Fregean Algebraic Tableaux:
Automating Inferences in Fuzzy Propositional Logic

Christopher Lepock[1] and Francis Jeffry Pelletier[2]

[1] University of Alberta clepock@ualberta.ca
[2] Simon Fraser University jeffpell@sfu.ca

Abstract. We develop a tableau procedure for finding theorems and consequence relations of $\mathrm{RPL}_\triangle$ (i.e., $Ł_\aleph$ extended with constants and a determinacy operator). $\mathrm{RPL}_\triangle$ includes a large number of proposed truth-functions for fuzzy logic. Our procedure simplifies tableaux for infinite-valued systems by incorporating an insight of Frege's. We take formulas of the language to be names for their truth-values, which permits them to be manipulated in the tableaux as if they were algebraic variables. Hence, we call our system FAT, for Fregean Algebraic Tableaux. We have additionally developed an automated procedure for proving theorems using FAT, which we will briefly describe.

1 Background

Before we embark on the main topic of this paper, describing how we approached automating inferences in fuzzy logic, we think it would be useful to address two preliminary topics. One concerns the question of just what is fuzzy logic, and the other concerns the question of why one would wish to automate such inferences. The two topics are interconnected because different answers given to the former question will vary the uses envisaged for fuzzy logic and therefore will vary the reasons one would wish to automate such inferences. And conversely, an investigator's reason to wish to automate a certain class of inferences might affect what sorts of systems are seen as being "really" fuzzy logic.

1.1 The Degree-Theoretic Approach

Fuzzy logic arises from an attempt to formalize what has become known as the "degree-theoretic" approach to vague linguistic terms – terms with borderline cases, in which they seem neither to clearly apply nor not to apply. Consider a man whose head is partially covered with hair, so that it is not clear whether he is bald or not. One approach to vagueness (the epistemic theory) maintains that, although this man either *is* bald or *is* not-bald, we do not know or cannot get enough information to determine whether the man is bald or not. On other approaches (truth-value gap theories or supervaluationism), the problem is that the predicate bald is neither true or false in this case. The basic idea of the degree-theoretic approach is that a borderline case of a vague predicate is neither strictly true nor false, but it does have *some* degree of truth. It is *somewhat* true to say that the man is bald and *somewhat* true to say that he is not bald. Thus it might seem that a predicate like bald can actually have three truth-values: true, false, and halfway-in-between.

A three-valued predicate, however, will still have borderline cases; consider a case on the borderline between its being false that the man is bald and its being half-true that the man is bald. By the reasoning that led us to a third truth-value, it seems we should now introduce two more, one between false and half-true and one between half-true and true. But this argument can be repeated ad infinitum, giving rise to so-called higher-order vagueness. Thus we are driven to the central tenet of the degree-theoretic approach (due to [1]), that there are infinitely many degrees of truth between completely false and completely true. Fuzzy logics model this by taking the truth-values to be all the reals or rationals[1] in the interval [0..1], with 0 being complete falsehood and 1 being complete truth.[2]

There is a great deal of heated controversy over the merits of the degree-theoretic approach: whether propositions really are true to degrees (espoused by, e.g., [3]), or whether this is merely a convenient way of modeling vague propositions ([4]); whether the world contains fuzzy objects and fuzzy properties, or whether only natural language terms have degrees of truth (see [5] for the former view and, e.g., [6, 7] against it); even whether modeling vagueness with fuzzy logic is fruitful or useless (see [8]). We will not adjudicate this difficult issue here.

Rightly or wrongly, the degree-theoretic approach has been taken in a very wide range of areas. Problem-solving is replete with vagueness, gaps in knowledge, and the like; and thus fuzzy logic has been proposed as a solution to a great variety of problems. The vagueness of natural-language terms has led many to think that an adequate description of language must be couched in fuzzy logic. Furthermore, natural language has been seen as embodying many different sorts of inexactitude that can best be understood by averting to a fuzzy logic. ([2] and [9] were among the early champions of this view; but there are now many advocates, including [10–12], and many others. The problems for this approach are well-summarized in [6, 8].)

The application of fuzzy logic in control systems and other practical applications is well-known, and we will not attempt to survey this field. As noted above, some theorists (e.g., [5]) have argued that reality is "essentially fuzzy" and can only be accurately described by a fuzzy logic. The central degree-theoretic tenets can also be interpreted in terms of knowledge of reality instead of reality itself, the idea being that information about reality comes in different "levels of granularity", and that we sometimes grasp one such level and other times a

[1] Theorists differ on which is to be preferred. Some truth-functions used in some systems, e.g., Zadeh's "dilation" operator $\phi^{0.5}$, can only be defined if the truth-values include all the reals. For the truth-functions with which we are concerned here, the stock of tautologies is the same either way.

[2] In some systems, the truth-values are not linearly ordered; see [2].

different level. But this sort of differential "grasping" can only be characterized by a fuzzy logic. (This interpretation of fuzziness leads directly into fuzzy expert systems. One of the earliest to recognize this was [13]. For the next two decades an immense literature developed around this idea.) Some have thought that probability theory (and its various uses, for example in quantum theory) could be derived from fuzzy logic, thereby making fuzzy logic be more basic than probability theory (see [12, 14–16]; see [17] for an alternative). Very many have thought that the human perception system is "essentially fuzzy", and that this calls for a fuzzy logic to describe the results of the various perceptual subsystems (see the discussion in [18] concerning the formal properties of such an approach, and the references to those who have embraced it). Many others have argued that the human reasoning system embodies "fuzziness" at its very core. (See [19] for a Piagetian-based system of children's linguistic and conceptual development; also [20]). And from this many people have concluded that the whole notion of choice (including risk analysis) in any social system is best characterized by a fuzzy logic; and as well, fuzzy logic should form the basis for the understanding of an individual's psychological makeup ([21–23]). Finally, there are many "paradoxes", such as the liar, the heap, and others, which are seen by some proponents as best answered by embracing fuzzy logic. ([5] is perhaps the most vociferous proponent here, but there are plenty of others; see, e.g., [3] for an analysis of the paradox of the heap.)

1.2 Fuzzy Logic

To accommodate the intuitions of the degree-theoretic approach to vagueness and the other areas mentioned above, fuzzy logicians have both extended the interpretation of the classical two-valued connectives to the [0..1] interval of real numbers and have invented some new connectives with no counterparts in classical logic. A wide variety of truth-functions has been proposed, and there is not much consensus on which connectives are to be preferred. It is generally agreed, first, that generalizations of classical connectives should agree with the classical functions on the classical values 0 and 1, and second, that conjunction should be a t-norm, disjunction should be a t-conorm, and implication should be a t-norm's residuum. (See [24] for discussion.) But these constraints have not been universally accepted; more importantly, they do not uniquely determine the choice of connectives. Another clear desideratum for a putative logic of vagueness (but, again, one not universally accepted) is to have the expressive power to "talk about" vague formulas. It does us little good to have infinitely many truth-values when we can only separate formulas that always take the value 1 from those that don't – theorems from non-theorems. (More on this below.) In keeping with the emphasis many fuzzy theorists place on practical value, it is sometimes argued (e.g., by [4]) that there are no categorically best formalizations of the truth-functions, and the particular demands of and constraints upon the particular task should determine which formalization should be used.

The most popular versions of *and, or*, and *not* seem to be those originally proposed by [1]. Using $[\![\phi]\!]$ to mean the truth-value of ϕ, these are defined as:

$$[\![\phi \vee \psi]\!] = \max([\![\phi]\!],[\![\psi]\!])$$
$$[\![\phi \wedge \psi]\!] = \min([\![\phi]\!],[\![\psi]\!])$$
$$[\![\neg\phi]\!] = 1-[\![\phi]\!]$$

For implication, a strong case can be made that the most generally best-suited definition for fuzzy purposes is the Łukasiewicz conditional, $\supset$:

$$[\![(\phi \supset \psi)]\!] = \min(1, 1-[\![\phi]\!]+[\![\psi]\!])$$
$$\text{(i.e.,} \quad = 1-([\![\phi]\!]-[\![\psi]\!]), \text{ if } [\![\phi]\!]\geq[\![\psi]\!]; = 1 \text{ otherwise)}$$

(See [25], and the arguments of [6, pp. 114-8] and [26, pp. 366-7].) However, many other implication connectives have been proposed.

With respect to the trend of inventing new connectives, it is easy to use the fact that one has the infinite domain [0..1] to define new truth-functions. Many of the connectives that are invoked like this were constructed with specific usages in mind, especially for dealing with natural language constructions. For example, while a vague word like *tall* is said to designate a fuzzy set and describe a certain evaluation curve when it is plotted against heights, the intensifier *very* is said to act as a squaring function so that the values on the curve for *very tall* would be the square of the corresponding values on the curve for *tall*. Thus, if $[\![$George is tall$]\!]$=0.7, then $[\![$George is very tall$]\!]$=0.49. ([10, 27] also use the square function for some specific uses of *most* and *usually*, in addition to the interpretation of *very*.) Weighted averages, geometric means, and many other functions have been suggested as possible interpretations of natural language expressions.

1.3 Theorem Proving for Propositional[3] Fuzzy Logic

Most approaches to fuzzy logic start with semantics. In other applications of logic, theorists generally start with a set of formulas that they wish to be true, and then find a semantics that validates them; but in fuzzy logic, it is most common to begin by defining the truth-functions and then determining what formulas including them are tautologous. Nonetheless, we must have some way of determining what the tautologies of these different systems actually are, to know what we are espousing when we endorse those systems, and to understand just what those connectives mean. This is particularly important given the wide range of proposed truth-functions for fuzzy logic. To decide between, say, two different accounts of the conditional, we must consider whether which one makes true the greatest number of formulas we find intuitively acceptable, and which leaves as nontautologies the greatest number of formulas we find intuitively unacceptable. For instance, degree theorists would presumably want to define disjunction in such a way that the law of excluded middle, $p \vee \neg p$, is not a tautology. But since

[3] We restrict ourselves to consideration of theorem proving for propositional fuzzy logic. The theorems of predicate $Ł_{\aleph}$ are not recursively enumerable (see [24]), making the lack of automated proof procedures for that system the least of anyone's worries. The question of how best to axiomatize anything about degree-theoretic intuitions in predicate logic is as yet still open.

there seems not to be any reason why DeMorgan's laws should be thrown out along with it, we would not want accounts of conjunction and disjunction for which these laws are not tautologies.

As we already mentioned, we are not interested in determining whether fuzzy logic is the right way to approach the various subjects to which it has been applied. Given the tremendous interest in the area, however, there is a real need for theorem-proving procedures for fuzzy logic. And just as theorem-proving procedures can help determine the relative merits of different formalizations of the degree-theoretic approach, so too in characterizing different systems by means of an automatic theorem prover, we may acquire a better understanding of the merits of that approach as a whole.

Given that there are many competing systems of fuzzy logic, it is particularly valuable for decision procedures to cover as many different truth-functions on [0..1] as possible. We have thus focused on a system, $\mathrm{RPL}_\triangle$,[4] that contains a substantial number of the connectives that can be found in the literature, including some truth-functions with no classical counterparts. It also has the capacity to "talk about" sentences that take intermediate truth-values, approximately true consequence-relations, and so forth. There are, however, many truth-functions – more precisely, $\aleph_1$ many – that are not definable in this system, which include some of those functions that have received a great deal of attention in the literature. The most notable omissions are perhaps the Goguen [2] conditional, conjunction, and disjunction, Zadeh's ϕ^2 and $\phi^{0.5}$, and the pseudo-Łukasiewicz family of conditionals. The reason for our omission concerns certain facts about our theorem prover – facts which we are not yet in a position to describe.

2 $\mathrm{RPL}_\triangle$

As we noted in §1.2, $Ł_\aleph$ seems to be the most popular fuzzy formal system. There is, however, a sense in which $Ł_\aleph$ is not really a fuzzy logic. For, although there are infinitely many truth-values, we can only "talk about" perfectly true or perfectly false formulas. For example, given that ⟦Mary is tall⟧=0.5 and ⟦George is tall⟧=0.7, there is no way in the language to express the conclusion that George is taller than Mary to degree 0.2 (or to any other degree). We thus cannot model the approximately true sentences and approximately valid arguments that are the hallmark of fuzzy logic because we cannot say how true an approximately true sentence is, or how truth-preserving an approximately valid inference is. $\mathrm{RPL}_\triangle$ extends $Ł_\aleph$ by giving it the full capacity to describe the truth-values of formulas within the system. To get $\mathrm{RPL}_\triangle$, we add a denumerable infinity of constant truth functions:

$[\![\mathrm{C}_i]\!]=i$, for each rational $i \in [0..1]$,

and a bivalent determinacy operator:

[4] $\mathrm{RPL}_\triangle$ extends [28]'s RPL by adding a determinacy operator (in our presentation, the J-operators). The system was first proposed in [29].

$[\![\mathrm{J}_i\phi]\!]= 1$ if $[\![\phi]\!]=i$, 0 otherwise.

$\mathrm{RPL}_\triangle$ can describe the truth-values of its formulas, since

$[\![\mathrm{C}_i \equiv \phi]\!]= 1$ iff $[\![\phi]\!]= i$,
$[\![\mathrm{C}_i \supset \phi]\!]= 1$ iff $[\![\phi]\!]\geq i$,
$[\![\phi \supset \mathrm{C}_i]\!]= 1$ iff $[\![\phi]\!]\leq i$,
$[\![\neg \mathrm{J}_i\phi]\!]= 1$ iff $[\![\phi]\!]\neq i$,
$[\![\neg \mathrm{J}_1(\mathrm{C}_i \supset \phi)]\!]= 1$ iff $[\![\phi]\!]>$ i,
$[\![\neg \mathrm{J}_1(\phi \supset \mathrm{C}_i)]\!]= 1$ iff $[\![\phi]\!]>$ i.

(Where $(\phi \equiv \psi)$ is defined as $(\phi \supset \psi) \wedge (\psi \supset \phi)$.)

Because $\mathrm{RPL}_\triangle$ can describe the truth-values of its formulas, it can handle many of the tasks fuzzy logicians expect from a language. It allows one to model approximately true statements and approximately valid arguments by stating the degree of truth the statement has or the argument preserves.[5] For instance, suppose we want to say that the inference from ϕ to ψ is *approximately* valid, in that while $[\![\psi]\!]$ might be strictly less true than $[\![\phi]\!]$, it cannot be very much less true. Then the formula $(\mathrm{C}_i \supset(\phi \supset \psi))$ indicates that $[\![\psi]\!]\geq[\![\phi]\!]-i$. If we want to add that the inference from ϕ to ψ is *only* approximately valid, we can conjoin that formula with $\neg \mathrm{J}_1(\phi \supset \psi)$, which says that $[\![\psi]\!]<[\![\phi]\!]$.

$\mathrm{RPL}_\triangle$ additionally allows us to define a wide array of truth functions that have been proposed in the literature. As we already noted, there are some very common and useful functions that cannot be defined in it. But still a very large proportion of the operators proposed in the literature can be defined in $\mathrm{RPL}_\triangle$. Table 1 of the Appendix gives a selection of these functions.[6]

There is a simple procedure by which we can define further operators in $\mathrm{RPL}_\triangle$. Suppose for some operator O, there are a finite number of input conditions $c_1, \ldots, c_n$ such that whenever condition c_i is satisfied O takes the value v_i. Suppose further that each of $i_1, \ldots, i_n$ can be described by a formula of $\mathrm{RPL}_\triangle$. Then O can be defined by the disjunction

$$(\mathrm{J}_1 c_1 \wedge \mathrm{C}_{v_1}) \vee (\mathrm{J}_1 c_2 \wedge \mathrm{C}_{v_2}) \vee \ldots \vee (\mathrm{J}_1 c_n \wedge \mathrm{C}_{v_n}).$$

Examples of this procedure can be found in Table 1, since it was used to define a number of the operators listed there.

[5] It will no doubt be objected that to say that a formula has a particular intermediate truth-value is not the same as saying that it is vague, and thus that even $\mathrm{RPL}_\triangle$ has insufficient expressive power to be a proper logic of vagueness. For instance, what truth-value is "roughly true"? These considerations have led some to adopt 'higher-order' fuzzy logics, in which the truth-values are themselves fuzzy sets ([10]). Such systems, however, have no meaningful syntax, according to [30], and the semantics is computationally too demanding to be worth the increase in expressive power [31, p. 17]).

[6] For discussion of these operators (among others), see [31, pp. 51-94, 304-12], [24, 32–34].

3 History of Fuzzy Logic Theorem Proving

The initial development of infinite-valued logics was in terms of axioms (see [35]). Axiomatically defined systems make it extremely difficult to characterize an algorithmic method by which one can determine the set of theorems and valid arguments, and indeed, even make it very difficult to see any clear heuristic methods to develop proofs. As a result, most attempts to automate an inference system for these fuzzy logics have focused their attention on the semantic evaluation of formulas and arguments of the logic. Of particular interest are those systems that generate semantic tableaux – methods that employ the semantic values of formulas to determine validity of arguments. Generally speaking, one attempts to assign an "undesignated" value to a formula that is being tested for theoremhood, and looks to the required assignments to the immediate subformulas that such an assignment would require. One continues this method on the subparts until the atomic parts are reached, whereupon one can immediately see whether or not one of these atomic sentences would be required to take on two different values. If so, then the initial attempt to make the formula "undesignated" is not possible and therefore it must be a theorem. If no such impossibility is encountered, then the method generally allows one to retrieve some particular assignment(s) that demonstrates that the formula need not be "designated".

In the interests of space, we will only discuss a few procedures in detail, but there are a number of others worth mentioning. There are several procedures that, like those of [36] or [37], apply just to the fragment of $Ł_{\aleph}$ having only the truth-functions $\wedge, \vee, \neg$.[7] [39] gives a procedure that can determine theoremhood in any decidable propositional fuzzy logic. As the run time of this algorithm is at least doubly exponential, its use is not practically feasible. [29] outlines a procedure that finds a finite number of subintervals of [0..1] such that if the formula in question takes the value 1 on an arbitrarily chosen value from each subinterval, the formula takes the value 1 on the entire range of truth-values. This procedure also appears to be quite complex in operation.

3.1 Tableau Systems for $Ł_{\aleph}$

[40] and [41–44] describe methods which are closely related to one another and to the method we will outline below. Although these systems were designed for $Ł_{\aleph}$, they can be extended to cover the full $RPL_{\triangle}$. Both are, in effect, tableau systems. Hähnle's constraint tableaux are extensions of signed tableaux to infinite-valued logics; Beavers's procedure consists of decomposing a formula into a set of equations to be checked for satisfiability, although he does not present these equations in tableau format.

The central idea of Beavers's procedure is that each formula of $Ł_{\aleph}$ corresponds to a set of linear polynomial functions, determined by the truth-functions

[7] See [38] for a detailed discussion and analysis of [36].

corresponding to the connectives found in the formula. Each formula will exhaustively divide the [0..1] interval into a bunch of disjoint subintervals, where each of these subintervals is describable by a linear function. The formula as a whole is merely the piecewise combination of these functions. Therefore, to find whether a formula is a logical truth (always takes the value 1), the task is to determine whether this combination of functions ever takes a value less than 1; and this amounts to determining whether any of the functions describing the formula over a subinterval takes a value less than 1. Beavers reports various implementations of an algorithm that makes use of a linear programming package to evaluate the polynomials generated by the formulas. One of the implementations he mentions has the linear programming package find the minimal value that satisfies the equations corresponding to the initial formula. Therefore he could have used this to determine whether formulas are logically true even in cases where the set of designated values was allowed to be all values between [r..1], and not just 1.

The Deduction Theorem in its standard form, i.e., $\Gamma, \phi \vDash \psi$ iff $\Gamma \vDash (\phi \supset \psi)$, is not valid for $Ł_{\aleph}$; this is because $(\phi \supset \psi)$ is a tautology iff $[\![\phi]\!] \leq [\![\psi]\!]$ on every valuation, while $\phi \vDash \psi$ iff $[\![\psi]\!] = 1$ on every valuation on which $[\![\phi]\!] = 1$. Hence there is no straightforward link between tautological conditionals and consequence relations in $Ł_{\aleph}$ or its extensions. ($RPL_{\triangle}$ has a deduction theorem in the form $\Gamma, \phi \vDash \psi$ iff $\Gamma \vDash (J_1\phi \supset \psi)$.) Beavers also proves a quasi-deduction theorem for his logic by introducing a new connective $\supset^n$, which is interpreted like $\supset$ except that the antecedent's value is multiplied by n, and he extends his procedure to decide this sort of consequence relation. We then have the deduction theorem in the form $\Gamma, \phi \vDash \psi$ only if $\exists$n $\Gamma \vDash (\phi \supset^n \psi)$. He does not apply his procedure to other sorts of consequence relations, particularly approximately true ones (since these cannot be expressed in $Ł_{\aleph}$).

As noted above, Hähnle's constraint tableaux are a variant on signed tableaux for finite-valued logics, which consist of formulas prefixed with signs indicating their truth values. In constraint tableaux, rather than representing individual truth values, the signs place constraints on the truth values of their formulas. The constraints can be either numerical constants or variables and are written inside a box; for instance

$$\boxed{<1}\ \mathrm{A} \qquad \text{and} \qquad \boxed{\leq j}\ \mathrm{A}$$

indicate, respectively, that

$$[\![\mathrm{A}]\!] < 1 \qquad \text{and} \qquad [\![\mathrm{A}]\!] \leq j.$$

The decomposition rules of constraint tableaux result in new constraints applied to the formulas of which the original was composed, and inequations (equations with inequality relations) representing the relations between the variables used in the signs. There are no branching rules; instead, Hähnle uses binary variables (variables that can take only 0 and 1 as values), which represent the same information as new branches of the tableau would contain. For instance, the decomposition rules for $\supset$ are:

$$\begin{array}{c} \boxed{\leq i}\ (A \supset B) \\ \Downarrow \\ \boxed{\geq (1 - i + jy)}\ A\ \ (y \leq i) \\ \boxed{\leq (j + y)}\ B \end{array} \qquad \begin{array}{c} \boxed{\geq i}\ (A \supset B) \\ \Downarrow \\ \boxed{\leq (1 - i + j)}\ A \\ \boxed{\geq j}\ B \end{array}$$

In the rule on the left, y is a binary variable. The information contained in the signs can also be entered into the series of inequations; for instance, from

$$\boxed{\leq k_2}\ B$$

we can infer $[\![B]\!] \leq k_2$. To show that a formula A is a theorem of fuzzy logic, we start a tableau with

$$\boxed{\leq i}\ A$$

and decompose A fully. We then evaluate the resulting inequations, and say that A is a theorem if and only if the least i that satisfies every inequation is 1.

Applying the decomposition rules for these tableaux is simple, particularly since there are no branching rules. The difficult part is determining whether a branch is open or closed. To do this, Hähnle treats the series of inequations associated with a tableau as a problem in linear programming, and in his implementation he passes these inequations off to a linear programming package.

3.2 Problems with Constraint Tablaux

Constraint tableaux were developed in keeping with signed tableaux for classical propositional logic. The central idea is that each line of a branch contains a syntactic entity (a formula of the logical language), together with a sign stating semantic information about the truth-value of the formula. This arrangement of information works well for classical logic, in which signs need represent only whether a formula takes the value 1 or 0. In tableaux for infinite-valued logic, the signs carry a great deal of information. They represent it, moreover, in a very inefficient manner.

Constraint tableaux are unnecessarily complex, a point that can only be fully appreciated after comparison with a more efficient procedure. A more apparent problem is that each application of, for example, a conditional decomposition rule in constraint tableaux forces the introduction of a new variable (the variable j in the rule listed above). The truth-value of a Łukasiewicz conditional depends on the relative values of the antecedent and consequent; this new variable, being found in the signs for both antecedent and consequent, carries information permitting the constraints introduced on the two formulas to be related to each other. But since an application of a decomposition rule introduces no new information, only rendering explicit that which is already there in the line being decomposed, there is no reason to introduce new variables with each new step in the tableau. The introduction of new variables is an artifact of the unnaturalness of the use of signs in tableaux for infinite-valued logics, one which may be easily remedied by use of an observation of Frege's, to which we now turn.

4 Fregean Algebraic Tableaux [FAT]

It is easy to see that the cognitive significance of a name is more than just its bearer. 'Bertrand Russell' and 'the author of *Principles of Mathematics*' have different cognitive significance, despite the fact that Bertrand Russell and the author of *The Principles of Mathematics* are identical. It is perfectly legitimate, for instance, to ask whether Bertrand Russell was the author of *Principles of Mathematics*, but if someone asked whether Bertrand Russell was Bertrand Russell, we would naturally assume he was joking. Frege accounted for this by proposing that a name has both a sense (*Sinn*) and a reference or denotation (*Bedeutung*). A name's reference is the object it names, and its sense is the way it presents that object. 'Bertrand Russell' and 'the author of *Principles of Mathematics*' designate the same object, but have different senses – i.e., present the object in different ways.

Frege extended this view of names to propositions. A proposition, he said, denotes its truth-value. All true propositions denote the same object, 'the true', and all false ones designate 'the false'. Besides its denotation, a proposition also has a sense, which can be thought of as the way it presents its truth-value.[8] So distinct but materially equivalent propositions, like *The moon orbits the earth* and *Ducks are birds*, designate the same truth-value, but present it in different ways. This is clearest in the case of non-atomic propositions; $(p \supset q)$, for instance, presents its truth-value as a particular function of the values of p and q, whereas $\neg p \vee q$ presents the same truth-value as a different function of the values of p and q. Note that on this view the material equivalence of two propositions can be expressed by $p = q$, since this formula is interpreted as saying that the object designated by p is identical to that designated by q.

Our tableau system applies Frege's insight by taking formulas to designate truth-values. Since we are concerned with fuzzy logic, we take the truth-values to be the rationals or reals (see fn. 1) in [0..1]. Each formula names exactly one value in this interval. On this approach, we can intelligibly mix logical formulas and arithmetical signs in the same expressions. This makes expressions like '$p \leq k$' meaningful; this expression says that the truth-value p names is less than k. If p designated some other object besides its truth-value, '$p \leq k$' would be ill-formed. Of course, we may not know what truth-value a formula designates, just as we might know that 'the guy who stole my wallet' designates a person without knowing which one. But we can at least express what information we can glean about the designation of a formula.

Our Fregean interpretation of propositions gives us interpretations of the logical operators as well. As already noted, a formula containing a connective can be thought of as presenting its truth-value as a function of the values it takes as inputs. On the conventional approach, constant truth functions are taken to be entities that take the same truth-value on every valuation. One might instead think of a constant as a proposition that wears its truth-value on its sleeve. Since

[8] Although Frege called senses of sentences 'thoughts', he firmly denied that they were mental objects and instead held that they "existed in reality".

we take propositions to name their truth-values, we can take constants to just be the truth-values they are assigned. The constant C_1 refers to the same thing as the number '1', and so we need make no distinction between them.

By taking propositions to name truth-values, we need not impose the arbitrary distinction between formulas and signs found in constraint tableaux; all the information that would be expressed in a sign affixed to a formula can just be said about the formula itself. The result is a system that can combine the advantages of a tableau procedure with the efficiency of Beavers's purely semantic procedure.

Part of our motivation in developing Fregean tableaux was to facilitate the teaching of fuzzy logic. The only way to really understand a logical system, especially one that proposes such far-reaching changes to our understanding of truth, is to work within it, and thus learn not just what is tautologous, but why. Tableaux are familiar and intuitive, and the part of the task assigned to the linear programmer in an automated procedure requires only high-school algebra to perform by hand (for fairly simple formulas, at any rate).

4.1 The tableau procedure

In the interests of space, we will present the procedure here for the basic connectives of $\mathrm{RPL}_{\triangle}$: $\supset, \neg$, the constants, and the J-operators. As noted above, a plethora of further connectives can be defined in terms of these. The complete system implemented in our automated prover [45] uses separate rules for many of these connectives (which eliminates a number of duplicated branches).

We begin with a conclusion A, which we wish to prove, and a possibly empty set of premises Γ. The procedure has three steps.

Step 1. If one wants to prove that the conclusion takes the value 1 whenever the premises do, the first line of the tableau should be A <1. Then, for any formula $\phi \in \Gamma$, a line of the form $\phi \geq 1$ should be entered in the tableau.

One may want to prove restrictions on the truth-values of one's conclusion, or put restrictions on the truth-values of the premises, other than that they are absolutely true. There are two ways to do this. One can state the restrictions using the language of $\mathrm{RPL}_{\triangle}$, and then use the rules applying to those connectives to decompose the formulas in the tableau. It is easier, however, to enter such restrictions directly into the tableau as inequalities. So for instance, if one wants to prove that $[\![\mathrm{A}]\!] \geq n$ for some n other than 1, the first line of the tableau would be A $< n$. To say that for some premise B, $[\![\mathrm{B}]\!] \geq k$, one would enter the line B $\geq k$. Thus we could evaluate the correctness of the claim: "whenever each of the premises takes a value grater than or equal to k, the conclusion must take a value $\geq n$. (In the examples of this paper, we always use 1 as the "designated value.")

Step 2. The following rules may be applied at any point in the derivation, and come in two categories.

(a) *Replacement rules.*

Rule C: In any line of a tableau, a constant truth function may be replaced with the truth-value it signifies.
Rule N: In any line of a tableau, any expression of the form $\neg\phi$ may be replaced with $1 - \phi$. (E.g., from $\neg p \geq C_{0.5}$, infer the line $\neg p \geq 0.5$ by rule C, and then $1 - p \geq 0.5$ by rule N.)
The justification of these rules (as should be obvious) is that, in keeping with our Fregean outlook, we take $\neg\phi$ and C_i to be names for $1 - [\![\phi]\!]$ and i, respectively.
(b) *Decomposition rules*. These can be applied at any time to any formulas of the forms specified. Names are given above the rules for ease of reference.

$$\begin{array}{c} \supset\text{GE} \\ \phi \supset \psi \geq \chi \\ \downarrow \\ \phi \leq \psi - \chi + 1 \\ \chi \leq 1 \end{array} \qquad \begin{array}{cc} \multicolumn{2}{c}{\supset\text{LE}} \\ \multicolumn{2}{c}{\phi \supset \psi \leq \chi} \\ \swarrow & \searrow \\ \phi \geq \psi - \chi + 1 & \phi \leq \psi \\ & \chi \geq 1 \end{array}$$

$$\begin{array}{c} \supset\text{SG} \\ \phi \supset \psi > \chi \\ \downarrow \\ \phi < \psi - \chi + 1 \\ \chi < 1 \end{array} \qquad \begin{array}{cc} \multicolumn{2}{c}{\supset\text{SL}} \\ \multicolumn{2}{c}{\phi \supset \psi < \chi} \\ \swarrow & \searrow \\ \phi > \psi - \chi + 1 & \phi \leq \psi \\ & \chi > 1 \end{array}$$

$$\begin{array}{ccc} \multicolumn{3}{c}{\text{JGE}} \\ \multicolumn{3}{c}{\text{J}_i\phi \geq \chi} \\ \swarrow & \downarrow & \searrow \\ \phi \geq i & \phi > i & \phi < i \\ \phi \leq i & \chi \leq 0 & \chi \leq 0 \\ \chi \leq 1 & & \end{array} \qquad \begin{array}{ccc} \multicolumn{3}{c}{\text{JLE}} \\ \multicolumn{3}{c}{\text{J}_i\phi \leq \chi} \\ \swarrow & \downarrow & \searrow \\ \phi > i & \phi < i & \phi \geq i \\ \chi \geq 0 & \chi \geq 0 & \phi \leq i \\ & & \chi \geq 1 \end{array}$$

$$\begin{array}{ccc} \multicolumn{3}{c}{\text{JSG}} \\ \multicolumn{3}{c}{\text{J}_i\phi > \chi} \\ \swarrow & \downarrow & \searrow \\ \phi \geq i & \phi > i & \phi < i \\ \phi \leq i & \chi < 0 & \chi < 0 \\ \chi < 1 & & \end{array} \qquad \begin{array}{ccc} \multicolumn{3}{c}{\text{JSL}} \\ \multicolumn{3}{c}{\text{J}_i\phi < \chi} \\ \swarrow & \downarrow & \searrow \\ \phi > i & \phi < i & \phi \geq i \\ \chi > 0 & \chi > 0 & \phi \leq i \\ & & \chi > 1 \end{array}$$

Step 3. When no further decomposition or replacement rules can be applied, linear programming can determine whether the resulting set of inequations is feasible. When calculating by hand, this question can be resolved by simple algebra. (In a hand tableau, a branch closes if one can derive impossible inequalities or equations indicating that some formula must take a truth-values outside [0..1]. The lines of any complete open branch containing no logical symbols except propositional variables describe a set of valuations where all the premises take a value greater than the designated value and the conclusion takes a lower value – i.e., a set of counterexamples to the inference or formula being investigated.)

4.2 A comparison of FAT and constraint tableaux

The chief advantage of FAT over constraint tableaux is that the former never introduce new variables to be calculated over; all information is represented using subformulas of the premises and conclusion at the head of the tableau. The rules are also simpler (compare Hähnle's $\leq i\ (A \supset B)$ rule with our $\supset$LE), which can lead in some cases to drastically simpler tableaux. For example, consider a formula of the form

$$p \supset (p \supset (\ldots . \neg p) \ldots),$$

with p repeated n times excluding the negated instance. Formulas of this form take the value 1 for $[\![p]\!] \leq \frac{n-1}{n}$, and the value $\frac{1-[p]}{n}$ otherwise. Thus, none is a tautology of $Ł_{\aleph}$. When comparing the two systems, assume for simplicity that the procedure applies all possible decomposition rules and then checks the results for feasibility, without checking for closed branches before all formulas have been decomposed. This is obviously inefficient, but so is using the linear programming module to check if branches that have not been fully decomposed are closed; since there are many ways of deciding which branches to continue to decompose and which not to, we will not assume any particular routine for doing so.

Under that scenario, the reader can easily verify the following statistics. A constraint tableau generated in that manner for a formula of the above form has $2n$ branches (from the possible configurations of the n binary variables generated); the FAT tableau has $n + 1$. Each branch of the constraint tableau has $2n$ lines containing no logical operators (which are the lines that the linear programmer must work with); the number of lines of that type in the FAT vary from 1 to n on different branches. Each branch of the constraint tableau has $n + 1$ distinct variables; each branch of the FAT has one (namely, p). The Fregean tableau allows the linear programming module to calculate the result from far less data and with far fewer possibilities to consider.

The moral of the story is: do less work; get FAT.

5 Implementation

The FAT method has been implemented in Java and runs on the usual platforms (Linux/Unix, Mac, Windows). The user is allowed to vary the designated value, so that we can test whether, for example, if all premises take at least the value r then so does the conclusion. There is also an option, for use with invalid arguments, to find the lowest value that the conclusion could take when the premises are all designated. (Sort of a "what is the 'most invalid' that the argument *could* be?" test.) The implementation and experiments with interesting formulas are described in a separate paper, [45], and will not be discussed here.

6 Conclusion

The purpose of this study was to describe a method to determine the validity of arguments in fuzzy logic which would be an extension of methods already in the literature, but which would be superior to them in various ways. One way we think our method is superior is simply in terms of coverage. The existing publications do not consider all the connectives we have given an account of. This is not to say that, for example, Beavers's and Hähnle's methods could not be extended to the class of connectives we describe; rather, we merely point out that they haven't been extended in this way.

A more important improvement, in our opinion, is the underlying explanation our method gives to determining argument validity in fuzzy logic. Our tableaux rules are much more natural extensions of finitely-many-valued rules, we think. And we attribute this to the "Fregean insight" that formulas are names of truth values. Each tableaux rule directly gives a statement about what truth value the subformulas must name, and for that reason they are much easier to understand than either the "general polynomial formulas" generated in Beavers's method or the series of inequations that Hähnle's rules give.

Of course, in the end all three methods make use of a linear programming package, and so in that sense they are all similar. But we think that the more natural statement of the methodology inherent in our method makes it preferable both for pedagogic reasons and for ease of understanding in novel applications.

References

1. Zadeh, L.: Fuzzy sets. Info. and Contr. **8** (1965) 338–353
2. Goguen, J.: The logic of inexact concepts. Synthèse **19** (1969) 325–373
3. Machina, K.: Truth, belief, and vagueness. Jour. Phil. Logic **5** (1976) 47–58
4. Gaines, B.: Foundations of fuzzy reasoning. Inter. Jour. Man-Machine Studies **8** (1976) 623–668
5. Kosko, B.: Fuzzy Thinking. Hyperion, NY (1993)
6. Williamson, T.: Vagueness. Routledge, London (1994)
7. Pelletier, F.: Another argument against vague objects. Jour. Phil. **86** (1989) 481–492
8. Keefe, R.: Vagueness by numbers. Mind **127** (1998) 565–579
9. Lakoff, G.: Hedges: A study in meaning criteria and the logic of fuzzy concepts. Jour. Phil. Logic **2** (1973) 458–508
10. Zadeh, L.: PRUF–a meaning representation language for natural languages. Inter. Jour. Man-Machine Studies **10** (1978) 395–460
11. Kuz'min, V.: A parametric approach to description of linguistic values of variables and hedges. Fuzzy Sets and Systems **6** (1981) 27–41
12. Santamarina, C., Salvendy, G.: Fuzzy sets-based knowledge systems and knowledge elicitation. Behaviour and Information Technology **10** (1991) 23–40
13. Mamdani, E.: Application of fuzzy logic to approximate reasoning using linguistic synthesis. IEEE Trans. Comput. **C-26** (1977) 1182–1191
14. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems **1** (1978) 3–28
15. Brown, C.: The fuzzification of probabilities. In: Proc. Specialty Conf. on Probabilistic Mechanics and Structural Reliability, NY, Amer. Soc. of Civil Engineers (1979)
16. Kosko, B.: Fuzziness vs. probability. Inter. Jour. of Gen. Syst. **17** (1990) 211–240

17. Hisdal, E.: Infinite-valued logic based on two-valued logic and probability. Part 1: Difficulties with present-day fuzzy-set theory and their resolution in the TEE model. Inter. Jour. Man-Machine Studies **25** (86) 89–111
18. Crowther, D., Batchelder, W., Hu, X.: A measurement-theoretic analysis of the fuzzy logic model of perception. Psych. Review **102** (1995) 396–408
19. Carpendale, J., McBride, M., Chapman, M.: Language and operations in children's class inclusion reasoning: The operational semantic theory of reasoning. Developmental Review **16** (1996) 391–415
20. Brainerd, C., Kingma, J.: Do children have to remember to reason? A fuzzy-trace theory of transitivity development. Developmental Review **4** (1984) 311–377
21. Unwin, S.: A fuzzy set theoretic foundation for vagueness in uncertainty analysis. RISK ANALYSIS **6** (1986) 27–34
22. Hesketh, B., Pryor, R., Gleitzman, M.: An application of computerised fuzzy graphic rating scale to the psychological measurement of individual differences. Inter. Jour. Man-Machine Studies **29** (1988) 21–35
23. Hesketh, B., Pryor, R., Gleitzman, M.: Fuzzy logic: Towards measuring Gottfredson's concept of occupational social space. Jour. Couns. Psych. **36** (1989) 103–109
24. Hájek, P.: The Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
25. Smets, P., Magrez, P.: Implication in fuzzy logic. Inter. Jour. Approximate Reasoning **1** (1987) 327–347
26. Paoli, F.: A really fuzzy approach to the sorites paradox. Synthèse **134** (2001) 363–387
27. Zadeh, L.: Fuzzy logic. IEEE Computer **221** (1988) 83–93
28. Pavelka, J.: On fuzzy logic I, II, III. Zeitschr. Math. Logik und Grundlagen der Math. **25** (1979) 45–52,119–134,447–464
29. Morgan, C., Pelletier, F.: Some notes on fuzzy logic. Ling. and Phil. **1** (1977) 79–97
30. Haack, S.: Deviant Logic, Fuzzy Logic: Beyond the Formalism. Univ. Chicago Press, Chicago (1996)
31. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Upper Saddle River, N.J. (1995)
32. Ruan, D., Kerre, E.: Fuzzy implication operators and generalized method of cases. Fuzzy Sets and Systems **54** (1993) 23–38
33. Mizumoto, M., Zimmermann, H.: Comparison of fuzzy reasoning methods. Fuzzy Sets and Systems **8** (1982) 253–283
34. Rescher, N.: Many-Valued Logics. McGraw Hill, NY (1969)
35. Łukasiewicz, J., Tarski, A.: Untersuchungen über den aussagenkalkül. In Tarski, A., ed.: Logic, Semantics, Metamathematics. Oxford UP, Oxford (1930) 38–59 Translated by J.H. Woodger.
36. Kenevan, J., Neapolitan, R.: A model theoretic approach to propositional fuzzy logic using beth tableaux. In Zadeh, L., Kacprzyk, J., eds.: Fuzzy Logic for the Management of Uncertainty. Wiley, NY (1992)
37. Lee, R., Chang, C.L.: Some properties of fuzzy logic. Info. and Contr. **19** (1971) 417–431
38. Entemann, C.: A fuzzy logic with interval truth values. Fuzzy Sets and Systems **113** (2000) 161–183
39. Gehrke, M., Kreinovich, V., Buochon-Meunier, F.: Propositional fuzzy logics: Decidable for some (algebraic) operators; undecidable for more complicated ones. Inter. Jour. Intel. Systems **14** (1999) 935–947
40. Beavers, G.: Automated theorem proving for łukasiewicz logics. Studia Logica **52** (1993) 183–196

41. Hähnle, R.: Automated Theorem Proving in Multiple-Valued Logics. Oxford UP, Oxford (1993)
42. Hähnle, R.: Many-valued logic and mixed integer programming. Annals of Mathematics and Artificial Intelligence **12** (1994) 231–264
43. Hähnle, R.: Proof theory of many-valued logic – linear optimization – logic design: Connections and interactions. Soft Computing **1** (1997) 107–119
44. Hähnle, R.: Tableaux for many-valued logics. In D'Agostino, M., Gabbay, D., Hähnle, R., Posegga, J., eds.: Handbook of Tableaux Methods. Kluwer, Dordrecht (1999) 529–580
45. Pelletier, F., Lepock, C., Li, G., Henkemans, D.: FLAT: Fuzzy logic automated tableaux. Submitted to LPAR (2005)

Table 1: A Selection of Operators Definable in RPL$_\triangle$

T-norms

Name	**Formula i(a,b)**	**RPL$_\triangle$ equivalent**
Bounded sum	max (0,$[\![a]\!]+[\![b]\!]$- 1)	$\neg(a \supset \neg b)$
Drastic intersection	$[\![a]\!]$if $[\![b]\!]=1$ $[\![b]\!]$if $[\![a]\!]=1$ 0 o.w.	$(J_1 a \wedge b) \vee (J_1 b \wedge a)$

T-conorms

Name	**Formula u(a,b)**	**RPL$_\triangle$ equivalent**
Bounded sum	min (1,$[\![a]\!]+[\![b]\!]$)	$\neg a \supset b$
Drastic union	$[\![a]\!]$if $[\![b]\!]=0$ $[\![b]\!]$if $[\![a]\!]=0$ 1 o.w.	$(\neg J_0 a \vee b) \wedge (\neg J_0 b \vee a)$

Implications

Name	**Formula d(a,b)**	**RPL$_\triangle$ equivalent**
Early Zadeh	max[1-$[\![a]\!]$,min($[\![a]\!]$,$[\![b]\!]$)]	$\neg a \vee (a \wedge b)$
Gaines-Rescher[9]	1 if $[\![a]\!] \leq [\![b]\!]$ 0 o.w.	$J_1(a \supset b)$
Gödel[10]	1 if $[\![a]\!] \leq [\![b]\!]$ $[\![b]\!]$o.w.	$J_1(a \supset b) \vee b$
Kleene-Dienes	max(1–$[\![a]\!]$,$[\![b]\!]$)	$\neg a \vee b$
Klir & Yuan 2	$[\![b]\!]$for $[\![a]\!]=1$ 1-$[\![a]\!]$for $[\![a]\!]$,$[\![b]\!]$<1 1 for $[\![a]\!]$<1, $[\![b]\!]$=1	$(J_1 a \wedge b) \vee (\neg J_1 a \wedge J_1 b) \vee (\neg J_1 a \wedge \neg J_1 b \wedge \neg a)$
LR	$[\![b]\!]$if $[\![a]\!]=1$ 1 o.w.	$\neg J_1 a \vee b$
LS	$[\![b]\!]$if $[\![a]\!]=1$ 1 -$[\![a]\!]$if $[\![b]\!]=0$ 1 o.w.	$(\neg J_1 a \vee b) \wedge (\neg J_0 b \vee \neg a)$

[9] Also called Standard Strict.
[10] Also called Standard Star.

Standard sharp	1 for $a < 1$ or $b = 1$ 0 o.w.	$\neg J_1 a \vee J_1 b$
Willmott	$\min[\max(1-[\![a]\!],[\![b]\!]),$ $\max(1-[\![a]\!],[\![a]\!]),$ $\max(1-[\![b]\!],[\![b]\!])]$	$(\neg a \vee b) \wedge (\neg a \vee a) \wedge (\neg b \vee b)$
Wu	1 for $[\![a]\!] \leq [\![b]\!]$ $\min(1-[\![a]\!],[\![b]\!])$ o.w.	$J_1(a \supset b) \vee (\neg a \vee b)$

Quasi-Implications

Name	**Formula d(a,b)**	**$RPL_\triangle$ equivalent**
Star/Star impl.	1 for $[\![a]\!]=[\![b]\!]$ $[\![b]\!]$for $[\![a]\!]>[\![b]\!]$ $1-[\![b]\!]$for $[\![a]\!]<[\![b]\!]$	$[J_1(a \supset b) \vee b] \wedge [J_1(\neg a \supset \neg b) \vee \neg b]$
Star/Strict impl.	1 for $[\![a]\!]=[\![b]\!]$ $[\![b]\!]$for $[\![a]\!]>[\![b]\!]$ 0 for $[\![a]\!]<[\![b]\!]$	$[J_1(a \supset b) \vee b] \wedge J_1(\neg a \supset \neg b)$
Strict/Star impl.	1 for $[\![a]\!]=[\![b]\!]$ 0 for $[\![a]\!]>[\![b]\!]$ $1-[\![b]\!]$for $[\![a]\!]<[\![b]\!]$	$J_1(a \supset b) \wedge [J_1(\neg a \supset \neg b) \vee \neg b]$
Wu implication 2	0 for $[\![a]\!]<[\![b]\!]$ $[\![b]\!]$o.w.	$\neg J_1(a \supset b) \vee b$

Complements

Name	**Formula c(a)**	**$RPL_\triangle$ equivalent**
Strong negation	1 for $[\![a]\!]=0$ 0 o.w.	$J_1 \neg a$
Weak negation	0 for $[\![a]\!]=1$ 1 o.w.	$\neg J_1 a$

Others

Name	**Formula o(a,b)**	**$RPL_\triangle$ equivalent**
Inhibition	1 for $[\![a]\!]>[\![b]\!]$ 0 o.w.	$\neg J_1(a \supset b)$
Strict equivalence	1 for $[\![a]\!]=[\![b]\!]$ 0 o.w.	$J_1(a \equiv b)$
Median	for $\lambda \in [0..1]$, $\max([\![a]\!],[\![b]\!])$ for $[\![a]\!],[\![b]\!] \in [0..\lambda]$ $\min([\![a]\!],[\![b]\!])$ for $[\![a]\!],[\![b]\!] \in [\lambda..1]$ λ o.w.	$[J_1(a \vee b \supset C_\lambda) \wedge (a \vee b)] \vee$ $[J_1(C_\lambda \supset a \wedge b) \wedge a \wedge b] \vee$ $[\neg J_1(a \vee b \supset C_\lambda) \wedge C_\lambda \wedge$ $\neg J_1(C_\lambda \supset a \wedge b)]$
λ-average	For i(a, b) a t-norm, u(a, b) a t-conorm, and $\lambda \in [0..1]$: $\min[\lambda,[\![u(a,b)]\!]]$ for $[\![a]\!],[\![b]\!] \in 0..\lambda]$ $\max[\lambda,[\![i(a,b)]\!]]$ for $[\![a]\!],[\![b]\!] \in [\lambda..1]$ λ o.w.	$[J_1(a \vee b \supset C_\lambda) \wedge C_\lambda \wedge u(a,b)] \vee$ $[J_1(C_\lambda \supset a \wedge b) \wedge (C_\lambda \vee i[a, b])] \vee$ $[\neg J_1(a \vee b \supset C_\lambda) \wedge \neg J_1(C_\lambda \supset a \wedge b) \wedge C_\lambda]$