Preface

In the recent decades mathematical logic has become more and more important in computer science and, in general, in system engineering. In fact, by definition, it is the way of expressing our reasoning in terms of mathematical formalism, thus supplying it with the typical rigor and precision of mathematics. Not by chance, automatic information processing is now pervasive and we find it practically in any human activity and artefact, from embedded, safety-critical systems, to e-commerce, to social networks, etc. Such a pervasiveness and the consequent heterogeneity of the involved systems mandate much more generality in the formalisms supporting the engineering activity than traditional specialized models such as, e.g., those for electric circuits and mechanical engines: mathematical logic, paired with computer applications, provides such generality.

In fact, not only mathematical logic is one of the historical theoretical foundations of Computer Science, starting from the bases laid by Aristotle, to the work by Leibniz, Hilbert, Turing, Goedel and many other pioneers, but it now supports the most recent software engineering tools and system specification methods.

For this reason we decided to offer these lecture notes. They consist of two parts: a first part introduces the basics of the discipline without the coverage of a traditional text on mathematical logic, but with sufficient breadth and depth to allow to apply them in a non-trivial way; the latter part shows how mastering mathematical logic can help modelling and analyzing in a precise and rigorous way practically any type of "system". Again the presentation style is introductory and focusing on the essentials, but –hopefully– not superficial at all.

These lecture notes have been designed as a complement to the text "Theoretical Foundations of Computer Science", John Wiley & Sons, by Dino Mandrioli and Carlo Ghezzi; however, they could be exploited as well either as the teaching aid for a self-contained short course or as a complement to other traditional texts on automata, formal languages, and theory of computation.

We assume that the reader has knowledge at an introductory level of set theory, formal language theory, and programming.

Warning: for timing reasons we deliver these lecture notes in a fairly preliminary version. We plan a revision thereof shortly hereafter.