Introduction

The Goal of this Course

This is *not* a course on mathematical logic!

While mathematical logic is the theory of mathematical languages, we will study *one mathematical language*, namely the language of *predicate logic*, for "speaking mathematics" as opposed to "speaking *about* mathematics" as it is often done in classical courses on mathematical logic.

This *is* a course on mathematical logic!

Logic is the study of reasoning; and mathematical logic is the study of the type of reasoning used in mathematics. We will use the tools offered by predicate logic and train the proper use of the same by doing some concrete mathematical reasoning in concrete mathematical areas.

In this course, we try to study the principles of mathematical activity by actually doing mathematics on concrete examples from areas treated in the introductory courses of the maths curriculum, i.e. Linear Algebra, Analysis, and Algorithmic Methods. While we view "mathematical activity" as an interplay between "proving", "solving", and "computing", this course will focus mainly on the aspect of *proving*.

What is a mathematical proof? Within mathematics, there is no formal definition of what a proof should be. Informally speaking, a proof should be some "convincing argument", why a particular statement is true or false. Why do we need proofs? Mathematics proceeds by starting from some given concepts (e.g. real numbers), then inventing new concepts based on the given ones (e.g. sequences of real numbers, arithmetical operations on sequences, limits of sequences) and explore the properties of the new concepts (e.g. limit of the sum or the product of two sequences). Intuition (and experience) leeds the mathematician in building up new mathematical areas. Intuition (and experiments) is also helpful in figuring out what facts might be true and which are not. But, due to the complexity of mathematical concepts, intuition is often not enough in order to gain certainty abouth truth or falsity. Moreover, one's own intuition often fails to convince someone else in case this person's intuition differs from one's own.

We will study the language of predicate logic as a language for formulating mathematical statements. Based on this language, we can study rules that tell us how to safely proceed from a known status to some new valid status. A proof of some statement S, in this view, is then just a sequence of statements starting from known facts ending in S, where each transition on the way from the known facts to the final statement S is justified by some rule. The rules used in this process are not arbitrary, in contrast, they should reflect the condensed experience of mankind in what is considered to be true. For instance, if we know that "A is true" and that "B is true whenever A is true", then "B must necessarily be true".

Of course, not everything can be proven. The first mathematical laws cannot be proven, because there are no earlier laws, from which to derive them. There must be some starting point, we call those laws *axioms*. Axioms should describe facts about the objects under consideration that are *obviously true*. For instance, consider the natural numbers, then the fact that "no natural number has 0 as its successor" is commonly accepted as an axiom for natural numbers. There is no rule how to choose appropriate axioms, rather, there are two extremes to be balanced:

- The more axioms the richer the expressive power of the theory, i.e. the more facts can be shown to be true.
- The less axioms the higher the chance that the axioms do not contradict each other.

As we will learn, a theory is *worthless* as soon as it contains just one contradiction, because it allows to derive any fact, i.e. in a theory containing a contradiction, everything is true.

It has been shown that *first order predicate logic* with *set theory* has sufficient expressive power to formulate in this theory *all of mathematics*. In most of this course, we will stay in this setting.

The Use of *Theorema* in this Course

Theorema is a software system implemented by the *Theorema* group at RISC-Linz under the direction of B. Buchberger, see www.theorema.org. The system provides a frame for all aspects of symbolic computation and mathematical exploration management. It consists of a growing library (a "magma") of provers, solvers, and simplifiers for various areas of mathematics and various tools for supporting mathematical exploration management. The participants of this course may download the current version of the system from the course's web-page and obtain a 6 months license for using the system. *Theorema* requires a license of Mathematica (at least version 3.0).

Theorema contains computer-support for predicate logic, set theory, and tuple theory, i.e. it contains a concrete syntax for these elementary language constructs. We will study the language of predicate logic in its concrete appearance as in the *Theorema* system. Moreover, *Theorema* allows to organize mathematical knowledge by giving definitions, stating theorems, and collecting knowledge into structured theories. *Theorema* would even generate some of the proofs, which we will study in this course, completely automatically! The aim of this course is, however, *not* to train the students to operate *Theorema* such that it generates proofs automatically. After having attended this course, students should be capable of recognizing the structure of theorems and the structure of facts in the knowledge base and, by this, have a clear understanding of how to structure a proof. For obtaining this insight into structure, we might occasionally inspect *Theorema*-generated proofs.

Theorema Demo

Structure of the Course

Introduction
Syntax and Informal Semantics of Predicate Logic
Proof Rules for Predicate Logic
Case Studies
Solutions to Exercises

Literature

Literature Intimately Related to the Course

Bloch Ethan D. (2000). Proofs and Fundamentals: A First Course in Abstract Mathematics, Birkhäuser, ISBN 0-8176-4111-4.

Buchberger Bruno (2002). Lecture Notes for “Predicate Logic as a Working Language”, University of Linz, SS02.

Buchberger Bruno. Lecture Notes for “Thinking Speaking Writing”, developed over the years.

Buchberger Bruno. Lecture Notes for “Algorithmic Mathematics”, FHS Hagenberg, 1996.

Literature on Mathematical Logic

Andrews Peter B. (1986). An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof. Academic Press Inc., ISBN 0-12-058537-7.

Buchberger Bruno, Lichtenberger Franz (1981). Mathematik für Informatiker I, Springer Berlin Heidelberg New York, ISBN 3-540-11150-6.

Ebbinghaus H.D., Flum J., Thomas W. (1984). Mathematical Logic. Springer Berlin Heidelberg New York Tokyo, ISBN 3-540-90895-1.

Shoenfield Joseph R. (1967). Mathematical Logic. Addison Wesley Inc.