Preface

In June 2000, the third named author visited New York University and was invited by Jack Schwartz to read what he called the ("common shared") *scenario*: a wide, carefully assembled sequence of definitions, theorems, and proofs, leading from the bare rudiments of set theory to the beginning of mathematical analysis. Proofs began to be gappy after a few hundred pages, and then totally absent, but the flow of definitions and theorems went on, to culminate in the definition of complex line integral and finally in the celebrated Cauchy integral theorem of complex analysis.

With an implementation appearing all but imminent, Jack had cast a significant piece of mathematics in rigorous formal detail, honestly asking himself whether a computer program could conceivably process and validate every single step. The resulting large-scale proof scenario was meant—in Jack's own words—"to serve as an essential part of the feasibility study that must precede the development of any ambitious proof-checker". Eventually, it would also serve as a testing-bench for the concrete implementation of the proof-checker.

The conception of this book on computational logic began then. According to our initial plans, the book would have described the structure of a proof verifier rooted in set theory and would also have surveyed a twenty-year long stream of results on decidable fragments of set theory.

One year later, the second named author visited New York in his turn; at his request Jack advanced the implementation work, speedily bringing into existence the proof-checker Referee, also known as Ref, or as ÆtnaNova. This is still a prototype, but it is reliable and fast enough to give us the possibility of debugging our proof scenarios. Some, though not all, of the content of this book is thus related to concrete experience, which we are now pleased to share with our readers.

A very large proof scenario is available today as a LaTeX-generated PDF-file. But given its size (over a thousand pages), it seems appropriate to publish it on the web (and eventually as a CD) rather than to print it. As for the proof verifier, it is usable on the web, but it depends on a SETL2 implementation. Since there is hardly anyone maintaining the SETL system today, Jack undertook with us a re-implementation of the proof verifier in a currently more popular language. But this will take some time; it should not be permitted to delay the publication of this book.

This is a posthumous publication, as its principal author passed away on March 2, 2009. In spite of his long illness, until the end of his life, Jack showed unbelievable resources of energy in the preparation of this book, in implementing and debugging Ref, and in drafting and writing the scenario. With the inspirer of this work gone, the book may not have achieved the degree of perfection that had been Jack Schwartz's goal; nevertheless we believe that the material he left behind will attract many readers and that its publication will be an appropriate tribute to a distinguished scientist.

A Word on the Audience for Whom This Book Is Intended

Any technical book must, by emphasizing certain details and leaving others unspoken, make definite assumptions about the prior knowledge of the reader.

This book assumes that the reader has a good knowledge of standard programming techniques, particularly of string manipulation and parsing, and also a general familiarity with those parts of mathematics that are analyzed in detail in the main series of definitions and proof scenarios to which much of the book is devoted.

On the other hand, little knowledge of formal logic is assumed. For this reason we try to present what is needed from logic in a reasonably self-contained way, emphasizing concepts likely to be important in continuations of the work begun here, rather than technicalities. Foundational issues, for example consideration of the strength or necessity of axioms, or the precise relationship of our formalism to other weaker or stronger formalisms studied in the literature, are neglected.

Because we expect our readers to be programmers of some sophistication, syntactic details of the kind that often appear early in books on logic are underplayed, and we repeatedly assume that anything programmable with relative ease can be taken as routine, and that the properties of such programmable operations can be proved when necessary to some theoretical discussion. This reflects our feeling that understanding develops top-down, focusing on details only as these become necessary.

We believe that too much detail is more likely to impede than to promote understanding. Who reads, or would want to read, the Whitehead–Russell *Principia*, or could testify that its hundreds of formula-filled pages are without error? But since we ask this question, why do we include hundreds of formula-filled pages in this book, which would not exist without the pioneering work of Whitehead and Russell? The reason lies in the fact that our formal proof text is, to a large extent, computer-checked. Though relatively useless to the human reader unless their correctness can be verified mechanically, long lists of formulae become useful once such verification becomes possible.

Content of This Book

Chapter 1 gives rapid overviews of the authors' approach to automated proof verification and of the large-scale formalized proof scenario whose development has accompanied the writing of this book.

Chapter 2 prepares for an extensive account of our proof verifier ÆtnaNova, by surveying three traditional branches of logic: propositional calculus, first-order predicate calculus, set theory. Completeness proofs are provided for the first two of these deductive systems; the much-debated issue of the consistency of Zermelo–Fraenkel set theory and of some of its proposed extensions is highlighted.

Chapter 3 provides an extended survey of inference mechanisms. Some of these belong to the initial endowment of ÆtnaNova, others are candidates for inclusion in that endowment should our proof verifier be re-implemented or enhanced. In some cases efficiency considerations show that an inference mechanism cannot be applied at its fullest; notwithstanding we present it because of the mathematical insight it provides. Two classics of the automated deduction field, Robinson's resolution principle and the Knuth–Bendix equational method, are also surveyed in this chapter.

Chapter 4 describes our verifier and its underlying design in more detail. In Chapter 5 we expand a broad survey of main definitions and theorems, showing the salient steps of a formalized proof scenario leading toward the (as yet) unachieved goal of proving the Cauchy integral theorem.

In Chapter 6, for completeness sake and to enjoy the intellectual insight that these results provide, we derive several of the main classical results on undecidability and unsolvability; in particular, Chaitin's theorem and the two celebrated Gödel's incompleteness theorems.

To convey the character of a scenario verifiable by means of our ÆtnaNova system, we conclude with Chapter 7 showing formalized proofs of many facts about ordinals, of various properties of the transitive closure operation, of finite and transfinite induction principles, and of Zorn's lemma.

Acknowledgements

We are grateful to Martin Davis, to Alfredo Ferro, and to Alberto Policriti for encouraging, through decades of scientific interaction with the authors, the maturation of many ideas in this book; to Salvatore Paxia, who enabled us to keep the ÆtnaNova system alive; to Alexandru Ioan Tomescu, who contributed to the development of proof scenarios. Various results reported in this book stem from joint work with Gianluca Cincotti, Piero Ursino, and Calogero Zarba; Emanuele Giaquinta gave us precious advice concerning LaTeX.

Diana Robinson Schwartz gave us major support throughout the preparation of this book.

This research was partially funded by MIUR/PRIN project 2006/2007 "*Large-scale development of certified mathematical proofs*" No. 2006012773, and by INdAM/GNCS (Istituto Nazionale di Alta Matematica "F. Severi", Gruppo Nazionale per il Calcolo Scientifico).

Catania, Italy — Domenico Cantone
Trieste, Italy — Eugenio G. Omodeo

Springer

http://www.springer.com/978-0-85729-807-2

Computational Logic and Set Theory
Applying Formalized Logic to Analysis
Schwartz, J.T.; Cantone, D.; Omodeo, E.G.
2011, XVII, 416 p., Hardcover
ISBN: 978-0-85729-807-2