket_{\mathscr{N}}) = \llbracket t\rrbracket_{\mathscr{M}}$.

If a non-standard model exists, it means that there is a structure $\mathscr{N}$ which makes Peano arithmetic true but interprets some term into an element e in the universe which cannot be mapped in some natural number.

Notice that the element e must be the image of a term under the interpretation function: so, for example, the real numbers consisting of all the non-negative integers, is **not** a non-standard model, even if it is constructed in a very different way from the naturals (all the reals are a quotient of Cauchy sequences).

Non-standard models

Proposition 26.3

There is a non-standard model for Peano arithmetic.

Proof. (i)

Define $S^0(0) = 0$, and $S^{i+1}(0) = S\,S^i(0)$. Evidently the term $S^n(0)$ gets interpreted in n in any model.

Let $\Sigma_n = \{x \neq S^i(0) : i < n\}$ be a collection of formulae, and let $\Sigma = \bigcup_{n \in \mathbb{N}} \Sigma_n$.

Calling $\mathscr{M}$ the structure of the standard model, and defining σ_n such that $\sigma_n(x) = n$, evidently the standard model $(\mathscr{M}, \sigma_n)$ makes Σ_n valid, together with all the axioms of Peano arithmetic.

Thus, any finite $\Xi \subset \Sigma$ has a model, because it is contained in Σ_n for some n. Thus, by the Compactness Theorem 14.4, Σ has a model $(\mathscr{N}, \sigma)$ which makes true also all the axioms of Peano arithmetic. $\hookrightarrow$

Non-standard models

$\hookrightarrow$ Proof. (ii)

In this model, $\sigma(x) \neq n$ for any $n \in \mathbb{N}$ because $[\![S^n(0)]\!]_{\mathcal{N}} = n$ but $x \neq S^n(0)$ occurs in Σ, so, by definition of interpretation, $\sigma(x) \neq [\![S^n(0)]\!]_{\mathcal{N}}$.

Hence, there is an element $k \notin \mathbb{N}$ such that $\sigma(x) = k$. But interpreting x on $\mathscr{M}$ leads to some $n \in \mathbb{N}$, whatever evaluation of variables we may choose. So, any function mapping $\mathscr{N}$ to $\mathscr{M}$ has to be non-invertible on the term x.

Thus, $(\mathscr{M}, \sigma)$ is a model of Peano arithmetic, which is not isomorphic to any standard model, so it is non-standard. □

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 $(\mathscr{M},\sigma)$ 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 26.3 to write a formula which holds in the non-standard model while is does not in any standard model. In fact, we used this property to synthesise the non-standard model from the standard ones.

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 26.3.

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 formula ϕ does not depend on the evaluation of variables, thus its truth variable would be defined by the structure of the model only. In a sense, 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, thus, 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 a single axiom, ϕ, or its complement, $\neg\phi$.

References

Peano arithmetic is illustrated in most textbooks about logic.

The relation between induction and recursion is deep and complex: an introduction to this is beyond the scope of the present course. The interested reader could refer to *Benjamin C. Pierce*, Types and Programming Languages, MIT Press, (2002), ISBN 978-0-262-16209-8.

The existence of non-standard models can be shown in many different ways. Proposition 26.3 is adapted from *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), ISBN 0-7204-28440.

Mathematical Logic

Lecture 27

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- Gödel's First Incompleteness Theorem
- The idea behind the proof
- Coding terms
- Coding formulae

Induction, again

The induction principle says that, fixed a property $P \subseteq \mathbb{N}$, if $0 \in P$ and, for any $n \in \mathbb{N}$, if $n \in P$ then $n+1 \in P$, then $P = \mathbb{N}$.

Clearly, the induction schema (7) in Peano arithmetic is just an approximation of the real induction principle: since $\left|\wp(\mathbb{N})\right| = 2^{|\mathbb{N}|}$ while the collection of formulae on the language of arithmetic has cardinality $|\mathbb{N}|$, we have not enough formulae to represent all the possible properties.

The gap between what can be formalised and what is the intended meaning about the structure of natural numbers, the induction principle at the first place, is responsible for non-standard models.

Theorem 27.1 (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 $T \nvdash G$ and $T \nvdash \neg G$.

There are a number of facts which must be clarified before proving the theorem:

- when a theory is *effective*;
- what is a *recursive* function;
- what is meant by representing all the recursive functions.

In fact, all these concepts are strictly related to each other, and they descend from the theory of computable functions.

Computable functions, again

An alternative but equivalent definition of computable functions is due to Kleene.

Definition 27.2 (Recursive function)

A *recursive function* is a member of the minimal class $\mathscr{R}$ of partial functions $f : \mathbb{N}^n \to \mathbb{N}$ for some $n \in \mathbb{N}$ such that

- $(\lambda x_1, \ldots, x_n . 0) \in \mathscr{R}$ for any n;
- $S \in \mathscr{R}$, the successor function is recursive;
- $(\lambda x_1, \ldots, x_n . x_i) \in \mathscr{R}$ for any $n \in \mathbb{N}$ and any $1 \leq i \leq n$;
- if $h_1, \ldots, h_n : \mathbb{N}^m \to \mathbb{N}$ and $g : \mathbb{N}^n \to \mathbb{N}$ are recursive, so is their *composition* $\lambda x_1, \ldots, x_m . g(h_1(x_1, \ldots, x_m), \ldots, h_n(x_1, \ldots, x_m))$;

↪

Computable functions, again

↪ (Recursive function)

- if $g\colon \mathbb{N}^m \to \mathbb{N}$ and $h\colon \mathbb{N}^{m+2} \to \mathbb{N}$ are recursive, so is the unique function $f\colon \mathbb{N}^{m+1} \to \mathbb{N}$ defined by *primitive recursion*, that, the solution to the pair of equations

$$f(x_1,\ldots,x_m,0) = g(x_1,\ldots,x_m)$$
$$f(x_1,\ldots,x_m,S\,y) = h(x_1,\ldots,x_m,y,f(x_1,\ldots,x_m,y))$$

  for any $x_1,\ldots,x_m \in \mathbb{N}$;
- if $g\colon \mathbb{N}^{m+1} \to \mathbb{N}$ is recursive, so is $f\colon \mathbb{N}^m \to \mathbb{N}$, defined by *minimalisation*, that is, the partial function which takes the value of the minimal element of $\{y \in \mathbb{N} \colon g(x_1,\ldots,x_m,y) = 0\}$ whenever this set is inhabited, and is undefined otherwise.

The minimal class of functions satisfying all the points above except for minimalisation is called the class of *primitive recursive* functions, and all its members are total, i.e., proper functions.
Finally, a set X is *recursive* when it characteristic function is both recursive and total.

Effective theories and coding

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, or, more in general, of recursive functions, is a total map g from the expressions of the syntax (terms, formulae, proofs) to $\mathbb{N}$ such that

- g is injective;
- g is recursive;
- g^{-1} on the image of g is recursive, too.

Proposition 27.3

For each recursive set R, there is formula A in Peano arithmetic such that, for each closed term t, $\vdash A[t/x]$ if and only if $[\![t]\!] \in R$ in the standard model.

We are not going to prove this result, which requires to develop a few results in computability theory which lie beyond the scope of this course.

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, formulae 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 27.4 (Coding terms)

The *Gödel's coding function g* on terms is inductively defined as follows:

- $g(0) = 2 \cdot 3$;
- $g(x_i) = 2 \cdot 3^2 \cdot 5^{i+1}$;
- $g(S\,t) = 2 \cdot 3^3 \cdot 5^{g(t)}$;
- $g(t+2) = 2 \cdot 3^4 \cdot 5^{g(t)} \cdot 7^{g(s)}$;
- $g(t \cdot s) = 2 \cdot 3^5 \cdot 5^{g(t)} \cdot 7^{g(s)}$.

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, intuitively, it is possible to write a formula in Peano arithmetic that tells whether its argument is a code of a term. This can be formalised by showing that the set of codes for terms is recursive, so that Proposition 27.3 yields the result.

Coding formulae

Definition 27.5 (Coding formulae)

The *Gödel's coding function* g on formulae 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 formulae

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 formulae 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, intuitively, it is possible to write a formula in Peano arithmetic that tells whether its argument is a code of a formula. This can be formalised by showing that the set of codes for formulae is recursive, so that Proposition 27.3 yields the result.

Coding sequences

Definition 27.6 (Coding finite sequences)

The *Gödel's coding function* g of a finite sequence $n_1, \ldots, n_k$ of natural numbers is $g(n_1, \ldots, n_k) = 2^3 \cdot \prod_{1 \leq i \leq k} p_{i+1}^{n_1}$, 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 formulae, and the set of codes for sequences can be represented, in the sense of Proposition 27.3, by some formula of Peano arithmetic, specifically by $\exists y . x = SSSSSSSS0 \cdot y$.

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), ISBN 0-7204-28440.
Nevertheless, the lecture has been prepared roughly following some unpublished notes from the course held by Silvio Valentini in 1991.

Mathematical Logic

Lecture 28

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- Coding proofs
- Numerals
- Representation

Coding proofs

Definition 28.1 (Coding proofs)

The *Gödel's coding function* g on proofs extends the previous coding g and it is inductively defined as:

- $g\left(\dfrac{\pi_1 : \Gamma \vdash A \quad \pi_2 : \Gamma \vdash B}{A \wedge B}\,\wedge\mathrm{I}\right) = 2^4 \cdot 3 \cdot 5^{g(\pi_1 : \Gamma \vdash A)} \cdot 7^{g(\pi_2 : \Gamma \vdash B)} \cdot 13^{g(A \wedge B)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash A \wedge B}{A}\,\wedge\mathrm{E}_1\right) = 2^4 \cdot 3^2 \cdot 5^{g(\pi : \Gamma \vdash A \wedge B)} \cdot 13^{g(A)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash A \wedge B}{B}\,\wedge\mathrm{E}_2\right) = 2^4 \cdot 3^3 \cdot 5^{g(\pi : \Gamma \vdash A \wedge B)} \cdot 13^{g(B)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash A}{A \vee B}\,\vee\mathrm{I}_1\right) = 2^4 \cdot 3^4 \cdot 5^{g(\pi : \Gamma \vdash A)} \cdot 13^{g(A \vee B)}$;
- $g\left(\dfrac{\pi : \Gamma \vdash B}{A \vee B}\,\vee\mathrm{I}_2\right) = 2^4 \cdot 3^5 \cdot 5^{g(\pi : \Gamma \vdash B)} \cdot 13^{g(A \vee B)}$;

$\hookrightarrow$

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{\pi_1 : \Gamma \vdash A \vee B \quad \pi_2 : \Gamma, A \vdash C \quad \pi_3 : \Gamma, B \vdash C}{C} \vee\mathrm{E}\right) =$
  $2^4 \cdot 3^6 \cdot 5^{g(\pi_1 : \Gamma \vdash A \vee B)} \cdot 7^{g(\pi_2 : \Gamma, A \vdash C)} \cdot 11^{g(\pi_3 : \Gamma, B \vdash C)} \cdot 13^{g(C)}$;
- $g\left(\dfrac{\pi : \Gamma, A \vdash B}{A \supset B} \supset\mathrm{I}\right) = 2^4 \cdot 3^7 \cdot 5^{g(\pi : \Gamma, A \vdash B)} \cdot 13^{g(A \supset B)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash A \supset B \quad \pi_2 : \Gamma \vdash A}{B} \supset\mathrm{E}\right) = 2^4 \cdot 3^8 \cdot 5^{g(\pi_1 : \Gamma \vdash A \supset B)} \cdot 7^{g(\pi_2 : \Gamma \vdash A)} \cdot 13^{g(B)}$;
- $g\left(\dfrac{\pi : \Gamma, A \vdash \bot}{\neg A} \neg\mathrm{I}\right) = 2^4 \cdot 3^9 \cdot 5^{g(\pi : \Gamma, A \vdash \bot)} \cdot 13^{g(\neg A)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash \neg A \quad \pi_2 : \Gamma \vdash A}{\bot} \neg\mathrm{E}\right) = 2^4 \cdot 3^{10} \cdot 5^{g(\pi_1 : \Gamma \vdash \neg A)} \cdot 7^{g(\pi_2 : \Gamma \vdash A)} \cdot 13^{g(\bot)}$;
- $g\left(\dfrac{}{\top} \top\mathrm{I}\right) = 2^4 \cdot 3^{11} \cdot 13^{g(\top)}$;

↪

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{\pi\colon \Gamma \vdash \bot}{A}\ \bot\mathrm{E}\right) = 2^4 \cdot 3^{12} \cdot 5^{g(\pi\colon \Gamma\vdash\bot)} \cdot 13^{g(A)}$;
- $g\left(\dfrac{}{A \vee \neg A}\ \mathrm{lem}\right) = 2^4 \cdot 3^{13} \cdot 13^{g(A\vee\neg A)}$;
- $g\left(\dfrac{\pi\colon \Gamma \vdash A}{\forall x.\,A}\ \forall\mathrm{I}\right) = 2^4 \cdot 3^{14} \cdot 5^{g(\pi\colon \Gamma\vdash A)} \cdot 13^{g(\forall x.\,A)} \cdot 19^{g(x)}$;
- $g\left(\dfrac{\pi\colon \Gamma \vdash \forall x.\,A}{A[t/x]}\ \forall\mathrm{E}\right) = 2^4 \cdot 3^{15} \cdot 5^{g(\pi\colon \Gamma\vdash\forall x.\,A)} \cdot 13^{g(A[t/x])} \cdot 17^{g(t)} \cdot 19^{g(x)}$;
- $g\left(\dfrac{\pi\colon \Gamma \vdash A[t/x]}{\exists x.\,A}\ \exists\mathrm{I}\right) = 2^4 \cdot 3^{16} \cdot 5^{g(\pi\colon \Gamma\vdash A[t/x])} \cdot 13^{g(\exists x.\,A)} \cdot 17^{g(t)} \cdot 19^{g(x)}$;
- $g\left(\dfrac{\pi_1\colon \Gamma \vdash \exists x.\,A \quad \pi_2\colon \Gamma, A \vdash B}{B}\ \exists\mathrm{E}\right) =$
  $2^4 \cdot 3^{17} \cdot 5^{g(\pi_1\colon \Gamma\vdash\exists x.\,A)} \cdot 7^{g(\pi_2\colon \Gamma, A\vdash B)} \cdot 13^{g(B)} \cdot 19^{g(x)}$;

↪

Coding proofs

↪ (Coding proofs)

- $g\left(\dfrac{}{\forall x. x = x}\,\text{ax}\right) = 2^4 \cdot 3^{18} \cdot 13^{g(\forall x. x=x)} \cdot 19^{g(x)}$;
- $g\left(\dfrac{}{\forall x, y. x = y \supset y = x}\,\text{ax}\right) = 2^4 \cdot 3^{19} \cdot 13^{g(\forall x,y. x=y \supset y=x)} \cdot 19^{g(x,y)}$;
- $g\left(\dfrac{}{\forall x, y, z. x = y \land y = z \supset x = z}\,\text{ax}\right) =$
  $2^4 \cdot 3^{20} \cdot 13^{g(\forall x,y,z. x=y \land y=z \supset x=z)} \cdot 19^{g(x,y,z)}$;
- $g\left(\dfrac{\pi_1 : \Gamma \vdash A[t/x] \quad \pi_2 : \Gamma \vdash t = r}{A[r/x]}\,\text{ax}\right) =$
  $2^4 \cdot 3^{21} \cdot 5^{g(\pi_1 : \Gamma \vdash A[t/x])} \cdot 7^{g(\pi_2 : \Gamma \vdash t=r)} \cdot 13^{g(A[r/x])} \cdot 19^{g(x)}$;
- $g\left(\dfrac{}{\forall x_1, \dots, x_n. \exists! z. z = f(x_1, \dots, x_n)}\,\text{ax}\right) =$
  $2^4 \cdot 3^{22} \cdot 13^{g(\forall x_1,\dots,x_n. \exists! z. z=f(x_1,\dots,x_n))} \cdot 17^{g(f(x_1,\dots,x_n))} \cdot 19^{g(x_1,\dots,x_n,z)}$;
- $g\left(\dfrac{}{\forall x. S x \neq x}\,\text{ax}\right) = 2^4 \cdot 3^{23} \cdot 13^{g(\forall x. Sx \neq x)} \cdot 19^{g(x)}$;

↪

Coding proofs

↪ (Coding proofs)

- $g\left(\overline{\forall x,y.\,Sx=Sy\supset x=y}^{\;\text{ax}}\right)=2^4\cdot 3^{24}\cdot 13^{g(\forall x,y.\,Sx=Sy\supset x=y)}\cdot 19^{g(x,y)}$;
- $g\left(\overline{\forall x.\,0+x=x}^{\;\text{ax}}\right)=2^4\cdot 3^{25}\cdot 13^{g(\forall x.\,0+x=x)}\cdot 19^{g(x)}$;
- $g\left(\overline{\forall x,y.\,Sx+y=S(x+y)}^{\;\text{ax}}\right)=2^4\cdot 3^{26}\cdot 13^{g(\forall x,y.\,Sx+y=S(x+y))}\cdot 19^{g(x,y)}$;
- $g\left(\overline{\forall x.\,0\cdot x=0}^{\;\text{ax}}\right)=2^4\cdot 3^{27}\cdot 13^{g(\forall x.\,0\cdot x=0)}\cdot 19^{g(x)}$;
- $g\left(\overline{\forall x,y.\,Sx\cdot y=x\cdot y+y)}^{\;\text{ax}}\right)=2^4\cdot 3^{28}\cdot 13^{g(\forall x,y.\,Sx\cdot y=x\cdot y+y))}\cdot 19^{g(x,y)}$;
- $g\left(\overline{A[0/x]\wedge(\forall x.\,A\supset A[Sx/x])\supset\forall x.\,A}^{\;\text{ax}}\right)=$
  $2^4\cdot 3^{29}\cdot 5^{g(A)}\cdot 13^{g(A[0/x]\wedge(\forall x.\,A\supset A[Sx/x])\supset\forall x.\,A)}\cdot 19^{g(x)}$;
- if $A\in\Gamma$ is a proof by assumption, $g(A)=2^4\cdot 3^{30}\cdot 5^{g(A)}\cdot 7^{g(\Gamma)}\cdot 13^{g(A)}$ with $\Gamma=\{\gamma_1,\dots,\gamma_n\}$ and $g(\Gamma)=g(\gamma_1,\dots,\gamma_n)$.

↪

↪ (Coding proofs)

It should be remarked that $g(e_1,\ldots,e_n)$, when e_i are not numbers should be read as $g(g(e_1),\ldots,g(e_n))$, i.e., the code of the sequence of codes of the elements.

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.

Numeral

Definition 28.2 (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 $\ulcorner e_1, \ldots, e_n \urcorner = S^{g(e_1,\ldots,e_n)}(0)$.

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.

Representation

Definition 28.3 (Representable)

Let $R \subseteq \mathbb{N}^k$ be any relation. Then, R is *representable* when there is a formula A with $\mathrm{FV}(A) = \{x_1, \ldots, x_k\}$ such that $(n_1, \ldots, n_k) \in R$ if and only if $\vdash A[S^{n_1}(0)/x_1, \ldots, S^{n_k}(0)/x_k]$ in Peano arithmetic. Then, A is said to represent R.

A function $f : \mathbb{N}^k \to \mathbb{N}$ is *representable* when there is a formula A with $\mathrm{FV}(A) = \{x_1, \ldots, x_k, y\}$ such that, for each $n_1, \ldots, n_k \in \mathbb{N}$, Peano arithmetic allows to prove $\vdash A[S^{n_1}(0)/x_1, \ldots, S^{n_k}(0)/x_k] = (y = S^{f(n_1, \ldots, n_k)}(0))$.

With this definition, we can restate Proposition 27.3 by saying that each recursive relation is representable, and we can easily derive that each recursive function is representable, too.

References

The references for this lesson are the same as the previous one.

A proof of Proposition 27.3 can be found in *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), ISBN 0-7204-28440. A more accessible introduction is given in Chapter 3 of *S. Barry Cooper*, Computability Theory, Chapman & Hall/CRC, (2003).

Mathematical Logic

Lecture 29

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- The fixed point lemma
- Gödel's First Incompleteness Theorem

Fixed point lemma

Lemma 29.1 (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)

Let $\Delta_{\mathscr{F}}$ be the map from formulae to formulae defined by $\Delta_{\mathscr{F}}(B) \equiv \exists x_1 . x_1 = \ulcorner B \urcorner \wedge B$. This function is total, computable and injective.

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, (primitive) recursive, and injective.

By hypothesis, there is a formula Δ with $\mathrm{FV}(\Delta) = \{x, y\}$ such that Δ represents the function $\Delta_{\mathbb{N}}$.

Let $F \equiv \exists y . \Delta[x_1/x] \wedge A$. Clearly, $\mathrm{FV}(F) = \{x_1\}$. Also, let $\delta_A = \Delta_{\mathscr{F}}(F)$, that is, $\delta_A \equiv \exists x_1 . x_1 = \ulcorner F \urcorner \wedge F$. Thus, $\mathrm{FV}(\delta_A) = \emptyset$. ↪

Fixed point lemma

↪ Proof. (ii)

Since $\exists y.\Delta[\ulcorner F \urcorner/x] \wedge A$ implies $\exists x_1, y.\Delta[x_1/x] \wedge A$ with $x_1 = \ulcorner F \urcorner$, we can prove that $\exists x_1.x_1 = \ulcorner F \urcorner \wedge \exists y.\Delta[x_1/x] \wedge A$, which is just δ_A, Hence, we have shown that $\vdash (\exists y.\Delta[\ulcorner F \urcorner/x] \wedge A) \supset \delta_A$.

Conversely, $\delta_A \equiv \exists x_1.x_1 = \ulcorner F \urcorner \wedge \exists y.\Delta[x_1/x] \wedge A$, so δ_A implies $\exists x_1, y.\Delta[x_1/x] \wedge A$ with $x_1 = \ulcorner F \urcorner$, thus we can prove that $\exists y.\Delta[\ulcorner F \urcorner/x] \wedge A$. Hence, we have shown that $\vdash \delta_A \supset (\exists y.\Delta[\ulcorner F \urcorner/x] \wedge A)$, thus δ_A and $\exists y.\Delta[\ulcorner F \urcorner/x] \wedge A$ are equivalent.

But Δ represents $\Delta_{\mathbb{N}}$, so Ξ allows to prove, for each $n \in \mathbb{N}$, $\vdash A[S^n(0)/x] = (y = S^{\Delta_{\mathbb{N}}(n)}(0))$. Specialising to $n = g(F)$, we obtain $\vdash \forall y.\Delta[\ulcorner F \urcorner/x] = (y = S^{\Delta_{\mathbb{N}}(g(F))}(0))$. ↪

Fixed point lemma

↪ Proof. (iii)

So the previous equivalence $\vdash \delta_A = (\exists y.\, \Delta[\ulcorner F \urcorner / x] \wedge A)$ allows to derive $\vdash \delta_A = (\exists y.\, y = S^{\Delta_{\mathbb{N}}(g(F))}(0) \wedge A)$.

Evidently, we can prove $\vdash A\left[S^{\Delta_{\mathbb{N}}(g(F))}(0)/y\right] = \left(\exists y.\, y = S^{\Delta_{\mathbb{N}}(g(F))}(0) \wedge A\right)$, thus we can immediately prove $\vdash \delta_A = A\left[S^{\Delta_{\mathbb{N}}(g(F))}(0)/y\right]$.

But $\ulcorner \delta_A \urcorner = S^{g(\delta_A)}(0) = S^{g(\Delta_{\mathscr{F}}(F))}(0) = S^{\Delta_{\mathbb{N}}(g(F)))}(0)$. Thus, the proof above can be rephrased as $\vdash \delta_A = A[\ulcorner \delta_A \urcorner / y]$. □

Provability predicate

Definition 29.2 (Provability predicate)

The formula $\mathscr{D}$ with $\mathrm{FV}(\mathscr{D}) = \{x, y\}$ is defined as
$\mathscr{D} \equiv \exists z.\, 13^y \cdot z = x \wedge \mathrm{isExpr}(x) \wedge \mathrm{isExpr}(y) \wedge \mathrm{isProof}(x) \wedge \mathrm{isFormula}(y)$.

The *provability predicate* T is the formula $\exists x.\mathscr{D}$, having $FV(T) = \{y\}$.

Clearly, $\mathscr{D}[\ulcorner \pi \urcorner / x, \ulcorner A \urcorner / y]$ holds exactly when A is the conclusion of the proof $\pi : \vdash A$. And, consequently, $T[\ulcorner A \urcorner / y]$ holds when A is provable.

The formulae isExpr(x), isExpr(y), isProof(x), and isFormula(y) in the definition of $\mathscr{D}$ have not been made explicit. While isProof(x) can be defined as $\exists z.\, 2^4 \cdot z = x$, and isFormula(y) can be defined as $(\exists z.\, 2^3 \cdot z = x) \wedge \neg$isProof(x), the definition of isExpr comes from the fact that the collection of codes forms a recursive set. It could be written down in an explicit way, but it is a cumbersome formula.

Theorem 29.3 (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 $T \nvdash G$ and $T \nvdash \neg G$.

Proof.

Consider the formula $\neg T[x/y]$: applying the fixed point lemma, there is G such that $\mathrm{FV}(G) = \emptyset$ and $\vdash G = \neg T[\ulcorner G \urcorner / y]$.

Assume there is $\pi \colon \vdash G$. Then $\vdash \neg T[\ulcorner G \urcorner / y]$. 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, $\vdash T[\ulcorner G \urcorner / y]$, making the theory non consistent. Hence $\nvdash G$.

Oppositely, suppose there is $\pi \colon \vdash \neg G$. Then $\vdash T[\ulcorner G \urcorner / y]$ 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 $x = \ulcorner G \urcorner$. Thus, again, we get a contradiction. Thus $\nvdash \neg G$. □

References

As this lesson continues the preceding ones, the previous references are still valid.

Mathematical Logic

Lecture 30

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- Gödel's Second Incompleteness Theorem
- Meaning and consequences

Properties of provability

Proposition 30.1

For any pair of formulae A and B in Peano arithmetic,

1. $\vdash T[\ulcorner A \urcorner / y]$ *if and only if* $\vdash A$;
2. $\vdash T[\ulcorner A \supset B \urcorner / y] \wedge T[\ulcorner A \urcorner / y] \supset T[\ulcorner B \urcorner / y]$;
3. $\vdash T[\ulcorner A \urcorner / y] = T[\ulcorner T[\ulcorner A \urcorner / y] \urcorner / y]$;
4. $\vdash (T[\ulcorner A \urcorner / y] \wedge T[\ulcorner B \urcorner / y]) = T[\ulcorner A \wedge B \urcorner / y]$;
5. *if* $\vdash A \supset B$ *then* $\vdash T[\ulcorner A \urcorner / y] \supset T[\ulcorner B \urcorner / y]$;
6. *if* $\vdash (T[\ulcorner A \urcorner / y] \wedge A) \supset B$, *then* $\vdash T[\ulcorner A \urcorner / y] \supset T[\ulcorner B \urcorner / y]$.

These properties, we are not going to prove, show that the provability predicate T allows (i) to prove A whenever there is proof the A is provable; (ii) it acts naturally with respect to implication and conjunction; (iii) proving provability is equivalent to prove that provability is provable.

Properties of provability

Proposition 30.2

In Peano arithmetic, if $\vdash A = \neg T[\ulcorner A \urcorner / y]$, *then* $\vdash T[\ulcorner A \urcorner / y] = T[\ulcorner \bot \urcorner / y]$.

Again, without proving it, the proposition says that every formula, which behaves like Gödel's G, is provable if and only if $\bot$ is provable, a fact that captures the content of Theorem 29.3. But, and this is important, the proposition proves that this fact holds **inside** the theory, which is no obvious.

Second incompleteness theorem

Theorem 30.3 (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 $\vdash C \supset \neg T[\ulcorner \bot \urcorner / y]$.

Proof.

Suppose there is C such that $\vdash C$ and $\vdash C \supset \neg T[\ulcorner \bot \urcorner / y]$. Then, $\vdash \neg T[\ulcorner \bot \urcorner / y]$, which means that $\bot$ is not provable, that is, Peano arithmetic cannot contain a contradiction, hence it is consistent.

From Theorem 29.3, there is a formula G such that $\vdash G = \neg T[\ulcorner G \urcorner / y]$, but $\nvdash G$. By Proposition 30.2, $\vdash T[\ulcorner G \urcorner / y] = T[\ulcorner \bot \urcorner / y]$, so $\nvdash \neg T[\ulcorner \bot \urcorner / y]$. Thus, we have a contradiction, showing that C cannot exist. □

Mathematical meaning

The incompleteness theorems close 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 other 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 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.

Popularity

The incompleteness theorems gained popularity also outside the mathematical world. A large part of the philosophy of logic, epistemology, and the philosophy of science analysed those theorems, obtaining sometimes deep and interesting results for mathematicians, too.

Popular science have often misused or misunderstood those theorems, drawing conclusion with no support or evidence. Unfortunately, the same happened in some philosophical studies, drawing conclusions about the "power of the rational mind" which are mere conjectures with respect to the evidence the theorems provide.

The truth is that the incompleteness theorems have a technical content which is unavoidable, and it is important to understand it properly in order to reason about the limit it imposes to the formal development of mathematics and of formal reasoning.

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 content. 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.

References

The technical proof of the second incompleteness theorem can be found, for example, in *John Bell* and *Moshé Machover*, A Course in Mathematical Logic, North-Holland, (1977), ISBN 0-7204-28440.

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), ISBN 0-444-863888-5.

As an example of a (very) popular book which deals with incompleteness, we signal *D. Hofstadter*, Gödel, Escher, Bach: an Eternal Golden Braid, Basic books, (1979), ISBN 0-465-02656-7. 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.

Mathematical Logic

Lecture 31

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- *Natural* incompleteness results
- Incompleteness in set theory
- Ordinal analysis

Natural incompleteness

The Gödel's incompleteness theorems show that there are sentences in Peano arithmetic which are true in the standard model, but they cannot be derived in the formal system.

A long-standing critique to those sentences is that they are "unnatural": they do not really speak about the properties of naturals as numbers, but about naturals as codes for properties about the theory, so they are "irrelevant" to the mathematical practise.

This position lasted for a very long time, and it is not difficult to find textbooks reporting it. But it is wrong.

Subsequent research has shown a number of sentences which cannot be proved in Peano arithmetic, but still, they are true in the standard model, and, more important, they have a genuine mathematical interest outside logic. In the following, we will illustrate some of them.

Natural incompleteness

Theorem 31.1 (Paris, Harrington)

For all $e, r, k \in \mathbb{N}$, there is $M \in \mathbb{N}$ such that, for every $f: \{F \subseteq \{0, \ldots, M\} : |F| = e\} \to \{0, \ldots, r\}$, there is $H \subseteq \{0, \ldots, M\}$ such that

- $|H| \geq \max\{k, \min H\}$, *and*
- *exists $v \leq r$ such that, for all $F \subseteq H$ with $|F| = n$, $f(x) = v$ for each $x \in F$.*

By using the Infinite Ramsey 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**. Nevertheless, it is possible to show, *within Peano arithmetic*, that the combinatorial principle in Theorem 31.1 implies the consistency of Peano arithmetic, thus it is impossible to prove in that theory, according to Gödel's second incompleteness theorem.

Natural incompleteness

Actually, a simplified version of Theorem 31.1 suffices:

Theorem 31.2

For all $n \in \mathbb{N}$, there is $M \in \mathbb{N}$ such that, for every function $f : \{F \subseteq \{0,\ldots,M\} : |F| = n\} \to \{0,1\}$, there is $H \subseteq \{0,\ldots,M\}$ for which, for all $F \subseteq H$ with $|F| = n$, $f(F) = \{0\}$, and $|H| > n(2^{n \min H} + 1)$.

This theorem and the previous one are *natural* in the sense that, changing the first condition in Theorem 31.1 to $|H| \geq k$, we get the Finite Ramsey Theorem, which is provable inside Peano arithmetic, and which is the starting point for a large branch of Combinatorics.

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 31.3

There is some $n \in \mathbb{N}$ such that, if $T_1, \dots, T_n$ is a finite sequence of trees, where T_k has $k+n$ vertices, then, for some $i < j$, there is an injective map $f\colon T_i \to T_j$ 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

Kruskal's Theorem plays an important role in the algebra of well quasi orders, a topic which has shown relevance in proving the termination of algorithms, so the above independence result has a direct, negative, application to Computer Science, for example.

In this sense, Kruskal's Theorem is "natural" and practically significant.

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.

In fact, the independence results in set theory and in Peano arithmetic are related. For example, Theorem 31.1 is a restriction to the finite case of the proof of independence about the existence of large cardinals.

A cardinal k is said to be *large*, simplifying a bit, when, for every $x \in k$, $\wp x \in k$, too. This fact is spelt as, k is large when, for every $f : \{\{k_1, k_1\} : k_1 \in k, k_2 \in k\} \to 2$, there is $\lambda \in k$ such that f restricted to λ is constant.

Ordinal analysis

There is a branch of proof theory devoted to study the "power" of deductive systems, showing which is the minimal ordinal to which transfinite induction can be relativised in order to prove a consistency statement.

This is a deep, delicate, difficult, and complex 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 statement can be shown to hold.

References

A dated, but still valid reference for Ramsey theory is *R. Graham, B. Rothschild, J.H. Spencer*, Ramsey Theory, 2nd edition, John Wiley and Sons, (1990), ISBN 0-4715-0046-1.

The original paper *J.B. 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 *M. Bóna*, A Walk Through Combinatorics, 2nd edition, World Scientific, (2006), ISBN 981-256-885-9.

References

The link between Kruskal's theorem and logic is analysed in depth in *J.H. 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), ISBN 0-444-863888-5.

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), ISBN 978-3-03719-022-7, written by a master of the field.

Mathematical Logic

Lecture 32

Dr Marco Benini

marco.benini@uninsubria.it

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

a.a. 2015/16

Syllabus

Limiting results:

- Incompleteness and computability

Conclusions

Incompleteness and computability

The incompleteness results have proved to be extremely useful in the study of computability. In fact, using the coding techniques developed to establish Gödel's theorem, a number of limiting results about what is computable have been derived.

Also, analogously to the notion of independence, it is possible to develop hierarchies of machines, computing modulo an oracle, that allow to classify the difficulty in solving problems, either by showing their distance to what is computable, e.g, the *arithmetic hierarchy*, or comparing them to efficient procedures to solve problems, e.g., the *polynomial hierarchy*.

As an example, we show that, fixed an enumeration $\{\phi_i\}$ of all the recursive functions,

Proposition 32.1

The set $H = \{i \in \mathbb{N} \colon \phi_i(i) \text{ is defined}\}$ *is not recursive.*

Proof.

Suppose it is. Then, its characteristic function χ_H has to be recursive and total. So

$$f(x) = \begin{cases} 0 & \text{when } \chi_H(x) = 0 \\ 1 + \phi_x(x) & \text{otherwise} \end{cases}$$

has to be recursive and total, too. Thus, there is $j \in \mathbb{N}$ such that $f = \phi_j$.

If $f(j) = 0$, then $\phi_j(j) = 0$, so $\chi_H(j) = 0$, which means $j \notin H$, that is, $\phi_j(j) = f(j)$ is not defined, despite it has the value 0. Hence, $f(j) \neq 0$.

So $f(j) = 1 + \phi_j(j) = 1 + f(j)$, thus $0 = 1$. So, H cannot be recursive. □

Incompleteness and computability

The fundamental theorem of complexity theory is an example of coding:

Theorem 32.2 (Cook, Levin)

The Boolean satisfiability problem is NP-complete.

Without developing the details, a *decision problem* is a pair $\langle I, P \rangle$ with $P \subseteq I$. It is solved by showing a recursive total function which calculates the function $f\colon I \to \{0,1\}$ such that $f(x) = 1$ if and only if $x \in P$. A decision problem lies in the class NP when the function f can be computed in polynomial time by a non-deterministic Turing machine with Σ-acceptance.

And a problem is *NP-complete* when it is in NP and every other problem in NP can be reduced to it by a deterministic, polynomial time transformation.

The Boolean satisfiability problem SAT takes as instances I the propositional formulae $\bigcap_{i=1}^{m} \bigvee_{j=1}^{n} l_{ij}$m where l_{ij} can be $\bot$, the variable x_j, or $\neg x_j$; the set of accepted instances P contains exactly those formulae for which there is an assignment σ of variables which makes the formula true.

The proof of Theorem 32.2 goes by showing that SAT is is NP, which is simple, as we just need to guess the string $b_1, \ldots, b_n$ of Boolean values interpreting $\sigma(x_1), \ldots, \sigma(x_n)$.

The big part of the proof codes a generic non-deterministic Turing machine as a formula in I, the set of instances. The polynomial time constraint ensures that the formula has a polynomial length, and since SAT can be solved in non-deterministic linear time, it allows to simulate any other problem in NP, that is, every problem in NP is reduced to SAT by encoding a non-deterministic Turing machine solving it as an instance of SAT, whose satisfying assignment is the code of the execution steps of the simulated machine.

Incompleteness and computability

Another famous result which uses coding in a substantial way is

Theorem 32.3 (Turing, 1936)

Fixed an enumeration $\{\phi_i\}$ *of Turing machines, there is* $U \in \mathbb{N}$ *such that* $\phi_U(e,x) = \phi_e(x)$ *for all* $e, x \in \mathbb{N}$.

Here, the idea is to abstract over a concrete coding function, and to use the enumeration instead. As soon as the enumeration is effective, it can be used as a coding.

But the influence of Gödel's proof is more evident in the following

Theorem 32.4 (Kleene)

Fixed an effective enumeration $\{\phi_i\}$ *of all the recursive functions, for any total recursive function* f*, there is* $k \in \mathbb{N}$ *such that* $\phi_{f(k)} = \phi_k$*.*

Proof.

Consider $\phi_{\phi_i(i)}$: for any $x \in \mathbb{N}$, calling u the index of the universal function, we can write $\phi_{\phi_i(i)}(x) = \phi_u(\phi_i(i), x) = \phi_u(\phi_u(i,i), x)$. Thus, $\phi_{\phi_i(i)}$ is a partial recursive, since it can be written as the composition of partial recursive functions.

So there is $e \in \mathbb{N}$ such that $\phi_{\phi_i(i)}(x) = \phi_e(x, i) = \phi_{S(e,i)}(x)$. The S function here is defined by $\phi_i(x, y) = \phi_{S(i,y)(x)}$, which corresponds to say that y is fixed. This is the S_1^1 Theorem of computability theory, we are not going to prove.

Now, $f(S(e,i)) = \phi_m(i)$ for some $m \in \mathbb{N}$ since e is fixed and f is recursive. Let $k = S(e,m)$: $\phi_k = \phi_{S(e,m)} = \phi_{\phi_m(m)} = \phi_{f(S(e,m))} = \phi_{f(k)}$. □

This difficult proof reminds the Fixed Point Lemma 29.1. In fact, it is just the same proof, rephrased on computable functions!

Although the parallel can be carried further, with many more examples, we will stop here.

References

A nice introduction to computability theory, which explains also the origin of the results can be found in *S.B. Cooper*, Computability Theory, Chapman & Hall/CRC Mathematics, (2004), ISBN 1-58488-237-9.

The original paper stating that SAT is NP-complete, is *S. Cook*, The complexity of theorem proving procedures, Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158, (1971). A classical text on complexity theory is *M.R. Garey, D.S. Johnson*, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, (1979), ISBN 0-7167-1045-5

Reading *A.M. Turing*, On Computable Numbers, with an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society 2(42), pp. 230–265, (1936) is highly recommended. But beware that there is one mistake, which has been corrected in 1937 by Turing himself.

The fixed point theorem can be found in *S.C. Kleene*, On Notation for Ordinal Numbers, Journal of Symbolic Logic 3, pp. 150–155, (1938).

The end