Notre Dame Journal of Formal Logic
Volume 37, Number 4, Fall 1996

Book Review

Reiner Hähnle. *Automated Deduction in Multiple-valued Logics*. Clarendon Press, Oxford, 1993, ix + 172 pages.

Automated theorem proving is now a firmly autonomous domain of investigation. At its early stage it was focused mainly on the problem of mechanizing classical proof procedures so they would be entirely covered or, at least, supported by an actual or theoretical computer. To obtain desirable results, the most frequently used were the methods based on different versions of refutation for propositional and first-order logic.

An interest in various nonclassical logics as related to computer science stemming from successful applications has grown recently. This, in turn, has stimulated investigation of automated proof procedures. Hähnle's book is intended to be a monograph on automated deduction in multiple-valued logics. The work consists of nine chapters, including an introduction and conclusion. These are followed and completed by references and an index.

1 In the introduction Hähnle professes the faith. First, he rightly states that the book is the first monograph exclusively devoted to automated theorem proving in multiple-valued logics. There and later, he uses the term 'many-valued' as a replacement for 'multiple-valued' and 'multi-valued'. It is common practice to use these terms interchangeably in the literature with, perhaps, an inclination to note 'multiple-valued' in the environment of computer science. The author remarks that the existing books deal either with automated theorem proving or with many-valued logics but never the two topics together. Furthermore, for some other systems of nonclassical logics such as intuitionistic, modal, linear, conditional, nonmonotonic, and temporal logics, the references on theorem proving are quite numerous. The author explains this by the fact that these systems found applications in computer science. As for many-valued logics there are several reasons which cause unavailability of a good device or algorithm appropriate for computation and proving; first, the widespread opinion that many-valued logic is not very useful and that it lacks convincing applications; and secondly, according to Hähnle, a nonhomogeneity of the subject of many-valuedness,

Received January 29, 1996

its conceptual opaqueness, and great dispersion of the systems, which are mutually difficult to compare. The latter would also be responsible for the absence of general proof procedures and uniform automated proving theory.

The author briefly presents the program of the book. It comprises parts of a general, expository character. Such are, for example, Chapters 7 and 8, where one may find the overview of applications of many-valued logics and a comprehensive historical account of activities in many-valued theorem proving. The main body of the book, which emerged from the author's Ph.D. thesis, presents an original semantic tableau framework for many-valued theorem proving. Hähnle sets the following catalog of properties which such a framework should satisfy: (1) wide applicability, (2) flexibility, (3) easy adaptability, (4) performance and, finally, (5) closeness to the classical version. The approach in question uses sets-as-signs, instead of signs, to achieve more efficient representation of many-valued computational space. The author promises to evaluate his proposal vis-à-vis the catalog of properties just collected.

2 Chapter 2, *Preliminaries*, contains definitions and elementary properties of some concepts from the abstract algebra, syntax, and semantics of propositional and first-order logic. The material is selected and organized with a view to its use in further parts of the book. One finds here the notions of an abstract and free algebra and, on the other hand, several concepts from the theory of logical calculi such as propositional formula, propositional language, valuation, satisfiability, model, and tautology. *Propositional logic* is understood as a pair $L = \mathbf{(L, A)}$, where **L** is a propositional language and **A** a matrix for **L**. A handy repertoire of notions important for theorem proving treatment of first-order formulas completes the stock. So, the reader will find readable definitions of a parameter, the Skolem function, a substitution, a syntactical variant, a literal, and a clause, to mention only a few.

Deserving special attention are the notions and concepts related to finite many-valuedness. Definition 2.11 on p. 7 specifies that a k-ary connective in n-valued propositional logic is an n-valued generalization of some classical connective of the same arity whenever the corresponding function of the matrix on classical submatrix coincides with the $0-1$ function of the latter. Further, a natural "ordering" of connectives simulating the order of their respective functions is described, and thus one may speak about weak and strong connectives.

In Section 2.3 closing the chapter, one finds definitions of particular many-valued connectives and logics or, rather, classes of logics, which are used several times throughout the rest of the book. All matrices are defined using the set $\mathbf{n} = \{0, 1/n-1, \ldots, n-2/n-1, 1\}$ and $\{n-k/n-1, n-k+1/n-1, \ldots, 1\}$ as the set of designated logical values. The language of the basic first-order logic L_M^n is without 0-ary predicate symbols and has the connectives of disjunction $\vee$, conjunction $\wedge$, negation $\neg$, and n unary connectives $\mathrm{J}_0, \ldots, \mathrm{J}_{n-1}$ corresponding to values from $\mathbf{n}$. The J_i are the well-known Rosser-Turquette's connectives, which in the corresponding logical algebra are characteristic functions of the respective values. The author claims that the propositional part of L_M^n is functionally complete, that is, that every n-valued connective is definable by the ones already present. This desirable property, however, can hold in general only when we assume that all or, in some cases, at least some elements of the universe of the matrix treated as constant functions correspond to some propositional 0-ary connectives, (i.e., constants). From the text it is not clear whether this is assumed or not.

Next, there are definitions of weak connectives in *n*-valued logic, which are simply the *classical* connectives of Rosser and Turquette. They are tailored relatively to the division of the set **n** into two subsets—designated and undesignated elements—in such a way that if one thinks about their characteristic functions then the resulting connectives are classical. Note that disjunction and conjunction previously defined remain the same. The author remarks that the Rosser and Turquette logics bear a close resemblance to the three-valued Kleene logics. Consequently, he straightforwardly generalizes Kleene's construction onto the case of *n*-valued logic and subsequently considers *n-valued strong Kleene logic*, which is, obviously, not functionally complete. Definitions of two families of *n*-valued first-order logics in this notation, Łukasiewicz and Post, including two infinitely many-valued versions of the latter close the introduction.

Let us note that in the definition of the first-order valuation in Definition 2.19 on p. 9, the *n-valued* quantifiers are interpreted as *min* and *max*, respectively. This way of defining quantifiers, acceptable for finitely-valued logics with linearly ordered values, is unacceptable in some other cases, not to mention infinite logics with uncountable sets of logical values.

Chapters 3–6 bring the most essential and relevant material. They concern the tableaux in classical and, mainly, many-valued logics. Chapter 3 starts with an exhaustive introduction to that part of the book. The reader finds here a sketch of a relatively short history of the subject. Let us mention only that the formal proof systems called semantic (or analytic) tableaux can be traced back to the early 1950s. They have two founding fathers: Beth and Hintikka. Beginners may have some problems with crediting the early constructions, since the author's reference to Beth is indirect—the original 1955 famous *Semantic Entailment an Formal Derivability* is cited through its 1986 appearance in a collection of logic texts published in Germany. Hähnle distinguishes the 1968 Smullyan version of tableaux as particularly elegant and underlines that most tableau systems used today are based on this formulation. He does not, however, give any analysis of the similarities and differences of the germ solutions by Beth and Hintikka.

3 Chapter 3 is entitled *The logical basis: signed analytic tableaux*. Its four sections form an introduction to semantic tableaux for classical logic, a sketch of two tableau methods for finitely-valued logics, and a discussion of problems concerning multiple-valued extension of tableau systems.

In Section 3.1, *Signed tableaux for classical logic*, we find an account of semantic tableaux for classical first-order logic. As is well known the tableau systems for the logic in question come in two versions: signed and unsigned. In the first version, a two-element set of prefixes, usually called signs, { F, T } with F corresponding to 0 and T corresponding to 1, is used. Furthermore, the two approaches are equivalent. The author prefers the signed approach since it is naturally adaptable to many-valued cases. The obvious step leading from the two-valued to the multiple-valued signed tableau system consists of introducing a more-than- two-element set of signs and thus generalizes the notion of a signed formula. Following Smullyan, the author divides the set of signed formulas into four classes: α for propositional formulas of conjunctive type, β for propositional formulas of disjunctive type, γ for quantified universal formulas, and δ for quantified existential formulas. Next there come all necessary definitions concerning tableaux, branches, closure, and so on.

Recall that the tableau method of constructing logics is a syntactic counterpart of the method of verifying by contradiction whether a given formula is a tautology or not. The property that φ is a first-order tautology is equivalent, in terms of Smullyan's signed approach, to the fact that $\{F\varphi\}$ is a closed tableau, that is, each of its branches contains a pair of complementary formulas: $F\psi$ and $T\psi$. This adequacy (soundness + completeness) theorem mentioned on p. 9 is completed by Remark 3.14 stating that the strong soundness and completeness can be easily obtained from the former by observing that the deduction theorem

$$\{\varphi_1, \ldots, \varphi_n\} \models \varphi \text{ iff } \models \varphi_1 \wedge \cdots \wedge \varphi_n \supset \varphi$$

holds true. The author emphasizes that the above theorem does not hold in most many-valued logics and, moreover, that the consequence relation is not necessarily characterizable by finite matrices.

Contrary to the classical case the notion of a 'sign' or 'prefix' is central to the approach to many-valued theorem proving presented in the book. Hähnle defines an *n*-valued propositional logic as a triple $L = (\mathbf{L}, \mathbf{A}, \mathbf{S})$ where **S** is a finite set of signs with $\mathbf{L} \cap \mathbf{S} = \varnothing$. Furthermore, he remarks that the selection of the set of signs will result in different proof systems and force one to express basic queries, such as completeness, differently.

Building a multiple-valued tableau starts with the choice of a set of signs which is an alphabetic variant of the truth value set. The rules of tableau construction reflect matrices. The external link structure remains classical. Finally, the overall procedure of constructing tableaux to verify tautologousness in every particular case depends upon the designated set of values D or, more directly, on its complement to the set of all logical values. If there is only one undesignated value, then one constructs one tableau starting with the signed formula, with the sign corresponding to the value in question. In general, as many tableaux must be opened as there are undesignated values.

Section 3.3, *Multiple-valued extension of tableau systems*, brings a concise presentation of Surma's method of extending tableaux to handle any finitely-valued first-order logic. The method was first presented by Surma in 1974 at the International Symposium on Multiple-Valued Logics. Surma's somewhat sketchy presentation was developed, completed, and extended by Carnielli in 1987. The subsequent steps of Hähnle's presentation of the Surma-Carnielli method of refutation are illustrated with examples anchored in the three-valued logic L^3_M defined in Section 2.3. The tableau rules mirror entries of the truth-tables of the connectives. Thus, for the disjunction $\vee$ characterized by the max function $i \vee j = max\{i, j\}$, the disjunctive formula signed with $^1/_2$, the rule emerges simply from the entry on the table described by the formula

$$^1/_2(\varphi \vee \psi) \text{ iff } (^1/_2\varphi \text{ and } 0\psi) \text{ or } (^1/_2\varphi \text{ and } ^1/_2\psi) \text{ or } (0\varphi \text{ and } ^1/_2\psi)$$

and it has the following form.

$$\begin{array}{c|c|c} \multicolumn{3}{c}{{}^{1}\!/_{2}(\varphi \vee \psi)} \\ \hline {}^{1}\!/_{2}\varphi & {}^{1}\!/_{2}\varphi & 0\varphi \\ 0\psi & {}^{1}\!/_{2}\psi & {}^{1}\!/_{2}\psi \end{array}$$

As usual the vertical line signifies branching. It is obvious that in general the increase of logical value increases both the number of rules and the size of branching.

The author also discusses Carnielli's use of *distribution quantifiers* in a many-valued tableau environment. The idea, originated by Mostowski in 1957, is roughly that the quantifiers, including the standard two, are defined by the use of mappings from the powerset of the set of logical values **n** into **n**. In this part of the book, Hähnle recalls some defects of Carnielli's early formalization, namely, the presence of incomplete quantifier rules, and he criticizes Carnielli's refinement of introducing additional rules with an empty premise as ineffective. The chapter closes with a discussion of the advantages and disadvantages of Surma and Carnielli's method. On the side of merits there is, first of all, the ability to give a tableau proof system for many-valued first-order logic including distribution quantifiers. Among the obstacles which make the actual use of the method in a theorem prover highly problematic, the most important and typical are: the redundancy of the representation of the many-valued space, the complexity of quantifier rules, and the excessivity of the branching factor.

4 Chapter 4, *A new technique: Truth value sets as signs*, is the first part of the author's own setting. It contains the detailed description of an original and new approach to automated deduction in multiple-valued proving. An ingenious solution, which has to decrease redundancy of tableau systems considered in the preceding chapter, is the use of sets of signs as prefixes instead of signs.

In Section 4.1, *Sets as signs*, we find the detailed presentation of the new framework. The algebra of signs for a given propositional logic $L = (\,\mathbf{L}, \mathbf{A}, \mathbf{S}\,)$ is defined as an algebra $\mathbf{A}_{\mathrm{S}} = (\mathbf{S}, \mathrm{f}'_1, \ldots, \mathrm{f}'_{\mathrm{r}})$ similar to $\mathbf{A} = (N, \mathrm{f}_1, \ldots, \mathrm{f}_{\mathrm{r}})$ with N finite and the operations defined as mappings from finite sequences of elements of **S** into sets of signs:

$$\mathrm{f}'_1(\mathrm{S}_1, \ldots, \mathrm{S}_{\mathrm{m}}) = \bigcup\{\mathrm{f}_{\mathrm{i}}(\mathrm{j}_1, \ldots, \mathrm{j}_{\mathrm{m}}) \,|\, \mathrm{j}_{\mathrm{k}} \in \mathrm{S}_{\mathrm{k}}, 1 \le \mathrm{k} \le \mathrm{m}\}.$$

Any algebra $\mathbf{A}_{\mathrm{S}}$ defines a semantics of L in terms of truth values sets corresponding to the members of **S**. Thus, for a formula $\varphi = \mathrm{F}(\varphi_1, \ldots, \varphi_m)$ two related interpretations f and f$'$ of F in **A** and $\mathbf{A}_{\mathrm{S}}$ are associated. The definition of an (L-) *tableau rule*, Definition 4.4 on p. 34, specifies the conditions ensuring all expected properties, that is, soundness, completeness, and some minimizing requirements, expressed in terms of linear subtrees called *extensions*. A collection of extensions is a *conclusion* of a tableau rule when it satisfies four conditions which relate possible functions of the matrix with extensions and homomorphisms from the language **L** into the algebra of signs $\mathbf{A}_{\mathrm{S}}$. The properties, which the class of homomorphisms associated to a given logic must satisfy, imply a kind of minimality of a number of extensions as well as exhaustiveness of the covering of the truth tables of the connectives. A minimal set of homomorphisms associated to a connective immediately leads to a tableau rule.

For the disjunction $\{1/2\}(\varphi \vee \psi)$ in the three-valued logic already considered, this set of homomorphisms has two elements h_1 and h_2:

$$h_1(\varphi) = \{1/2\}, h_1(\psi) = \{0, 1/2\}$$

and

$$h_2(\varphi) = \{0, 1/2\}, h_2(\psi) = \{1/2\}.$$

This, in turn, means that we get the following rule.

$$\frac{\{1/2\}(\varphi \vee \psi)}{\begin{array}{c|c} \{0, 1/2\}\varphi & \{1/2\}\varphi \\ \{1/2\}\psi & \{0, 1/2\}\psi \end{array}} \tag{1}$$

The new paradigm requires some further changes in the conceptual environment. To provide them all one should collect and consider all possible queries or, at least, reduce them to a small set. The author is aware of that, and he gives a definition of a contradiction set of signed formulas: A signed formula for which no rule is defined is self-contradictory.

The section closes with two examples. The first, Example 4.8, brings a full tableau system for the propositional part of the three-valued logic L^3_M using the following set of signs:

$$\{\{0\}, \{1/2\}, \{1\}, \{0, 1/2\}, \{1/2, 1\}\}.$$

The set of rules, plainly presented on pages 38 and 39, consists of schemes such as (1) for every set in the family just specified and for every connective of L^3_M. It might be interesting to mention that for Rosser-Turquette connectives no rule with the prefix $\{1/2\}$ exists, which is a simple consequence of the fact that they range over the set $\{0, 1\}$. The last example, Example 4.9, contains the L^3_M tableau proof of validity of the formula $\neg \mathrm{p} \supset (\sim \mathrm{p} \wedge \neg \mathrm{p})$. The functions corresponding to connectives are: $\neg i = 1 - i$; $\sim i = 0$ if $i = 1$ and $\sim i = 1$ otherwise; $i \wedge j = \min\{i, j\}$; $i \vee j = \max\{i, j\}$; and finally, $i \supset j = j$ for $i = 1$ and $i \supset j = 1$ otherwise. Now, the proof tree in the system designed by the author appears as follows:

(1) $[-]\{0, 1/2\}(\neg \mathrm{p} \supset (\sim \mathrm{p} \wedge \neg \mathrm{p}))$
|
(2) $[1]\{1\}\neg \mathrm{p}$
|
(3) $[1]\{0, 1/2\}(\sim \mathrm{p} \wedge \neg \mathrm{p})$
|
(4) $[2]\{0\}\mathrm{p}$

(5)	$[3]\{0, 1/2\} \sim \mathrm{p}$	(7)	$[3]\{0, 1/2\}\neg \mathrm{p}$
	|		|
(6)	$[5]\{1\}\mathrm{p}$	(8)	$[7]\{1/2, 1\}\mathrm{p}$
	closed with (4,6)		closed with (4,8)

The next two sections, 4.2 *Soundness* and 4.3 *Completeness*, provide a proof of adequacy of the formalism. Consider a multiple-valued logic with N being the set of logical values and D the set of distinguished values, let N and D denote the sets of signs corresponding to N and D, respectively. Then the fact that there is a closed proof tree over $\mathsf{N}-\mathsf{D}\varphi$ is abbreviated with the string $\vdash_S \varphi$ and the property that φ is an L-tautology with $\models_L \varphi$. Soundness of a tableau formalism, that is, the implication that $\vdash_S \varphi$ implies $\models_L \varphi$, holds true whenever the set S of signs is *complete with respect to L*, which means that it contains all signs corresponding to all connectives of L (see p. 41 for details). The last property yields that an appropriate rule(s) for signed formulas in which F is the main connective is (are) defined and is ultimately the first part of adequacy.

The completeness proof for the system may be obtained by closely following the lines of standard tableau completeness proofs. In Section 4.3 the author makes appropriate modifications to the definitions of a Hintikka Set and of the Analytic Consistency Property adapting the whole apparatus to the many-valued case. Then, after proving Hintikka's Lemma and the Model Existence Theorem, a standard proof of the implication from $\models_L \varphi$ to $\vdash_S \varphi$ follows. Hähnle claims, providing no justification, that it is easy to extend and to pass through the whole procedure for first-order formulas.

Section 4.4, *Size of proof trees*, brings an analysis of the size of proof trees depending on the form of formulas. The only result here, *Proposition 4.22*, says that if for a logic L the set $\mathbf{S}_L$ of signs contains a sufficient number of signs, then no rule constructed according to the accepted standard has more than n extensions with at most two formulas in each. Furthermore, the author claims that his approach achieves a substantial improvement over several common many-valued logics: an appropriate analysis permits him to delineate a class of such "well-behaving" systems and leads to proofs which are not longer than in classical cases. Hähnle, however, is aware of limitations of the method. The proofs for such logics as Łukasiewicz logics become intractable, even for small n.

Section 4.5, *Function minimization*, is devoted to the problem of finding SOP, that is, sum-of-products, minimal representations of many-valued logical functions. The method here is to adapt the well known two-valued device of Karnaugh to describe the many-valued connectives. The main goal and the result of the section is to give an algorithm which permits one to find tableau rules for a signed formula whose main connective is binary.

5 Chapter 5 is entitled *Uniform notation regained: regular logics*. It brings an exhaustive discussion of the two problems related to the sets-as-signs formalization: the classification problem for many-valued tableau rules and the question of introducing quantifier rules.

In Section 5.1, *Primary multiple-valued connectives*, one finds an adaptation of the well-known classification by Smullyan. First, eight primary connectives in a given n-valued propositional logic are listed. For uniform description of these connectives the notion of the so-called *conjugate* truth value of a given $i \in N$ is defined, which equals $1-i$ and is the value associated to negation of a formula evaluated as i. The set $D^* = \{1-i : i \in D\}$, that is, conjugate of the set D of designated values, plays an important role in the characterization which follows. Namely, α and β rules for the nine connectives are given using four sets-as-signs: D, its complement D', D^*, and $\mathsf{D}^{*\prime}$. Depending on how the sets D' and D^* are related via inclusion, three kinds of logics obtain. If, for example, $\mathsf{D}' = \mathsf{D}^*$ and, consequently, $\mathsf{D}^{*\prime} = \mathsf{D}$, the resulting logic (with primary connectives) amounts to the classical logic, and its tableau system is standard. In the remaining cases, $\mathsf{D}' \subset \mathsf{D}^*$ and $\mathsf{D}' \supset \mathsf{D}^*$, one gets two dual classes of logics for which the tableau systems are unique and do not depend on the choice of D, D', D^*, or $\mathsf{D}^{*\prime}$. The author finds this disappointing and finds it useful to consider other possibilities for combining signs and rules so as to preserve the validity of the α and β rule schemata.

The following Section 5.2, *Regular logics*, is a partial reply to the request. The author observes that a certain regularity can be found in the truth tables of several multiple-valued functions. This applies in particular to functions corresponding to primary connectives. A thorough analysis in this direction ends with a definition of *regular logic* which, roughly speaking, is any n-valued logic containing only "regular" connectives and the set of signs which consists of sets of the form $\{0, \ldots, i - 1/n - 1\}$ and $\{i + 1/n - 1, \ldots, 1\}$ denoted by $\boxed{< i}$ and $\boxed{> i}$ in the text (see Definition 5.9 on p. 63). The device is so created that the resulting sound and complete tableau system is given by the uniform notation style α and β component rules which are natural generalizations of their classical counterparts and fall under the overall schemata established in the previous section.

The main result in Section 5.3, *On the scope of regular logics*, Theorem 5.23, says that for any n there is a functionally complete regular logic L. The proof of the theorem is based on the functional completeness of Post logics and the fact that the Post negation is definable by the use of regular operators. Next to this, one finds an estimation of the number of regular operators in a given n-valued logic.

In Section 5.4, *First-order multiple-valued logics*, the author recovers the question of introducing quantifiers and uniform tableau rules for these multiple-valued operators. One of the aims of the approach is to get more compact rules than those of Carnielli, discussed in Chapter 3. The author starts with the example of adapting the sets-as-signs to two quantifiers in three-valued logic already appearing in Example 3.19 and showing the incompleteness of Carnielli's approach. Now, the two quantifiers Qx and Rx are described using sets-as-signs (more precisely, only one case of the tableau for $1Qx$ and $1Rx$ and a simple exemplary tableau proof is presented of inconsistency of the set $\Phi = 1(Qx)p(x), 1(Rx)p(x)$ in Example 5.24 on pages 71 and 72). Contrary to the previous approach now, using new rules, the inconsistency of Φ can be proved. The rules are

$$\frac{\{1\}\ (Qx)\varphi(x)}{\begin{array}{c}\{0\}\varphi(c_1)\\ \{{}^1\!/\!_2\}\varphi(c_2)\\ \{0, {}^1\!/\!_2\}\varphi(t_2)\end{array}} \qquad \frac{\{1\}\ (Rx)\varphi(x)}{\begin{array}{c}\{0\}\varphi(c_3)\\ \{1\}\varphi(c_4)\\ \{0, 1\}\varphi(t_2)\end{array}}$$

where c_i are new and t_i are arbitrary parameters. The tableau proof of inconsistency of the set Φ runs in a straightforward way, the only difference in comparison to the proofs in propositional logics is that now the use of Skolem constants is indispensable. In this case also, an appropriate substitution for t_2 in order to close the tableau should be made.

Although this kind of characterization of quantifiers is handy and the proof of completeness of first-order tableau systems is easy to obtain (cf. p. 72), the rules are not standard, that is, they are neither of γ- nor of δ- type. For the purpose of getting a required description and to extend the uniform notation of classical quantifiers, the author concentrates on many-valued generalizations of the quantifiers $\forall$ and $\exists$. The author limits his attention to the case of regular logics, that is, when only signs $\boxed{> i}$ and $\boxed{< i}$ are allowed. Later then, the cases of the signs $\boxed{\geq i}$ and $\boxed{\leq i}$ and the singleton sign $\{\mathrm{i}\} = \boxed{\geq}\ i\ \cap\ \boxed{\leq i}$ are also used. Theorem 5.26 on soundness and completeness of natural many-valued counterparts of the classical uniform tableau for first-order regular logics is a straightforward generalization of the classical one. For its proof the author refers to Fitting and gives only "a nontrivial modification" of some necessary results in Lemma 5.27 describing conditions of satisfiability of signed by $\boxed{> i}$ and $\boxed{< i}$ formulas with quantifiers.

In Section 5.5, *Extensions*, we find a discussion of the scope of the method provided in the book. First, the author conjectures that every logic with a uniform notation style tableau system is pseudo-regular. The property is connected with a possibility of reordering of truth values and the use of special so-called filter connectives: a logic is pseudo-regular if it can be made regular with the help of reordering truth values and use of filter connectives. Next, he claims that there is room for further extensions received when the totally ordered N is replaced with partially ordered sets of truth values and, for example, semi-latices. Hähnle remarks that all such structures lead to nonlinear many-valued logics considered by Gabbay in "LDS-Labelled Deductive Systems." On the other hand, he claims that infinitely many-valued regular logics can be straightly handled using the rules from Section 5.2. Finally, the author is aware of the difficulties arising in some particular cases of many-valued logics, with Łukasiewicz logics foremost.

6 Chapter 6, *Beyond tableaux*, is concerned mainly with applicability of the concepts developed by the author in the context of inference procedures which are potentially more efficient than pure tableaux are. The concepts such as sets-as-signs, regular signs, and connectives are adopted in many-valued variants of certain refinements of analytic tableaux and some other classical procedures and other inference techniques.

In Section 6.1, *Lemma generation—asymmetric rules—analytic cut*, one finds a presentation of the problem of the technique called *Lemma Generation* in application to β-rules in classical sentential logic. In general, there are several equivalent tableau rules; for example, for $\mathrm{F}(\varphi \wedge \psi)$ they are as follows.

$$\begin{array}{c|c} \multicolumn{2}{c}{\mathrm{F}(\varphi\wedge\psi)} \\ \hline \mathrm{F}\varphi & \mathrm{F}\psi \end{array} \qquad \begin{array}{c|c} \multicolumn{2}{c}{\mathrm{F}(\varphi\wedge\psi)} \\ \hline \mathrm{F}\varphi & \mathrm{F}\psi \\ & \mathrm{T}\varphi \end{array} \qquad \begin{array}{c|c} \multicolumn{2}{c}{\mathrm{F}(\varphi\wedge\psi)} \\ \hline \mathrm{F}\varphi & \mathrm{F}\psi \\ \mathrm{T}\psi & \end{array}$$

Choosing any of the right side rules in a current application lies in the heart of the technique. Which one of the rules must be used depends on the situation on the branch under consideration. Since these asymmetric rules provide, in a sense, more information then the left-side basic rule, getting possible closure of the branch is simpler. In each such case the second formula in the column may be treated as a lemma, which is proved by the closed branch in which it appears.

The author discusses the complete formulas DA_k of order k. These are conjunctions of all possible disjunctions of k propositional variables and their negations. For $k = 2$ the formula in question is of the form

$$(p_1 \vee p_2) \wedge (p_1 \vee \neg p_2) \wedge (\neg p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2).$$

It follows that all formulas DA_k are unsatisfiable, although as it was shown by D'Agostino in 1992 the shortest closed tableau is relatively long. Just the use of lemma generation permits us to show that there are tableaux of polynomial size with respect to the size of the input. The example supports the thesis that usual, unmodified tableaux are less effective than truth table checking and gave D'Agostino the incentive to design a system KE, which is distinctive in that only one of its rules has more than one extension, while all its other rules are unary. The rule in question is the *principle of bivalence* (PB). A version of KE with the *restricted principle of bivalence* (RPB) is equivalent to tableaux with lemma generation.

In 6.1.3, *Lemma generation in multiple-valued logics*, Hähnle shows how lemma generation can be extended to the many-valued case using the sets-as-signs framework and outlines a many-valued KE system (MKE). The principle of multivalence in MKE has, for any covering $\{S_1, \ldots, S_m\} \subseteq \mathbf{S}_L$, $S_1 \cup \cdots \cup S_m$ and a formula φ, the following form

$$\frac{}{S_1\varphi \mid \quad \ldots \quad \mid S_m\varphi}$$

and is a convenient tool for proof branching. The use of (PM) is illustrated on p. 89 on a version of the tableau proof of the signed formula $\{0, 1/2\}(\neg p \supset (\sim p \wedge \neg p))$: after Step (4)—see above—one uses (PM) for the covering $\{\{0, 1/2\}, \{1\}\}$ and the formula $\sim p$, and branches the proof. It is worthwhile to note that the proof thus received is longer than the original one. On the contrary, much shorter and more elegant is a linear MKE proof of the same formula given on p. 90. It runs as follows.

(1)	$[-]\ \{0, 1/2\}\ (\neg p \supset (\sim p \wedge \neg p))$
(2)	$[1]\ \{1\}\neg p$
(3)	$[1]\ \{0, 1/2\}(\sim p \wedge \neg p)$
(4)	$[2]\ \{0\}p$
(5)	$[2, 3]\ \{0, 1/2\}\neg p$
(6)	$[5]\ \{1\}p$

Section 6.2, *Tableaux as integer programming problems*, is devoted to so-called mixed integer programming. The author recalls a definition of a general MIP problem (mixed integer programming) and its restricted versions, b-MIP problems (bounded MIP problems). The former consists of minimalizing a linear function with respect to a set of constraints given by linear inequalities in which rational and integer variables occur. In the latter, all solutions must be in the rational interval [0,1]. If there are no rational variables present we have a bounded integer programming (bIP) problem.

In 6.2.2, *Tableau proofs with constraints*, the author gives many-valued constraint rules for the set of signs S being a sum of m rational intervals in N—each signed formula can be represented by at most $2m$ regular signs, such that any k-ary connective F is b-MIP-representable. The rules are given for Łukasiewicz implication in terms of $\boxed{\leq i}$ and $\boxed{\geq i}$: the constraints are additional conditions appearing in conclusions of the rules, and they either mark values of parameters or are inequalities (possibly equalities) in which parameters are involved. Practically, constraints are algebraic formulas defining connectives. The author defined the concept of a closed tableau in this paradigm, appropriately choosing the bIP problem, in such a way that a formula φ is a tautology iff there is a completed tableau for $\boxed{\leq {}^{n-2}/_{n-1}}$, built up using constraints rules, which represents a constraint tableau proof (cf. Theorem 6.14, p. 94). For the purpose of comparison, the author gives, in 6.2.3 *Example*, two proofs of the signed formula $\boxed{\leq {}^{1}/_{2}}$ (p $\supset$ (q $\supset$ p)) in three-valued Łukasiewicz logic, one derivation without and one derivation with constraints. This simple case in logic with a small number of values shows how the latter may shorten a derivation. The "constraint" derivation has only nontrivial branches and the bIP problem corresponding to it appears complex. For consolation, the author emphasizes that there exist very efficient algorithms for solving bIP problems. He also remarks that although the bIP problem does not become substantially more complex when n grows, the conventional tree proofs grow considerably.

The subsection 6.2.4, *Complexity of multiple-valued logics*, raises a natural, though as the author says not necessarily interesting, theoretical perspective. Given a logic with many-valued connectives one may consider the classification (of many-valued logics) with respect to the maximum of the sum of all connectives and signs S(i), where S(i) is one of $\boxed{\leq i}$, $\boxed{\geq i}$. The complexity, for example, of the classical logic and Post logic is 2; the complexity of the classical logic with the equivalence connective is 4.

In 6.2.5, *A reduction from multiple-valued deduction to bMIP*, it is shown that a (constraint) tableau can be translated into a single constraint system, and thus it can, in some sense, be linearized. In the following subsection the author shows that by using constraint tableaux it is posssible to give a decision procedure for some infinitely-valued logics. The result generalizes two previous approaches to infinite Łukasiewicz logics by Beavers and by Mundici. Hähnle's approach applies to all infinite logics whose connectives are bMIP-representable.

Section 6.3, *Other inference systems*, briefly presents some techniques which improve classical deductions and which can also be applied in a many-valued environment. Among these are tableau-like methods, decision diagrams, dissolution, and resolution. The point is made that all these techniques are compatible with methods elaborated in the book. Next, the author gives his evaluation of the degree to which the framework proposed in the book meets the desirable properties from the Introduction. Chapter 6 closes with information on experimental implementations of the three-valued theorem prover developed in the years 1990–92 in Germany under a joint project between the University of Karlsruhe and the IBM Science Center in Heidelberg. The author lists some results of that implementation on a set of theorems of interval arithmetic founded on many-valued logic. He compares Carnielli's method and the sets-as-signs approach. The second approach proves to give better results.

7 Chapter 7, *Applications*, is of high importance though not only for automated theorem proving. As is well known, the problem with application of many-valued logics is still among the contested questions of contemporary logic. Hähnle comes directly to the point and asks, "Does many-valued theorem proving have any use?" The question is a variation of the title of the well-known paper by Scott on many-valued logic.

7.1, *Overview*, contains an exhaustive list of actual or possible applications of many-valued logics. The list is divided into eight parts corresponding to different areas of interest. These are: (1) program verification, specification, and synthesis; (2) artificial intelligence; (3) logic; (4) natural language processing; (5) error-correcting codes; (6) interval arithmetic; (7) hardware verification; and (8) quantum physics. The following two sections contain more detailed descriptions of some listed applications.

In 7.2, *Applications of a theoretical nature*, one finds a short account of independence proofs in Hilbert systems and a discussion of Belnap's four-valued propositional logic corresponding to the axiom system of a first-order entailment logic. The four values of this logic form a lattice. They are: Truth (T), Falsity (F), Both (values), and None (its complement). Hähnle cites the 1990 paper in which D'Agostino gave a simple signed tableau based on the set of signs, which is a special case of the technique elaborated in the book.

Next comes a short description of the1974 Suchoń tableau. The author rightly claims that this is the first tableau axiomatization of *n*-valued logic. Further, he remarks that in Suchoń's work the "sets-as-signs" notion is implicitly present. The last theoretic application is an improvement of an S5-implementation by Caferra and Zabel. This concerns using the sets-as-signs for a theorem prover for some propositional modal logics based on the propositional part of a many-valued theorem prover based on the already mentioned work by Carnielli. The author outlines such a use of his framework which permits saving much of the branching that cannot be saved by identifying equivalent states.

Section 7.3, *Applications of a practical nature*, contains remarks on the use of three-valued logic by Gerberding in modeling interval arithmetic, that is, such an extension of the usual arithmetic, which also operates on intervals. In the second part of the section, entitled "Hardware verification," Hähnle lists several potential application areas for a many-valued theorem prover such as verification of genuinely many-valued circuits, test pattern generation by propagation of undefined or error values, and verification of the implementation of gates on the basis of switch level modes. Besides bibliographical references and factography, one may find a number of remarks on the actual use of many-valuedness and related concepts, such as temporal logics in the domain. Closing this short part of the book, the author argues that "genuinely many-valued reasoning techniques are potentially more efficient than reduction [of a temporal multiple-valued logic to classical propositional logic]" and that "many-valued automated deduction is flexible and not restricted to any particular class of logics ...."

8 Chapter 8, *A history of multiple-valued theorem proving*, is an elegant, concise and homogeneous overview of proof systems for many-valued logics. The existing multiple-valued techniques may be, with a small exception, divided into two groups:

systems based on some version of the *resolution rule* and approaches founded upon some *procedures for classical logic*.

Section 8.1, *Resolution-based systems,* starts with an account of a historically first (formulated by Morgan in 1976) many-valued system designed for many-valued theorem proving. The system was primarily motivated by fuzzy logic and, more precisely, it emerged for the purpose of finite approximation of that logic. The main feature of Morgan's solution consists in using the J*i* functions as characteristic functions of logical values of a first-order many-valued logic. Thus, clauses are of the form J*i*(p), for atomic p.

The next resolution-based proof system presented by Hähnle is a system constructed by Schmitt for a three-valued logic introduced by Fenstad et al. in connection with natural language processing.

Slightly more space is then devoted to a nonorthodox Stachniak's resolution system, which is based on Tarski's concept of consequence operation. The model for the approach is the nonclausal resolution rule for the classical logic. Recall that in the latter an important role is played by the disjunction connective, which is present in the conclusion. Stachniak wants to get a tool for dealing with strongly finite logics, that is, consequences which may be characterized by a finite class of finite matrices. The main reason for dealing with nonclausal type resolution is that some logics may not have sufficient expressive power to allow normal forms. To make his approach more flexible and interesting the author assumes that logics taken into account have a disjunction connective and, therefore, an **SF**-logic-resolution rule is defined as a multiple-conclusion rule or, in other words, a branching schema. The system has also the so-called transformation rules and $\Box$-rules (inconsistency rules). Here the role, similar to that of t and f in the classical logic, is taken by the so-called verifiers. The $\Box$-rules are defined on subsets of a given set of verifiers. Finding a (minimal) set of verifiers for a given **SF** logic is a nontrivial and not easy task. Just for that purpose the use of some properties of consequence operation essentially weights. Ending his instructive description of the framework, Hähnle remarks that there are several difficulties connected with the size of the resulting systems due to the branching factor related to the number of verifiers, which are also not easy to find. He concludes that an efficient implementation would be far from straightforward.

Next is a brief review of a resolution-based system for fuzzy logic. Here a fuzzy logic means a logic where formulas have truth-values in the real interval [0,1]. We find information on a first-order Lee and Chang system founded on the designated set [0.5,1] and an extended *n*-valued first-order Post logic of Di Zeno. The latter logic contains additional disjunction and conjunction for each logical value, and its restriction to some connectives is isomorphic with a suitable restriction of Lee and Chang's system. In turn, the author briefly outlines infinite generalizations of Post logics, $\omega + 1$-valued Post logics, constructed and investigated in Poland in the early 1970s. These infinite-valued logics were primarily motivated as a tool for describing logic of programs. Hähnle points out that the infinite algebras, based on the chain $0 < 1 < \cdots < \omega$, are potentially interesting for fuzzy application. This explains their inclusion in this part of the book. The special feature of Post logics is the use of special D_i operators, which correspond to the signs $\boxed{> i}$ and $\boxed{< i}$ in regular logics. The conjunctive normal forms (D-CNF) have literals of the form $\mathrm{D}_i(\mathrm{p})$ or $\neg\mathrm{D}_j(\mathrm{p})$ where p are atomic. Although in general the D-CNF form of a given formula involves infinitely many clauses, it is possible to show, following Orłowska's proof of the Herbrand theorem, that all but finitely many statements of the form D_i(p) are redundant. Consequently, a finite resolution principle is sufficient. The section closes with information on paraconsistent logics and resolution procedures for some systems of this type.

Section 8.2, *Other approaches*, is a review of proof methods for many-valued logics either related to semantic tableaux or having a special character.

In 8.2.1, *Decision diagrams*, we find a short introduction to the method, the main idea of which is to express any binary propositional function with a ternary if-then-else connective. These are known as binary decision diagrams (BDD), and their tree representations are Shannon graphs. The author outlines some modifications of the method which lead to *n*-ary decision diagrams, and discusses some possible improvements in the new environment. In the end, we find information on another approach based on decision diagrams, which represents multiple-valued deduction as a unification problem.

8.2.2, *Approaches based on tableaux and Gentzen calculi*, provides information on the history of the subject, especially important since the author's work departs from the method of semantic tableaux. A brief exposition ends with references to Hähnle's own sets-as-signs approach and related formulations.

In 8.2.3, *Path dissolution by Murray and Rosenthal*, the author explains the idea of an inference for the classical first-order logic, the base of which is removing the so-called *links* which are pairs of complementary formulas. Several remarks concerning the problem of developing a many-valued dissolution rule follow. Hähnle claims that dissolution seems to be a promising technique for theorem proving in many-valued logics.

8.2.4, *Beavers' approach to Łukasiewicz logic L_ω*, brings a short account of a proof method based on the McNaughton criterion of definability of connectives in Łukasiewicz logics.

8.2.5, *Mellouli's three-valued extension of Plaisted's modified problem reduction format*, is an abbreviated exposition of a proof procedure invented as a kind of extension of Prolog-style Horn clause logic programming to the first-order logic.

8.2.6, *General frameworks*, completes an exposition of proof methods. The author mentions that there are two general purpose approaches: Morgan's AUTOLOGIC and more recently Gabbay's *labeled deductive systems*. The first of these works for arbitrary propositional logics is characterized by finite axiom schemata and inference rules, that is, having a Hilbert-style axiomatization. The second technique uses *labels* that are attached to each formula and make semantical or meta-level information explicit in a formal derivation.

The purpose of Section 8.3, *Discussion*, is to give an overview and to evaluate the work done in many-valued theorem proving. The author passes through the catalog of criteria appearing previously. The discussion is rather shallow and not convincing. We learn, for example, that the author is of the opinion that wide applicability excludes all special purpose systems and that the first-order logics are difficult to cover in AUTOLOGIC, adaptability of which depends on a Hilbert-style axiomatization.

9 Chapter 9, *Conclusion*, briefly summarizes the program of the book. It ends with a projection of next steps to be made in the development of many-valued theorem proving.

The book is well organized and clearly written. Although the main emphasis is put upon the author's proper constructions—sets-as-signs and regular logics—the reader obtains exhaustive and extremely concise information on automated proof techniques directly or indirectly related to many-valued logics. This part of the contribution brings perhaps the most comprehensive overview of activities to many-valued theorem proving. The bibliographical documentation is certainly exhaustive: the *References* section numbers13 pages. The conciseness of the monograph is its big advantage. However, for some readers the text might be too advanced to follow at once. Some difficult parts are much more compact than the others.

The quality of editing is excellent. In the course of careful reading it is difficult even to find print errors. Therefore, the reviewer is proud to mention one small flaw. The row (7) in Figure 5.5 (p. 72) should read [2] $\{1\}\,p(c_4)$ instead of [2] $\{{}^1\!/_2\}\,p(c_4)$!

The author's original sets-as-signs construction is ingenious and natural. It is, however, limited to linearly ordered sets of values: the very conception has an algebraic flavor. If we take into account the fact that the matrix method and the algebraic approach to logical calculi is nowadays one of the most powerful tools for investigation, we may come to the conclusion that proposals such as Hähnle's are of particular interest. Possible extensions of the method onto other sets of values should present no conceptual difficulties. In this respect the criteria establishing sufficient and necessary conditions for a logic, defined through matrices, to have a given type of characterization are important. Recall that the author was about to state such a criterion for a class of these finite-valued logics with linearly ordered values. Actually, he only conjectured in Section 5.5 as to which logics of this kind have a uniform notation style tableau system. So, I claim that there remains still a lot to do in this direction.

In the end, I would like to add a general remark concerning the shape of many-valued logical constructions. The author is rather insensible of the distinctions between different existing systems of many-valued logics. True, he is aware of the seeming heterogeneity of many-valued logics and conceptual opaqueness, which makes it hard to compare different systems against each other . . . (see p. 1). However, in Section 2.3. he writes that Rosser and Turquette logics bear a close resemblance to the three-valued Kleene logics and he even generalizes the Kleene's construction onto the case on $n \geq 4$. In connection with that, let us remark first, that the two systems have totally different intuitive and philosophical motivations. Subsequently, for both the implication counts. The implication connectives are definable using the classical pattern, that is, as ¬p ∨ q. The difference now is very transparent: in the case of RT logics,the formula in question defined a standard implication, that is, such that p →q is undesignated if and only if p is designated and q is not. For Kleene, the defined connective is totally different: for example p → p is not a tautology when, as Kleene wished, the truth is the only designated value. Moreover, since functional completeness is often the required property of systems of many-valued logics taken into consideration, one may also tell that from the algebraic and technical point of view such problems are not of great importance. One must, nevertheless, not forget that this is an actual use or application which forces a choice of one bunch of basic connectives or another.

The book is interesting and is an important contribution. I think that it should find many readers not only among those who are working in theorem proving based on many-valued, but also, more generally, those in nonclassical logic.

Grzegorz Malinowski
Department of Logic
University of Łódź
POLAND
email: gregmal@krysia.uni.lodz.pl