Preface

Many-valued logic are gaining increasing importance in various areas such as mathematics, physics, technology, social sciences, and especially in computer science.

In the first volume of this book we gave an extensive presentation of the theoretical foundations of many-valued logics. In this second volume we focus on automated reasoning and so selected examples of practical applications of such logics.

In the first chapter we introduce some additional algebraic notions that were not covered in the preliminaries of Volume 1. The second chapter contains sequent calculi for finitely valued Post logics where a sequent is defined to be any finite sequence of formulae (in particular, an empty one). The system considered here consists of decomposition rules (rules of elimination for logical connectives and quantifiers) and fundamental sequences as the axioms of the system. A sequent is deducible iff leaves on its proof tree are labeled by such axioms only. Of course, the appropriate completeness theorem holds.

The third chapter presents an overview of sequent calculi and there we define a sequent in n-valued logics to be an n-tuple of sequences of formulae. Once again, some of the sequences may be empty.

The most famous sequent calculi of the type are those introduced by G. Rousseau and Moto Takahashi. Both systems have certain similarities in presentations but differ in the technicalities. Takahashi's notion of a sequent is that of a matrix with n rows. However, in order to make the notation uniform we have allowed ourselves to change it to an ordered n-tuple of sequences of formulae. It did not affect Takahashi's idea. His calculus is closest to the classical Gentzen calculus for two-valued logics and, similarly, it is a deduction system. A proof tree is built starting with axiom labelling the leaves and the formula at the root is the goal. Moreover, apart from rules introducing logical connectives and quantifiers there are weakening and cut rules. Rousseau's system is also of the deduction type and both systems require a partial proof search algorithm. We conclude Chapter 3 with a brief presentation of Fitting's calculus for a limited class of logics in which sequent is just a pair of sets of formulae.

In Chapters 4 and 5 we discuss the resolution principle for finitely many-valued logics defined through matrices. We also mention some results by Z. Stachniak and P. O'Hearn. In Chapter 6 we describe a resolution principle for an n-valued Post logic. The principle, due to C.G. Morgan, is a generalization of the classical resolution.

Chapter 7 presents various reasoning systems for classical many-valued logics and their selected applications in some, often very different areas. Such a wide range of applications was made possible by the highly abstract and general presentations of the theory, a feature characteristic to these logics.

Finally, we have decided to devote the last two chapters of the book to processing of imprecise and approximate information since it has become a crucial issue in many areas. In Chapters 8 and 9 we discuss methods of formal presentation and reasoning with imprecise and approximate data, inspired by classical many-valued logics. In particular, methods based on fuzzy logics are considered in Chapter 8, where by fuzzy logics we mean many-valued logics with a generalized notion of an axiom and a generalized consequence relation, linguistic logics and reasoning systems with a generalized modus ponens rule. In Chapter 9 we take a rough set approach to the problem and introduce a logic constructed from this viewpoint. Most importantly, in both chapters we describe related applications, usually taken from the field of decision support systems, and give respective reasoning and control systems, or systems of pattern recognition and data perception.

We would like to add that the presentation of topics in the last three chapters, however brief, also shows how the classical many-valued logics of Łukasiewicz and Post have inspired the new systems in use today. Fuzzy set theory in Chapters 7 through 9 is based exclusively on results of L. Zadeh and authors who developed his theory further; on the other hand, in our presentation of the rough set theory – introduced originally by Z. Pawlak – we mention primarily his results and the research of those who extended the theory, for example, A. Skowron, R. Słowiński and J. Grzymała-Busse.

We would like to thank the above authors for their consent to present their results in this book. It must be emphasized that their work has provided the fundamentals for any current and future research in the area of many-valued logics.

We are also grateful to R. Hähnle, S. Gerberding and R. Zach: we have learned a lot from their work and often used their results in the book.

We also thank J. Cytowski, P. Rychlik, M. Srebrny and A. Szałas who have read the draft and suggested many improvements. Yet another person we have received many valuable comments from is B. Konikowska who has also helped us with editing. Moreover, we would like to thank G. Domańska-Żurek for typing the final version of this book.

Finally, our sincere thanks go to the editor Dr. Hans Wössner at Springer-Verlag and the copy-editor J. Andrew Ross. It was their effort and involvement that led to the publication of the book.

We hope that the two volumes of the monograph will indeed give the reader an overview of the state of the art in this rapidly developing area, and that the reader will find the monograph a convenient presentation of both theoretical foundations of many-valued logics and the vast range of their applications.

Warsaw, July 2003

Leonard Bolc
Piotr Borowik

Introduction:
The History of Automated Reasoning

The history of automated reasoning goes back to the 17th century, when Descartes proved that algebraic methods can provide solutions to problems of Greek geometry. This implied a certain mechanization of mathematics, of which Descartes was fully aware. In [Descartes 1637] he wrote:

> ...all the problems of classic geometry can be constructed with the aid of not more than the four pictures I have described. I believe this is the issue that the ancient have overlooked, for otherwise they would not put so much effort in writing so many books, where the presentation of theorems proves that they knew no method for proving everything...

As can be noticed, at that time already the idea of a method for proving all the theorems of mathematics was alive. Today we know that this was nothing but a dream. This chapter is, in fact, a description of the process of humanity's attempts to make automation of reasoning a reality. In this case, automation means the existence of an algorithm providing solution of problems without the intervention of human inventiveness and intelligence. The dream of a clear and direct mechanization of reasoning was first made explicit in Leibniz's works in the 17th century. Leibniz was charmed by the idea of Lullus to design a method for reducing all notions to some root notions which would constitute the basic alphabet. The set of all true propositions would then be composed of mechanical combinations of the root notions. In the course of his research on the language and methods of proof, Leibniz introduced two new notions: *calculus ratiocinator* and *lingua characteristica*. The first, *calculus ratiocinator*, denoted a deductive calculus which would allow logical conclusions to be drawn by transforming its expressions. *Lingua characteristica* was intended to be a universal language, that is, a language adequate for all the domains of human knowledge. These two notions determined the directions of further research in this area. In spite of having worked out many technical details, Leibniz was not able to arrive at far-reaching conclusions; even the great restrictions he imposed on the language were of little help. From our point of view, Leibniz developed a fragment of the theory of Boolean algebras with one binary operation only. The operation was sometimes interpreted as meet, other times as join. Leibniz's achievement consisted in treating logic as an abstract language with many possible interpretations, which is evidenced, for example, by his different interpretations of the binary operation symbol in the calculus. He also noticed the possibility of basing deduction on an axiom system. Unfortunately, most of Leibniz's work was not published until the 20th century and thus had very little influence on his contemporaries. In the 18th and the 19th century, several authors, including Segner, Lambert, Ploucquet, Holland, De Castillon, and Gergonne, undertook the investigation of similar problems, but none of them attained anything new with respect to the accomplishments of Leibniz.

Only in the middle of the 19th century was the *calculus ratiocinator*, in the sense of Leibniz, formed by Boole ([Boole 1847]). Boole noticed that operations on sets could be treated as an abstract system based on a set of axioms and, moreover, that the operations have an interpretation in logic, if logical values are substituted for the symbols. This idea was then used by Jevons, an economist and logician, to devise the first machine for the verification of Boole's identities in 1869 ([Kneale 1962]). The machine can be considered the first device ever produced for automated reasoning.

Ten years later, Frege developed the predicate calculus. At that time the present form of logical propositions was shaped: predicates with a finite number of arguments, linked by Boolean logical connectives and quantifiers. Frege's language was also the first example of a fully formalized artificial language. The form he suggested is still reflected in today's programming languages. Moreover, Frege introduced the rule of detachment or *modus ponens*, due to which he was able to describe the deduction process in the language of logic. Frege considered himself a follower of Leibniz. His logic was a development of the idea of *lingua characteristica*, a part of which was the *calculus ratiocinator*, i.e., a scheme for proving theorems, based on axioms and inference rules. However, Frege's work remained virtually unknown and was not understood, perhaps because it was quite ahead of its time ([van Heijenoort 1967]). The symbols he used were hardly intuitive and differed greatly from those used by contemporary mathematicians.

Some years later, Peano constructed his own formalized system of logic, to which we owe the present shape of the language of logic. But Peano took a step backwards with respect to Frege's approach. Unlike Frege, Peano did not use formal inference rules and consequently the entire deduction process was described in natural language.

Peano was aware of the great significance his work had for science. Nevertheless, he was severely criticized by Poincaré. Poincaré's attacks were highly emotional. Even if he clearly realized the importance of Peano's results, he could not agree with the project of transforming mathematical reasoning into a mechanical procedure. In his opinion, this would bring the destruction of the beauty of mathematics. Poincaré considered absurd the idea of ignoring creative features of mathematical thought. Similar opposition was aroused by the use of infinity in mathematics. Gauss considered inadmissible the introduction of the notion of infinity into the language of mathematics. Brouwer went even farther (in his critique of Cantor), initiating a new doctrine, called intuitionism. The doctrine rejected all non-constructive methods in mathematics. Brouwer postulated that proofs of existence be based on the explicit indication of the object whose existence is to be proved. Intuitionism was a very strong movement within mathematics at that time. Almost all mathematicians, even those who criticized it, fell under its influence. Even today, after the majority have accepted non-constructive proofs, mathematicians attempt to find a constructive one whenever possible.

A critique of intuitionism was carried out by Hilbert, who introduced the notion of metamathematics, a theory intended to prove the consistency of mathematics. Although an opponent of intuitionism, Hilbert included the requirement of constructive proofs in metamathematics. In fact, his intention was to present a constructive proof of the consistency of classical mathematics. Post went even farther by rejecting altogether the restrictions of intuitionism and constructing his own metamathematical program ([Post 1921]).

A key role in the history of automated reasoning must be attributed to Skolem, who investigated the notion of satisfiability in the predicate calculus. In 1928 Skolem formulated the notion known today as the Herbrand universe and introduced function symbols which made possible the elimination of existential quantifiers in logic formulae ([Skolem 1928, van Heijenoort 1967]). His proof method, based on the notion of a Herbrand universe, became a pattern for proof procedures. However, in his proofs Skolem used the axiom of choice, which made the method inadmissible for Hilbert.

In the same year a very significant publication appeared: [Hilbert, Ackermann 1928]. Its authors expressed some fundamental ideas lying at the foundations of automated reasoning. Two main problems were considered: the completeness of the axiom system presented in the book and the Entscheidungsproblem, i.e., the problem of finding an algorithm deciding the satisfiability of propositions in the predicate calculus. The first of these problems was solved by Gödel in 1930 ([Gödel 1930]). Gödel proved the completeness of the predicate calculus of Hilbert and Ackermann. In the next year he proved the existence of undecidable propositions in every formal system which contained the Peano arithmetic of natural numbers. There was yet another important result in Gödel's work: there is no consistent formal system powerful enough to prove its own consistency ([Gödel 1931]). Thus Hilbert's dream of proving the consistency of mathematics faded away.

A few years later Kleene and, independently, Mostowski introduced a hierarchy of notions defined in arithmetic. The hierarchy had infinitely many levels, the first of which contained decidable notions, the two next contained partially decidable notions and their complements, and all the subsequent levels notions of increasing complexity. In the following years many new negative results appeared, showing more and more new mathematical theories to be undecidable. At the same time intensive research was being done on decidable theories. In this context, the names of mathematicians like Ackermann, Langford, Presburger, Tarski, and Rabin must be quoted.

In 1930 another important work, [Herbrand 1930], appeared which determined a working scheme for inference systems valid to this day. The scheme uses the fact that the proof of a theorem can be reduced to the question of the existence of a model for the axiom system extended by the negation of the theorem. The reasoning then consists in reducing a formula to a certain normal form and eliminating the existential quantifiers by Skolemization. Next, a Herbrand universe is defined for the set of formulae obtained at the previous stage. Herbrand's theorem states that a set of formulae has no model if and only if there exists a non-satisfiable set obtained from a finite subset by substituting elements of the Herbrand universe for variables. This result implies an important simplification of deduction, for it reduces it to the consideration of a very concrete structure and of finite subsets of formulae without free variables.

The next significant step in the development of automated reasoning techniques was made in 1934, when Gentzen ([Gentzen 1934]) and, independently, Jaśkowski ([Jaśkowski 1934]) defined the sequential calculus, also known as natural deduction. In the hitherto applied reasoning methods the property to be proved being the final conclusion, in the natural deduction method it becomes the starting point. The process of proof consists in decomposing the formula into simpler ones until axioms are reached. Thus the burden of inference is transferred from the axioms to inference rules.

Further success in the area of logical foundations of mathematics was due to the works of Turing and Church. They proved, independently, the undecidability of the set of tautologies of first-order logic, that is to say, that there is no algorithm yielding "yes" when a formula is a tautology, and "no" in the opposite case. The two have attained the result in a different way. Turing used the notion of an abstract computing machine, today known as "Turing machine". He reduced the problem of deciding whether a given statement is a tautology to the halting problem, which is undecidable ([Turing 1937]). On the other hand, Church applied the lambda calculus invented by himself. In the proof, he also used the fact that the Hilbert–Ackermann axiom system for the predicate calculus was complete ([Church 1936, Davis 1965]). Thus, the set of tautologies of the classical first-order predicate logic is not decidable. Later Gödel improved the theorem by showing that this set is partially decidable ([Gödel 1931]), i.e., there exists an algorithm that yields "yes" for a formula which is a tautology, while in the case of a non-tautology it either yields "no" or enters a loop.

Another result of Gödel which is significant from the point of view of automated reasoning is the proof of the theorem stating that every partially decidable problem can be reduced to the problem of deciding whether a given statement is a tautology of classical logic. This entails important restrictions on new inference systems, for instance, in non-classical logics. It turns out that we cannot go beyond some fixed limits.

And this is actually the end of the early period in the history of automated reasoning. It is closely related to the history of new discoveries in the foundations of mathematics and logic. To make machines "think", the man had to analyse his own way of reasoning as well as its emanations, such as mathematics. Only after having overcome some difficulties in defining the process of deduction and its language precisely was it possible to initiate deduction simulation on computers. The rapid development of computer techniques and their accessibility greatly contributed to the process.

The first deductive programs began to appear. One of them was the so-called "logic machine" of Newell, Shaw and Simon. This was a program meant to prove theorems of the first-order predicate calculus; it was based on the Hilbert-Ackermann axiom system. The historical significance of the program consisted mainly in the fact that structures typical to most future applications were applied here for the first time. It has shown the direction that would be followed by subsequent proof programs.

In 1960 the first deductive systems based on theoretical results of Herbrand ([Herbrand 1930]) appeared. These systems were introduced in the papers [Gilmore 1960, Prawitz, Voghera 1960, Davis, Putnam 1960, Wang 1960a].

All these applications intended to imitate human thinking. This was trenchantly emphasized by Minsky ([Minsky 1961, Feigenbaum, Feldman 1963]):

> It seems to be clear that a program for the solution of real mathematical problems must be a combination of both Wang's mathematical refinement and the heuristic refinement of Newell, Shaw, and Simon.

In the 1960s the "geometric machine" of Gelernter appeared. It was able to prove geometric theorems. Among others, it proved the equality of angles at the base of an isosceles triangle. Simultaneously, Wang ([Wang 1960, 1960a, 1963]) announced a program capable of proving all the theorems of Whitehead and Russell's *Principia Mathematica.*

And so we have reached the year 1965, when Robinson made his resolution principle known ([Robinson 1965]). To this day the principle plays the main role in automated theorem proving (see Section 4.3). It brought forth an avalanche of new inference systems, theoretical investigation, generalizations of the original principle. Most modern proof programs use the resolution principle, often as the only rule. The resolution method requires the conjunctive form of a formula; this is one of the constraints to be obeyed when extending the method to non-classical logics. Many variants of the resolution method have appeared since then. Most changes are intended to restrict the search tree. Modern algorithms based on the resolution principle hardly remind the classic principle. There is even a version which can do without the conjunctive form ([Murray 1982]).

Another successful method is that of analytic tableaux. It originated in the semantic tableau method of Beth from the 1950s. The connection or link path method is relatively new. It was suggested by Bibel ([Bibel 1987]) and, unlike the resolution principle, it uses the disjunctive form of a formula. It also provides direct proofs, i.e., verifies the property of a formula of being true rather than the non-satisfiability of its negation, as is the case in resolution methods.

Analogously, many automated reasoning systems are based on the natural deduction method (see, e.g., [Constable et al. 1986, Gordon et al. 1986]). The great advantage of this technique relies on the fact that the proofs produced by the method are readable by humans. The method has found many serious applications in software design (cf. [Shankar et al. 1993]).

Clearly, the first automated reasoning systems were adapted to classical logic. Even if most systems are still addressed to this logic, soon systems for non-classical logic began to appear, in particular, for many-valued logics. One of the first works in this direction was [Morgan 1976].

The extension of the scope of applications for the resolution principle turned out to be relatively easy for some logics. After Morgan, new systems appeared, among them [Orłowska 1978, Schmitt 1986]. However, they referred to a rather narrow class of many-valued logics. In the case of many-valued logics, the lack of an adequate normal form for formulae required by the resolution method may cause serious problems. They have been overcome to a great extent in [Stachniak, O'Hearn 1990]. This seems to be the most general approach to automated reasoning based on resolution in many-valued logics. The system is applicable to all finitely-valued logics.

It turned out recently that the method of analytic tableaux can be very well adapted to many-valued logics. The first to produce such adaptations were Surma ([Surma 1984]) and Suchoń ([Suchoń 1974]). Later the idea was developed by Carnielli ([Carnielli 1991]) and Hähnle ([Hähnle 1990]).

And here is where history ends and the present day begins. We are witnessing a rapid development of automated reasoning and the emergence of many programs which find applications in various areas, like artificial intelligence, computer science, electronics, and medicine. The significant increase in computer power available today has made this possible.