Kleene's Three Valued Logics and Their Children

Melvin Fitting
Dept. Mathematics and Computer Science
Lehman College (CUNY), Bronx, NY 10468
Depts. Computer Science, Philosophy, Mathematics
Graduate Center (CUNY),
33 West 42nd Street, NYC, NY 10036
mlflc@cunyvm.cuny.edu *

Abstract. Kleene's strong three-valued logic extends naturally to a four-valued logic proposed by Belnap. We introduce a *guard connective* into Belnap's logic and consider a few of its properties. Then we show that by using it four-valued analogs of Kleene's weak three-valued logic, and the asymmetric logic of Lisp are also available. We propose an extension of these ideas to the family of *distributive bilattices*. Finally we show that for *bilinear* bilattices the extensions do not produce any new equivalences.

1 Introduction

Multiple-valued logics have been introduced for many reasons: purely mathematical, as with Post, or philosophical, as with Łukasiewicz. Few have been as useful or as natural as the three-valued logics of Kleene [12], introduced for computer science purposes — or at least they would have been if computer science had existed at the time. Kleene thought of the third truth value as *undefined* or *undetermined*, rather than as contingent or of probability 1/2. This reading suggests a natural condition: the behavior of the third truth value should be compatible with any increase in information. That is, if the value of some propositional letter, P say, is changed from undefined to either *true* or *false*, the value of any formula with P as a component should never change from *true* to *false* or from *false* to *true*, though a change from undefined to one of *false* or *true* is allowed. Kleene referred to this as *regularity*; today we phrase it in terms of *monotonicity* in an ordering that places undefined below both *false* and *true*. Among three-valued logics satisfying this regularity condition, Kleene observed, there is a weakest and a strongest, and there are several intermediate ones.

*Research partly supported by NSF Grant CCR-9104015.

Kleene's strong three-valued logic has been generalized in several ways, only some of which concern us here. We are not going to consider algebraic notions like Kleene algebras. Instead we want to consider particular structures motivated by computer science concerns. In [1] Kleene's logic was extended to a four-valued version, with intended database applications; then in [11] this logic was further generalized to a whole family of multiple-valued logics called *bilattices*, many of which contain natural analogs of Kleene's strong three-valued logic. Bilattices were created with AI applications in mind, but in [7, 9] bilattices were shown to have direct application to the semantics of logic programming. In [8] Kleene's logic was generalized directly, and it was shown that the resulting logics could be extended to bilattices, much as Kleene's logic had been extended to the four-valued version. It is to the family of bilattices that we will look for our generalizations. But all that we have said so far concerns Kleene's *strong* three-valued logic. What about other regular logics? In this paper we consider his weak version, and an intermediate one of natural interest, propose a reasonable extension of them to four-valued logic, and then to the family of distributive bilattices. We sketch a tableau system, but we do not give axiom systems. We believe these logics are natural objects to study, and the primary point of this paper is to encourage interest.

This paper began as a talk at the Bulgarian Kleene Conference in 1990. Subsequently a written paper circulated informally. The present paper is a revised and enlarged version of that.

2 Background

Most programming languages provide some version of the Boolean connectives, but since truth values of the constituents of a Boolean expression can be the results of procedure calls which might not terminate, the underlying logic is best thought of as three-valued. One way different programming languages can differ is in which three-valued logic is involved, but in general, regularity is a requirement.

In what follows we will use $\wedge$, $\vee$ and $\neg$, or some reasonable variant, to symbolize conjunction, disjunction and negation, and their extensions to many-valued logics as well. Likewise we will use $\perp$ for *undefined* (rather than **u** as Kleene did).

Most implementations of Pascal require that (classical) truth values of all components of a Boolean expression be available before the value of the expression itself is calculated. In particular, $P \wedge Q$ must be given the value $\perp$ if either P or Q has the value $\perp$ (that is, the procedure calculating P or the procedure calculating Q does not terminate). Otherwise $P \wedge Q$ behaves classically. Similarly for $\vee$ and $\neg$. This is the *weak* Kleene logic. In general we will write $P \wedge^w Q$ for a conjunction in this logic (and its generalizations), and $P \vee^w Q$ for a disjunction.

On the other hand, one can imagine a language allowing a degree of parallelism, in which the calculation of a value for $P \wedge Q$ proceeds as follows. Values for P and Q are calculated in parallel; if either P or Q turns out *false*, work on the other is halted and $P \wedge Q$ is assigned the value *false* on the grounds that the value of the other won't matter. If one of P or Q turns out *true* (say P), work must continue

on the other component (Q) because its value now is critical; the value of $P \wedge Q$ is whatever the value of Q turns out to be. In such a system $P \wedge Q$ is *true* if both components are *true*; $P \wedge Q$ is *false* if one component is *false*; $P \wedge Q$ is $\perp$ otherwise. This means $P \wedge Q$ is $\perp$ if one of P or Q is *true* but the other is $\perp$, or if both are $\perp$. Again similar considerations apply to $\vee$ and $\neg$. This is the *strong* Kleene logic. Every regular three-valued logic is intermediate between the weak and the strong one.

There could also be a sequential evaluation of $P \wedge Q$, say from left to right, so that P is evaluated first. If P evaluates to *false*, work stops and $P \wedge Q$ is assigned *false*. If P evaluates to *true* then Q is evaluated, and its value becomes the value assigned to $P \wedge Q$. This is an asymmetric logic. For instance, if P is *false* but Q is $\perp$, $P \wedge Q$ evaluates to *false*, but $Q \wedge P$ evaluates to $\perp$. This logic is also regular in Kleene's sense, and its connectives correspond to the AND, OR and NOT of Lisp. It is also the logic of Prolog as actually implemented, with its left-right, top-down evaluation. We will generally refer to it as *Lisp logic*, and we will write $P \vec{\wedge} Q$ for a conjunction in this logic (and its generalizations), and $P \vec{\vee} Q$ for a disjunction.

It is well-known that Kleene's strong three-valued logic finds a natural extension in Belnap's four-valued logic [1]. We will show that by adding a natural connective, which we call a *guard* connective, to Belnap's logic, Kleene's weak logic, and Lisp logic can also be extracted from Belnap's system. Then we will propose what we believe is a natural generalization of the guard connective, and consequently of Kleene's weak logic, and Lisp logic, to the family of distributive bilattices, introduced by Matt Ginsberg [11]. Along the way we give simple tableau rules and prove some related results.

3 The Belnap four-valued logic

A shift in emphasis from programming languages to data bases makes it natural to move from a three to a four-valued logic. A program might not come up with an answer to a yes/no question. A data base may do this as well, but in addition the data base could be inconsistent and so come up with both answers. We should allow $\top$, or *overdefined*, as well as $\perp$. Dunn introduced just such a four-valued logic in conjunction with his work on relevance logic [2, 3, 4]. Belnap pointed out the utility of this logic for computer science, and made the important observation that two partial orderings, not one, were involved [1]. Ginsberg understood and generalized the relationships between these orderings [11], producing bilattices. We will discuss the four-valued Belnap logic taking advantage of Ginsberg's insights. By now bilattices are known to a moderately wide group of people, but they are still not common items, so we discuss their basic properties, beginning with the special case of Belnap's logic.

Consider the double Hasse diagram of Figure 1, for a logic we will call $\mathcal{FOUR}$. There are two partial orderings displayed here. If a is to the left of b (or the same as b) symbolize this as $a \leq_t b$. Likewise if a is below b (or the same as b) write $a \leq_k b$. The $\leq_t$ ordering is, intuitively, one on degree of truth. Thus *false* $\leq_t \top$ since $\top$, which we can think of as both true and false, is 'truer' than *false*. Likewise $\top \leq_t$ *true* since *true* is of a lower degree of falseness than $\top$, which is false as well as true. The $\leq_k$

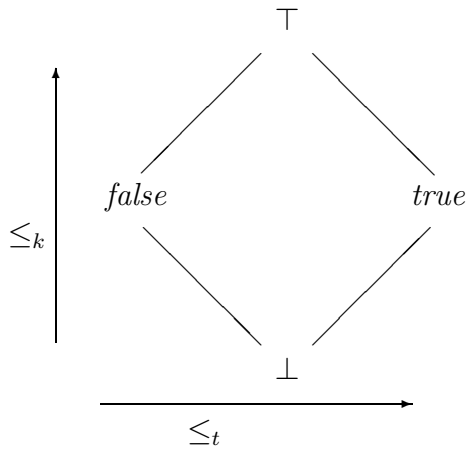

Figure 1: Belnap's Four-Valued Logic, $\mathcal{FOUR}$

ordering is on degree of information. Thus $\bot$, or no information, is at the bottom, and $\top$, or too much information, is at the top.

Both orderings yield complete lattices. We will use $\wedge$ and $\vee$ for meet and join under $\leq_t$, and $\otimes$ and $\oplus$ for meet and join under $\leq_k$. In fact $\wedge$ and $\vee$, confined to $\{\mathit{false}, \mathit{true}\}$, are the classical connectives, and on $\{\mathit{false}, \mathit{true}, \bot\}$ they are the connectives of the strong Kleene logic. The connectives $\otimes$ and $\oplus$ have no classical counterparts, but they are equally natural nonetheless. Think of $\otimes$ as a *consensus* operator: $P \otimes Q$ is the most information that P and Q agree on. Likewise $\oplus$ is an accept-anything, or *gullability* operator.

The two orderings are not independent of each other; rather they are strongly connected. They satisfy the *interlacing conditions*: the truth connectives, $\wedge$ and $\vee$, are monotonic with respect to the $\leq_k$ ordering, and the information connectives, $\otimes$ and $\oplus$, are monotonic with respect to the $\leq_k$ ordering. Monotonicity with respect to $\leq_k$ is what corresponds to Kleene's regularity condition, in Belnap's setting, so the interlacing conditions imply the regularity of $\wedge$ and $\vee$. Incidentally, $\wedge$ and $\vee$ are monotonic with respect to $\leq_t$ as well — this follows from the fact that $\leq_t$ yields the structure of a lattice. Similarly for $\otimes$, $\oplus$, and $\leq_k$, of course. In fact, something stronger than interlacing holds. Each of $\wedge$, $\vee$, $\otimes$, and $\oplus$ is distributive over the rest — 12 distributive laws in all. It is easy to check that interlacing follows from distributivity.

Apart from the lattice operations, there are two natural symmetries: left-right, and top-bottom. Let $\neg$ be the left-right symmetry: $\neg \mathit{true} = \mathit{false}$; $\neg \mathit{false} = \mathit{true}$; $\neg\top = \top$; and $\neg\bot = \bot$. On $\{\mathit{false}, \mathit{true}, \bot\}$ this is the negation of the Kleene strong (or weak) logic. Negation is monotonic with respect to $\leq_k$ (it is regular), but not with respect to $\leq_t$. Finally, let $-$ be the top-bottom symmetry: $-\top = \bot$; $-\bot = \top$; $-\mathit{false} = \mathit{false}$; and $-\mathit{true} = \mathit{true}$. We call this operator *conflation* [6]. It will find its motivation in Section 7.

The following observations are trivial, but will take on greater significance soon. In the first place, $\{\mathit{false}, \mathit{true}\} = \{x \mid x = -x\}$, and this is the sublogic known as classical. Second, $\{\mathit{false}, \mathit{true}, \bot\} = \{x \mid x \leq_k -x\}$, and these are the values of the strong Kleene logic.

4 Tableau Rules

In [3] Dunn introduced a tableau system, essentially for Belnap's logic, using a notion of coupled trees. Here we present a variant of that system.

Two versions of tableaus for classical logic were presented in [14], one using signed formulas, one not. We use the signed version here. A *signed formula* is of the form TX or FX, where X is a formula. Classically TX is read as "X is true", and FX as "X is false." A tableau is a tree with signed formulas as node labels. It is said to be *for* the signed formula at its root. It is constructed using certain branch extension rules, to be given in a moment. A tableau is a refutation argument; to prove X, one begins with FX and derives a contradiction. Intuitively this says X can not be false, and hence must always be true. The form a contradiction takes is syntactically simple. Call a branch *closed* if it contains TA and FA for some A, and call a tableau closed if every branch is; a closed tableau for FX constitutes a proof of X.

Probably most readers are familiar with the Smullyan branch extension rules. We give two as representative examples. If $TX \wedge Y$ occurs on a branch, both TX and TY may be added to the end (since classically $X \wedge Y$ is true if and only if both X and Y are true). Likewise if $FX \wedge Y$ occurs on a branch, the end of the branch may be split, with FX added to one fork and FY to the other (since $X \wedge Y$ is false if and only if either X is false or Y is false).

In order to move from two-valued classical logic to the four-valued Belnap logic, the most obvious thing we could do is introduce four signs, corresponding to the four truth values of $\mathcal{FOUR}$. For a variety of reasons we do not do this. For one thing, we have been thinking of $\bot$ as representing the lack of a (classical) value. It would be disquieting to introduce into our proof procedure a symbol which, in effect, would represent a value of no value. Analogous remarks apply to $\top$. Still, classical two-valued logic is, in a sense, behind $\mathcal{FOUR}$: the values of $\mathcal{FOUR}$ can be thought of as representing an answer of *true* only, or *false* only, or neither, or both. We want our tableau proof procedure to reflect this.

Suppose we are told by a generally reliable person that a proposition is *true*. We do not know, on the basis of this information alone, that the appropriate Belnap truth value to use is *true* since later another equally trustworthy person may tell us the proposition is *false*, and so the 'real' truth value is $\top$. In short, the best we can expect from a single piece of information is a lower bound on the possibilities: at least *true* (*true* or $\top$), rather than exactly *true*; at least *false* (*false* or $\top$) rather than exactly *false*.

In using classical tableaus one shows X is always *true* by assuming it could be *false*, that is by beginning with FX, and deriving a contradiction. Using $\mathcal{FOUR}$ we can show X is at least *true* by supposing it isn't, that it is either *false* or $\bot$, and

deriving a contradiction. This leads directly to a reading for the signs T and F. We will think of FX as saying X is either *false* or $\bot$. Then if we arrive at a contradiction by starting with FX, it follows that X must be either *true* or $\top$, that is, at least *true*. Similarly we will read TX as saying X is either *true* or $\bot$, so a contradiction deriving from TX will mean X is at least *false*. If contradictions follow from both TX and FX, it tells us X is $\top$, and if contradictions follow from neither, X is $\bot$. We should mention that Dunn in [3] followed a somewhat different convention, thinking of FX as saying X is at least *false*, that is, either *false* or $\top$, and similarly for TX. Thus some of our rules will look different than in Dunn's system, though the differences do not show up for the analogs of the classical connectives, only for $\otimes$ and $\oplus$. The systems are, in a natural sense, duals of each other.

It is simple to check that, in $\mathcal{FOUR}$, $X \wedge Y$ is either *true* or $\bot$ if and only if both X and Y are either *true* or $\bot$. Likewise $X \wedge Y$ is either *false* or $\bot$ if and only if one of X or Y is either *false* or $\bot$. In other words, if we use the four-valued interpretation of the signs in place of the two-valued reading, we still wind up with the same tableau rules for conjunction. In fact, every one of Smullyan's propositional branch extension rules continues to hold under the four-valued reading. We can also devise rules to cover $\oplus$ and $\otimes$ as well. It is easy to verify that $X \oplus Y$ is *true* or $\bot$ if and only if both X and Y are *true* or $\bot$. Similarly $X \oplus Y$ is *false* or $\bot$ if and only if both X and Y are *false* or $\bot$. That is, both $TX \oplus Y$ and $FX \oplus Y$ will have non-branching rules. Similarly both $TX \otimes Y$ and $FX \otimes Y$ will have branching rules.

In order to present his system elegantly Smullyan introduced *uniform notation*, which condensed several similar rules into one. Two categories of propositional signed formulas were defined, α signed formulas, which behave conjunctively, and β signed formulas, which behave disjunctively. For each α two *components*, α_1 and α_2 were defined: the idea is that an α signed formula is, in an obvious sense, equivalent to the conjunction of the corresponding α_1 and α_2. Similarly each β is equivalent to the disjunction of its components, β_1 and β_2. Smullyan's notation is easily extended to cover the operations $\oplus$ and $\otimes$. We present the result in Figure 2.

α	α_1	α_2
$TX \wedge Y$	TX	TY
$FX \vee Y$	FX	FY
$TX \oplus Y$	TX	TY
$FX \oplus Y$	FX	FY

β	β_1	β_2
$FX \wedge Y$	FX	FY
$TX \vee Y$	TX	TY
$TX \otimes Y$	TX	TY
$FX \otimes Y$	FX	FY

Figure 2: $\mathcal{FOUR}$ Conjunctive and Disjunctive Cases

Now the branch extension rules are easily given, in Figure 3 — in fact they have the same appearance as in Smullyan. If an α occurs on a branch, α_1 and α_2 can be added to the end. If a β occurs on a branch, the branch splits and β_1 is added to one side and β_2 to the other. The negation rules are straightforward.

We are not done, however: classical branch closure rules simply do not carry over. After all, TA and FA is no contradiction — if A has the value $\bot$, both TA and FA are

$$\frac{\alpha}{\alpha_1 \\ \alpha_2} \qquad \frac{\beta}{\beta_1 \mid \beta_2} \qquad \frac{T\,\neg Z}{F\,Z} \qquad \frac{F\,\neg Z}{T\,Z}$$

Figure 3: Branch Extension Rules

realized. The problem is essentially unsolvable: Belnap's logic, and Kleene's logics as well, simply have no tautologies! But this does not make the tableau system useless. On the one hand we could enlarge the language by adding propositional constants, *true* and *false*, along with the closure rules: a branch containing T *false* or F *true* is closed. This solves the problem by avoiding it.

It is also possible to use tableaus in a different way, allowing us to test formulas for *equivalence*. For instance, $X \wedge Y$ and $Y \wedge X$ always have the same truth value, no matter how the four values of Belnap's logic are assigned to X and Y. This notion of equivalence is not a definable connective in Belnap's system, but it is available 'from the outside,' so to speak. We symbolize it with the equality sign, and read $X = Y$ as 'X and Y are equivalent.'

Definition 4.1 A *valuation* is a mapping from propositional variables to $\mathcal{FOUR}$. It is extended inductively to all formulas in the obvious way.

Definition 4.2 We say X *restricts* Y if under any valuation in $\mathcal{FOUR}$, if X is at most *true* ($\bot$ or *true*) so is Y. We say X *requires* Y if under any valuation in $\mathcal{FOUR}$, if X is at least *true* (*true* or $\top$) so is Y.

These are not independent notions. It is easy to see that X restricts Y if and only if $\neg Y$ requires $\neg X$, for instance. Dunn showed that X requires Y if and only if X entails Y in relevance logic, though this plays no role here. Our interest in it is more elementary: if X and Y each restrict and require the other, they must have the same value in $\mathcal{FOUR}$ under any valuation; that is, they must be equivalent. A tableau test for this is simple.

Definition 4.3 Call a tableau *complete* if every non-atomic formula occurrence has had the appropriate rule applied to it, on each branch on which its node lies.

Let θ_1 and θ_2 be tableau branches. We say θ_1 *subsumes* θ_2 if every signed atomic formula on θ_2 also occurs on θ_1.

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be complete tableaus. We say $\mathcal{T}_1$ *covers* $\mathcal{T}_2$ if each branch of $\mathcal{T}_1$ subsumes some branch of $\mathcal{T}_2$.

The following relates the tableau notion of covering with the semantical notions of restriction and requirement.

Proposition 4.4 *Let X and Y be arbitrary formulas.*

1. *X restricts Y if and only if a complete tableau for TX covers a complete tableau for TY.*

2. *X requires Y if and only if a complete tableau for FY covers a complete tableau for FX.*

3. *X and Y are equivalent if and only if each restricts and requires the other.*

Proof We only sketch the underlying ideas. Suppose a complete tableau $\mathcal{T}_1$ for TX covers a complete tableau $\mathcal{T}_2$ for TY. (This is, in effect, the soundness direction.) Let us say a *signed* formula, TA, is true under a valuation if the valuation maps A to either *true* or $\bot$; likewise FA is true under a valuation if the valuation maps A to either *false* or $\bot$. It is easy to check that if a valuation makes all the signed formulas on some tableau branch true, and any branch extension rule is applied, the valuation will make all the signed formulas on some branch of the resulting tableau true as well. Now, suppose v is a valuation mapping X to either *true* or $\bot$. Since v makes TX true, in the tableau $\mathcal{T}_1$ all formulas on some branch are made true by v. Since $\mathcal{T}_1$ covers $\mathcal{T}_2$, there is a branch of $\mathcal{T}_2$ such that all signed *atomic* formulas on it are made true by v. Now an induction on complexity shows that *all* signed formulas on the branch are made true by v, in particular the top formula, TY. Thus X restricts Y.

In the other direction, suppose a complete tableau $\mathcal{T}_1$ for TX does not cover a complete tableau $\mathcal{T}_2$ for TY. Say branch θ of $\mathcal{T}_1$ does not subsume any branch of $\mathcal{T}_2$. Use θ to define a valuation v as follows. For an atom A: if both TA and FA occur on θ, set $v(A) = \bot$; if TA but not FA is on θ, set $v(A) = \mathit{true}$; if FA but not TA is on θ, set $v(A) = \mathit{false}$; finally if neither TA nor FA is on θ, set $v(A) = \top$. It can be checked that v will make every signed formula on θ true, in particular, TX. But also v does not make the set of signed atomic formulas on any branch of $\mathcal{T}_2$ true, and it follows by an induction on complexity that it does not make TY true. Consequently X does not restrict Y. ∎

An example of a tableau construction will be found in the next Section, after additional four-valued connectives have been introduced.

5 The Guard Connective

When considering a logic programming language based on $\mathcal{FOUR}$ [6] the usefulness of a particular connective, which we call a *guard* gradually emerged. We write $P : Q$, and read it as "P guards Q." Loosely the idea is, if the guard is passed, then we evaluate Q to get the result; if the guard can't be passed, we get no information. More formally, if P has the value *true* or the value $\top$ (i.e. if P is at least *true*) then $P : Q$ has the value of Q; otherwise $P : Q$ has the value $\bot$. (We do not give a truth table for this, since it is quite elementary.) The guard connective, though not lattice theoretic, has several interesting properties. It is monotonic with respect to $\leq_k$, in both inputs, so it is regular. With respect to $\leq_t$ it is monotonic only in the second input, though we also have the following curious property:

$$P_1 \leq_t P_2 \Rightarrow (P_1 : Q) \leq_k (P_2 : Q).$$

In addition the identities in Figure 4 hold. These identities are strong enough to imply the monotonicity results cited above. For instance, suppose $P_1 \leq_t P_2$. Then $P_1 = P_1 \wedge P_2$ so: $P_1 : Q = (P_1 \wedge P_2) : Q = (P_1 : Q) \otimes (P_2 : Q) \leq_k P_2 : Q$.

$$\begin{array}{rcl}
(P \otimes Q) : R & = & (P \wedge Q) : R \\
 & = & (P : R) \otimes (Q : R) \\
 & = & (P : Q) : R \\
 & = & P : (Q : R) \\
(P \oplus Q) : R & = & (P \vee Q) : R \\
 & = & (P : R) \oplus (Q : R) \\
P : \neg Q & = & \neg (P : Q) \\
P : (Q \wedge R) & = & (P : Q) \wedge (P : R) \\
P : (Q \vee R) & = & (P : Q) \vee (P : R) \\
P : (Q \otimes R) & = & (P : Q) \otimes (P : R) \\
P : (Q \oplus R) & = & (P : Q) \oplus (P : R) \\
P \oplus (Q : P) & = & P \\
P \otimes (Q : P) & = & (Q : P)
\end{array}$$

Figure 4: Guard Connective Properties

Tableau rules for the guard connective are fairly simple. It turns out that both T and the F signed formulas act disjunctively, so we only need to add two more lines to the β table. This is done in Figure 5.

β	β_1	β_2
$FX : Y$	FX	FY
$TX : Y$	FX	TY

Figure 5: The Guard Connective Cases

As noted earlier, the Belnap logic contains the strong Kleene logic as a sublogic. Using the guard connective the asymmetric Lisp logic connectives have a simple characterization: $P \ \vec{\wedge} \ Q = P \wedge (P : Q)$ and $P \ \vec{\vee} \ Q = P \vee (\neg P : Q)$. These identities are correct when values are restricted to $\{\mathit{false}, \mathit{true}, \bot\}$; the restriction is necessary since the Lisp connectives are only defined for these values. The meaning is reasonably intuitive: consider conjunction as an example — we do not consider the second component unless we pass the guard of the first component. In addition the following equations characterize the weak Kleene connectives: $P \wedge^w Q = (P \ \vec{\wedge} \ Q) \otimes (Q \ \vec{\wedge} \ P)$ and $P \vee^w Q = (P \ \vec{\vee} \ Q) \otimes (Q \ \vec{\vee} \ P)$. Thus the weak connectives can be seen as a consensus of sequential evaluations, left-right and right-left, accepting only what the two directions agree on. It is interesting to note that the conjunction and disjunction of the strong Kleene logic can be recovered from the asymmetric version as well, using the gullability operator: $P \wedge Q = (P \ \vec{\wedge} \ Q) \oplus (Q \ \vec{\wedge} \ P)$ and $P \vee Q = (P \ \vec{\vee} \ Q) \oplus (Q \ \vec{\vee} \ P)$. The equations characterizing $\vec{\wedge}$ and $\vec{\vee}$ in terms of $\wedge$ and $\vee$, given above, are correct

only for consistent truth values, *true*, *false*, and $\bot$. This is trivially so, since these are all that $\vec{\wedge}$ and $\vec{\vee}$ are defined for. We propose reversing things, and using the equations to define natural extensions of the three-valued connectives to all of $\mathcal{FOUR}$.

Definition 5.1 The following defines connectives in $\mathcal{FOUR}$.

$$\begin{array}{rcl} P \,\vec{\wedge}\, Q & = & P \wedge (P : Q) \\ P \,\vec{\vee}\, Q & = & P \vee (\neg P : Q) \\ P \wedge^w Q & = & (P \,\vec{\wedge}\, Q) \otimes (Q \,\vec{\wedge}\, P) \\ P \vee^w Q & = & (P \,\vec{\vee}\, Q) \otimes (Q \,\vec{\vee}\, P) \end{array}$$

For convenience we summarize the behavior of $\vec{\wedge}$ and $\vec{\vee}$ in the truth tables of in Figure 6. Incidentally, the equations recovering the strong Kleene connectives from the asymmetric ones continue to work, using the extension to Belnap's four truth values. Certain properties are lost in making the extension, however. The distributivity law, $P \,\vec{\wedge}\, (Q \,\vec{\vee}\, R) = (P \,\vec{\wedge}\, Q) \,\vec{\vee}\, (P \,\vec{\wedge}\, R)$, is an identity when three truth values are used, but it does not extend to four: taking $P = R = \top$ and $Q = \bot$ falsifies it.

$\vec{\wedge}$	$\top$	*true*	*false*	$\bot$
$\top$	$\top$	$\top$	*false*	*false*
true	$\top$	*true*	*false*	$\bot$
false	*false*	*false*	*false*	*false*
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

$\vec{\vee}$	$\top$	*true*	*false*	$\bot$
$\top$	$\top$	*true*	$\top$	*true*
true	*true*	*true*	*true*	*true*
false	$\top$	*true*	*false*	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

Figure 6: Four-valued Tables for $\vec{\wedge}$ and $\vec{\vee}$

Since there are tableau rules for the guard connective, these yield tableau rules for the Lisp connectives and the weak connectives. These do not fall into the α, β scheme — we give the derived Lisp connective rules in Figure 7.

$$\frac{F\,P \,\vec{\wedge}\, Q}{F\,P \mid F\,Q} \qquad \frac{T\,P \,\vec{\vee}\, Q}{T\,P \mid T\,Q} \qquad \frac{T\,P \,\vec{\wedge}\, Q}{\begin{array}{l} T\,P \\ F\,P \mid T\,Q \end{array}} \qquad \frac{F\,P \,\vec{\vee}\, Q}{\begin{array}{l} F\,P \\ T\,P \mid F\,Q \end{array}}$$

Figure 7: Lisp Connectives Rules

We noted earlier that, although $P \,\vec{\wedge}\, (Q \,\vec{\vee}\, R)$ and $(P \,\vec{\wedge}\, Q) \,\vec{\vee}\, (P \,\vec{\wedge}\, R)$ are equivalent when restricted to Kleene's three values, they are not when $\top$ is allowed. Figure 8 gives complete tableaus for $F\,P \,\vec{\wedge}\, (Q \,\vec{\vee}\, R)$ and $F\,(P \,\vec{\wedge}\, Q) \,\vec{\vee}\, (P \,\vec{\wedge}\, R)$. In this Figure, the second tableau covers the first, but the first does not cover the second. The middle branch of the first tableau does not subsume any branch of the

second. Following the proof of Proposition 4.4 a valuation mapping P and R to $\top$ and Q to $\bot$ shows that $(P \vec{\wedge} Q) \vec{\vee} (P \vec{\wedge} R)$ does not require $P \vec{\wedge} (Q \vec{\vee} R)$. In fact this is the valuation we used above to show the distributive law does not hold in $\mathcal{FOUR}$. Incidentally, tableau rules for the 'Kleene part' of $\mathcal{FOUR}$ can be based on the fact that the tableau completeness proof essentially tells how to construct all counterexamples. If the only counterexamples involve $\top$, the equivalence in question is valid in Kleene's logic.

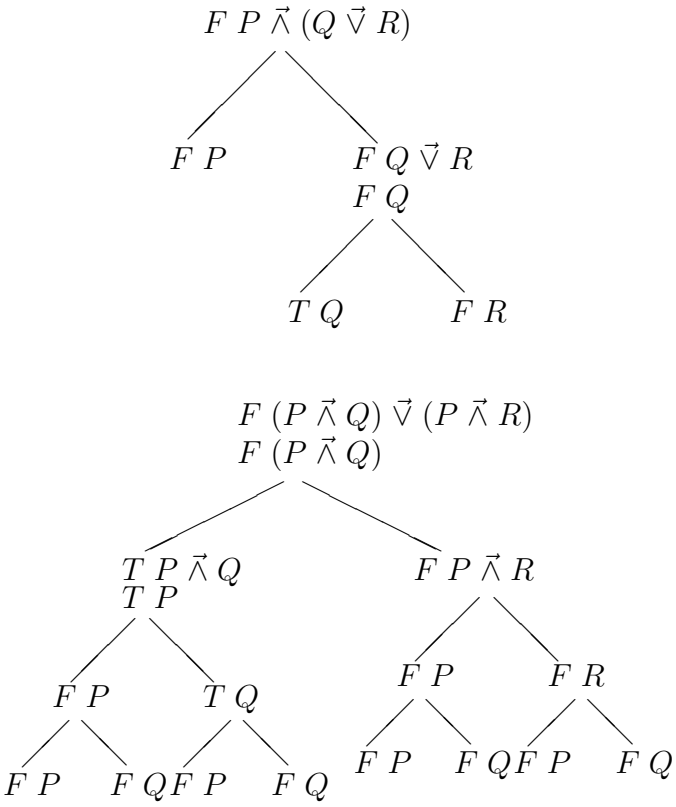

Figure 8: Tableaus $\mathcal{T}_1$ and $\mathcal{T}_2$

6 Bilattices

The logic $\mathcal{FOUR}$ has found a natural generalization in Ginsberg's notion of *bilattice* [11]. Actually, there are several kinds of bilattices — this is not the appropriate place to go into details of the whole family. Instead we will concentrate on the kind that most directly generalizes Belnap's logic: distributive bilattices.

Definition 6.1 A *pre-bilattice* is a structure $\langle \mathcal{B}, \leq_t, \leq_k \rangle$ where each of $\leq_t$ and $\leq_k$ is a partial ordering on $\mathcal{B}$ that gives it the structure of a lattice with a top and bottom.

If $\langle \mathcal{B}, \leq_t, \leq_k \rangle$ is a pre-bilattice then meets and joins exist for both orderings. We will use $\wedge$ and $\vee$ for meet and join under $\leq_t$, and $\otimes$ and $\oplus$ for meet and join under $\leq_k$. Also least and greatest elements must exist under each ordering. We will use *false* and *true* for the least and greatest element under $\leq_t$, and $\bot$ and $\top$ for the least and greatest element under $\leq_k$. Also we will assume all pre-bilattices are non-trivial: *true*, *false*, $\bot$, and $\top$ are all distinct.

Definition 6.2 Let $\langle \mathcal{B}, \leq_t, \leq_k \rangle$ be a pre-bilattice. It has a *negation* if there is a mapping $\neg$ from $\mathcal{B}$ to itself such that:

1. $x \leq_t y \Longrightarrow \neg y \leq_t \neg x$;
2. $x \leq_k y \Longrightarrow \neg x \leq_k \neg y$;
3. $\neg\neg x = x$.

The pre-bilattice has a *conflation* if there is a mapping $-$ from $\mathcal{B}$ to itself such that:

1. $x \leq_k y \Longrightarrow -y \leq_k -x$;
2. $x \leq_t y \Longrightarrow -x \leq_t -y$;
3. $--x = x$.

Ginsberg's definition of bilattice is essentially: a pre-bilattice with negation. (Actually, he also requires each lattice to be complete, something we omit.) Notice that if there is a negation, the de Morgan laws will hold with respect to $\neg$ and $\wedge$ and $\vee$, while $\otimes$ and $\oplus$ will be their own duals under $\neg$. If there is a conflation, the situation is exactly reversed: $\otimes$ and $\oplus$ are interchanged by $-$ while $\wedge$ and $\vee$ are left alone. If there is a negation it reverses *true* and *false*, and leaves $\top$ and $\bot$ alone. Likewise if there is a conflation, it reverses $\top$ and $\bot$, while not affecting *true* and *false*.

Definition 6.3 A pre-bilattice $\langle \mathcal{B}, \leq_t, \leq_k \rangle$ is a *distributive bilattice* if each of $\wedge$, $\vee$, $\otimes$ and $\oplus$ distributes over the other three operations.

Then $\mathcal{FOUR}$ is a distributive bilattice with a negation and a conflation that commute with each other. It is one among many, in fact. We will give other examples in the next Section, but first some general remarks.

A bilattice is *interlaced* if each of the four operations, $\wedge$, $\vee$, $\otimes$ and $\oplus$ is monotonic in both orderings, $\leq_t$ and $\leq_k$. Interlacing played a fundamental role in [5] and [7]. Every distributive bilattice is interlaced. As we remarked earlier, interlacing is the direct generalization of Kleene's notion of regularity. In a distributive bilattice, or even in an interlaced one, each extremal element is a simple combination of other extremal elements. We always have $\mathit{true} \oplus \mathit{false} = \top$, $\mathit{true} \otimes \mathit{false} = \bot$, $\top \wedge \bot = \mathit{false}$ and $\top \vee \bot = \mathit{true}$. In the Belnap logic all elements are extremal. Other more complicated examples are given in the next section. Finally, it is easy to check that if $\mathcal{B}$ is a bilattice and S is a set, the function space $\mathcal{B}^S$, with the induced pointwise orderings, is a bilattice again. And it will have a negation if $\mathcal{B}$ has one; similarly for conflation.

7 Bilattice constructions

There are two simple ways of producing distributive bilattices. One directly generalizes the Kleene strong three-valued logic, then completes the result to a bilattice; details can be found in [8]. The other generalizes Belnap's logic directly; it is due to Ginsberg, though its origins can be traced back to Dunn. It is this version we sketch now.

Definition 7.1 Let $\langle L_1, \leq_1 \rangle$ and $\langle L_2, \leq_2 \rangle$ be lattices, each with a bottom and top. By $L_1 \odot L_2$ we mean $\langle L_1 \times L_2, \leq_t, \leq_k \rangle$ where:

1. $\langle x_1, x_2 \rangle \leq_t \langle y_1, y_2 \rangle$ if $x_1 \leq_1 y_1$ and $y_2 \leq_2 x_2$;
2. $\langle x_1, x_2 \rangle \leq_k \langle y_1, y_2 \rangle$ if $x_1 \leq_1 y_1$ and $x_2 \leq_2 y_2$

It is quite straightforward to check that $L_1 \odot L_2$ will always be an interlaced bilattice, and if L_1 and L_2 are each distributive, $L_1 \odot L_2$ will be a distributive bilattice. Thus there are many bilattice examples indeed. It follows from Definition 7.1 that the bilattice connectives have easy characterizations — we give them in Figure 9.

$$\begin{array}{rcl} \langle x_1, x_2 \rangle \wedge \langle y_1, y_2 \rangle & = & \langle x_1 \wedge y_1, x_2 \vee y_2 \rangle \\ \langle x_1, x_2 \rangle \vee \langle y_1, y_2 \rangle & = & \langle x_1 \vee y_1, x_2 \wedge y_2 \rangle \\ \langle x_1, x_2 \rangle \otimes \langle y_1, y_2 \rangle & = & \langle x_1 \wedge y_1, x_2 \wedge y_2 \rangle \\ \langle x_1, x_2 \rangle \oplus \langle y_1, y_2 \rangle & = & \langle x_1 \vee y_1, x_2 \vee y_2 \rangle \end{array}$$

Figure 9: Operations in $L_1 \odot L_2$

The rather technical construction above turns out to have a natural motivation. Suppose we are collecting evidence about propositions. Perhaps this evidence is statistical in nature, or perhaps it involves derivations from information in a database. At any rate we can, and often do, come up with reasons *for* something that are quite independent of our reasons *against* it. Now, suppose the lattice L_1 is used to record our belief in, evidence for, proofs of some proposition, while L_2 is used to record our disbelief in, evidence against, disproofs of that proposition. Then a pair, $\langle x, y \rangle$, with $x \in L_1$ and $y \in L_2$, serves as a summary of a situation. Notice that L_1 and L_2 need not be the same. We might ask Expert One whether P should be accepted, and that expert might reply with a probability, and we might ask Expert Two whether P should be rejected, and this reply might be either a 'yes' or silence. Whether the lattices are the same or different, the role of the two bilattice orderings of $L_1 \odot L_2$ should now be plausible: there is an increase in the 'truth' ordering if evidence for goes up, but evidence against goes down. There is an increase in the 'information' direction if all evidence goes up, both for and against. Notice that *true* amounts to maximum possible evidence for, but none against, while $\top$, or overdefined, amounts to maximum possible evidence both for and against. Of course, depending on the lattices L_1 and L_2, various degrees of overdefinedness are possible.

There are several specific examples of considerable interest. If $L_1 = L_2$ is the two element lattice $\{0, 1\}$, where $0 < 1$, then $L_1 \odot L_2$ is an isomorphic copy of

$\mathcal{FOUR}$. Taking L_1 and L_2 to be the unit interval $[0, 1]$ yields a natural probability bilattice. Taking L_1 to be the unit interval, and L_2 to be the two element lattice gives us a bilattice appropriate to an example earlier, where the 'for' expert spoke in probabilities, while the 'against' expert would say nothing more complicated than "yes."

If $L_1 = L_2$ then a natural notion of negation is available in $L_1 \odot L_2$: switch around the roles of belief and doubt. That is, set

$$\neg\langle x, y\rangle = \langle y, x\rangle.$$

It is easy to check that this meets the conditions of Definition 6.2.

Even if $L_1 = L_2$ there may not be a conflation operation. But suppose that, in addition, the underlying lattice is a *de Morgan lattice*: a distributive lattice with a de Morgan complement — a map from the lattice to itself that reverses the order relation and is an involution. Under these circumstances a natural conflation is available. Let us write x^- for the de Morgan complement of x. Then set

$$-\langle x, y\rangle = \langle y^-, x^-\rangle.$$

It is straightforward to check that the conditions of Definition 6.2 are met.

The intuition behind the conflation construction is quite direct. Suppose, for instance, that $L_1 = L_2$ is the unit interval. We can think of a member of the bilattice $L_1 \odot L_2$ as an encoding of a degree of belief and a degree of doubt in some proposition. The map that sends x to $1 - x$ is a de Morgan complement on the unit interval. Then the bilattice map that sends $\langle x, y\rangle$ to $\langle 1 - y, 1 - x\rangle$ is a conflation operation. It amounts to moving from a situation to a *conflated* version of the same situation in which we believe to the extent that we did not doubt before, and we doubt to the extent that we did not believe before. As another example, suppose S is a set, which we can think of as consisting of potential items of evidence. Consider the lattice $\mathcal{P}(\mathcal{S})$ whose members are subsets of S, ordered by $\subseteq$. Then a member of the bilattice $\mathcal{P}(\mathcal{S}) \odot \mathcal{P}(\mathcal{S})$ summarizes evidence for, and evidence against some proposition. The map that sends a set X to its complement relative to S is a de Morgan complement. Then the map that sends $\langle X, Y\rangle$ to $\langle S - Y, S - X\rangle$ is a bilattice conflation. In the conflated version of $\langle X, Y\rangle$ we are counting as evidence in favor anything that did not count against before, and as evidence against anything that did not count in favor.

One final remark about the constructions above. If L has a de Morgan complement, $L \odot L$ will have both a conflation and a negation operation. It is easy to check that these will commute with each other.

8 Representation Theorems

In the preceding section we saw that if L_1 and L_2 are distributive lattices, $L_1 \odot L_2$ is a distributive bilattice. In fact, the converse of this is true as well.

Theorem 8.1 *If $\mathcal{B}$ is a distributive bilattice then there are distributive lattices L_1 and L_2 such that $\mathcal{B}$ is isomorphic to $L_1 \odot L_2$.*

This result is due to Ginsberg, [11]. There is a more direct proof of it in [7] as well. The basic idea of the proof is as follows. Take for L_1 the set $\{x \vee \bot \mid x \in \mathcal{B}\}$ with an ordering $\leq_1$ where $x \leq_1 y \Leftrightarrow x \leq_t y$. Take $L_2 = \{x \wedge \bot \mid x \in \mathcal{B}\}$, and set $x \leq_2 y \Leftrightarrow y \leq_t x$. Then the mapping $\theta : \mathcal{B} \to L_1 \odot L_2$ given by $x\theta = \langle x \vee \bot, x \wedge \bot\rangle$ is a bilattice isomorphism. Incidentally, θ^{-1} is given by $\theta^{-1}(\langle a, b\rangle) = a \oplus b$. We omit details, which can be found in the papers cited above.

The Representation Theorem 8.1 can be strengthened to include negation and conflation. The negation result appears in [11] (as a consequence of Theorem 9.21). We give a proof that continues along the lines of the previous theorem.

Theorem 8.2 *Suppose $\mathcal{B}$ is a distributive bilattice with negation. Then there is a distributive lattice L such that $\mathcal{B}$ is isomorphic to $L \odot L$ under an isomorphism that preserves negation.*

Proof It is technically more convenient to show $\mathcal{B}$ is isomorphic to $L_1 \odot L_2$ where, in turn, L_1 and L_2 are isomorphic lattices.

Let L_1 and L_2 be as in the proof sketch for Theorem 8.1 above. Note that if $a \in L_1$, $a = x \vee \bot$, so $\neg a = \neg(x \vee \bot) = \neg x \wedge \bot \in L_2$. Then the map $\eta : L_1 \to L_2$, given by $a\eta = \neg a$, is well-defined. We claim it is an isomorphism.

First, $(a\eta)\eta = a$, so the map is $1-1$. And if $b \in L_2$, $b = x \wedge \bot$ for some x, so $\neg b = \neg(x \wedge \bot) = \neg x \vee \bot \in L_1$. Now $(\neg b)\eta = \neg\neg b = b$, so η is onto. Finally, suppose $a, b \in L_1$. Then $a \leq_1 b \Leftrightarrow a \leq_t b \Leftrightarrow \neg b \leq_t \neg a \Leftrightarrow b\eta \leq_t a\eta \Leftrightarrow a\eta \leq_2 b\eta$. Thus η is an isomorphism.

Using η, $L_1 \odot L_2$ has a negation, $\neg\langle a, b\rangle = \langle b\eta, a\eta\rangle$. Now, using the bilattice isomorphism θ from the proof of Theorem 8.1, $\neg(x\theta) = \neg\langle x \vee \bot, x \wedge \bot\rangle = \langle (x \wedge \bot)\eta, (x \vee \bot)\eta\rangle = \langle \neg x \vee \bot, \neg x \wedge \bot\rangle = (\neg x)\theta$. Thus θ preserves negation. ∎

Finally we extend the representation results to include conflation.

Theorem 8.3 *Suppose $\mathcal{B}$ is a distributive bilattice with negation and conflation operations that commute with each other. Then there is a distributive lattice L with a de Morgan complement, such that $\mathcal{B}$ is isomorphic to $L \odot L$ under an isomorphism that preserves negation and conflation.*

Proof We continue the proof of Theorem 8.2, using the same notation. In particular, instead of working with $L \odot L$ we work with $L_1 \odot L_2$, where L_1 and L_2 are isomorphic via the mapping η. Note that if $a \in L_1$, then $a = x \vee \bot$ and so $a \vee \bot = x \vee \bot \vee \bot = x \vee \bot = a$. Now let $i : L_1 \to L_1$ be given by $ai = -\neg a \vee \bot$. We claim i is a de Morgan complement.

First, for $a \in L_1$, $(ai)i = -\neg(-\neg a \vee \bot) \vee \bot = (a \wedge \top) \vee \bot = (a \vee \bot) \wedge (\top \vee \bot) = (a \vee \bot) \wedge \mathit{true} = a \vee \bot = a$. Thus i is an involution.

Next, suppose $a, b \in L_1$. Then $a \leq_1 b \Leftrightarrow a \leq_t b \Leftrightarrow \neg b \leq_t \neg a \Rightarrow -\neg b \vee \bot \leq_t -\neg a \vee \bot \Leftrightarrow bi \leq_t ai \Leftrightarrow bi \leq_1 ai$. (The converse implication follows from the fact that i is an involution.) Thus i is order reversing.

Finally, $L_1 \odot L_2$ has a convolution: $-\langle a, b\rangle = \langle (b\eta)i, (ai)\eta\rangle$. We leave it to you to verify that $-(x\theta) = (-x)\theta$. ∎

Thus the methods introduced in the previous section for constructing distributive bilattices are entirely general.

9 Kleene's logics, generalized

So far we have talked about distributive bilattices, which directly generalize the Belnap four-valued logic. But we began with Kleene's three-valued logics — to what extent are there natural generalizations of them? Such generalizations have been proposed before, of course. We are particularly interested in those related to bilattices. One possibility can be found in [8], where Kleene's strong three-valued logic is shown to be one of a family of logics, and each member of this family is part of a bilattice. But we also want to generalize the Kleene weak three-valued logic, and Lisp logic as well. We begin with the strong three-valued logic and the guard connective, since the other connectives can be characterized using this machinery.

In $\mathcal{FOUR}$ both the classical and Kleene truth values can be specified in terms of conflation. The classical truth values are those left unchanged by conflation and the Kleene values are those that contain no more information than their conflations. Here is something on which to base a generalization. The following is taken from [5, 7].

Definition 9.1 Suppose $\mathcal{B}$ is a bilattice with conflation. A member x of $\mathcal{B}$ is *exact* if $x = -x$, and is *consistent* if $x \leq_k -x$.

Incidentally, this notion of consistent is different from one of the same name in [11]. Now the following is straightforward to establish.

Theorem 9.2 *In any distributive (or just interlaced) bilattice with conflation the exact truth values contain true and false and are closed under $\wedge$ and $\vee$. If there is a negation that commutes with conflation, they are closed under negation as well. The exact truth values do not contain $\top$ or $\bot$ and are not closed under $\otimes$ or $\oplus$.*

Thus the exact values of $\mathcal{B}$ are a candidate for a generalization of classical two-valued logic. Of more interest for this paper is the following.

Theorem 9.3 *In any distributive (or again, just interlaced) bilattice with conflation the consistent truth values include the exact ones, $\bot$, and are closed under $\wedge$ and $\vee$. If there is a negation that commutes with conflation, they are closed under negation as well. Further, the collection of consistent members is closed under $\otimes$; and under $\oplus$ when applied to members having a common upper bound.*

Most of this is straightforward. If the bilattice is complete (that is, if each of $\leq_k$ and $\leq_t$ is a complete lattice ordering), the results extend to infinitary operations as well, with closure of consistent members under the infinitary version of $\oplus$ for directed sets. The only tricky part is the last, closure under directed sups. A proof of this, due to Visser, [15], can be found in the bilattice setting in [5, 7].

The closure results are curiously important. Kripke showed that a treatment of truth in languages allowing self-reference could be based on Kleene's strong three-valued logic [13]. It was a fixpoint approach, using a 'truth revision' operator. The existence of a smallest fixpoint depended on the closure of the Kleene logic under arbitrary infs (and by extension, the similar closure of the family of interpretations in this logic). Further, a special kind of fixpoint called *intrinsic* played an important role. Although there is no largest fixpoint, there is a largest intrinsic fixpoint. To establish its existence, closure of the Kleene logic (and of the family of interpretations) under directed sups is needed. It was showed in [5] that the Kripke approach extended to arbitrary interlaced bilattices, and the features of the Kleene logic that were significant in Kripke's approach carried over generally to collections of consistent truth values in bilattices, with similar results obtainable. Thus, at least from the point of view of Kripke's theory, the consistent truth values in a bilattice yield a reasonable generalization of the Kleene strong three-valued logic. In addition, in [10] the notion of intrinsic was given a role in the area of stable model semantics for logic programming as well, and again the consistent part of a bilattice plays a significant role.

In what follows, let $\mathcal{B}$ be a distributive bilattice with a negation and a conflation that commute. Our immediate task is to propose a generalization of the guard connective to $\mathcal{B}$, since $\mathcal{B}$ is a direct generalization of $\mathcal{FOUR}$ and consequently contains a generalization of Kleene's strong logic. Of course in principle a generalization proposal can be rather arbitrary — we will argue that our proposal is a natural one since it carries over to the whole family what seem like the essential features of $\mathcal{FOUR}$. Since every distributive bilattice can be obtained as the $\odot$-product of distributive lattices, it is enough to give a definition for such bilattices.

Definition 9.4 Let $L \odot L$ be (isomorphic to) the bilattice $\mathcal{B}$. Set $\langle a, b\rangle : \langle c, d\rangle$ to be $\langle a \wedge c, a \wedge d\rangle$, where these meets are calculated in L.

Thus in $\langle a, b\rangle : \langle c, d\rangle$ belief and doubt of $\langle c, d\rangle$ is modified by the degree to which the guard is believed, an intuitively plausible notion to consider. Using the Representation Theorems of Section 8, $\mathcal{FOUR}$ is isomorphic to $\mathcal{TWO} \odot \mathcal{TWO}$, where $\mathcal{TWO}$ is the lattice $\{\mathit{false}, \mathit{true}\}$, with $\mathit{false} < \mathit{true}$. Applying Definition 9.4 to an example from $\mathcal{FOUR}$ we have the following.

$$\begin{aligned} \mathit{false} : P &= \langle \mathit{false}, \mathit{true}\rangle : \langle p_1, p_2\rangle \\ &= \langle \mathit{false} \wedge p_1, \mathit{false} \wedge p_2\rangle \\ &= \langle \mathit{false}, \mathit{false}\rangle \\ &= \bot \end{aligned}$$

In fact, Definition 9.4 yields the same results generally in $\mathcal{FOUR}$ as our earlier characterization. In addition, all of the identities of Figure 4 continue to hold for the generalized version of the guard connective as well. Just as it did with $\mathcal{FOUR}$, it follows from some of these identities that the guard connective is monotonic with respect to $\leq_k$, and is monotonic with respect to $\leq_t$ in its second input.

There are many other ways of characterizing the generalized guard connective. The one given above is, we feel, the most intuitive and useful. The following characterization is easily verified to be equivalent:

$$P : Q = [(P \otimes \mathit{true}) \oplus \neg(P \otimes \mathit{true})] \otimes Q$$

Suppose Q is a consistent member of $\mathcal{B}$, that is, $Q \leq_k -Q$. Using the characterization above, we have the following.

$$\begin{array}{rcl} P : Q & = & [(P \otimes \mathit{true}) \oplus \neg(P \otimes \mathit{true})] \otimes Q \\ & \leq_k & Q \\ & \leq_k & -Q \\ & \leq_k & -[(P \otimes \mathit{true}) \oplus \neg(P \otimes \mathit{true})] \oplus -Q \\ & = & -(P : Q) \end{array}$$

Thus if Q is consistent in $\mathcal{B}$, so is $P : Q$. Now the earlier characterizations, from Belnap's logic, can be proposed as candidates for generalization from $\mathcal{FOUR}$ to $\mathcal{B}$.

$$\begin{array}{rcl} P \,\vec{\wedge}\, Q & = & P \wedge (P : Q) \\ P \,\vec{\vee}\, Q & = & P \vee (\neg P : Q) \\ P \wedge^w Q & = & (P \,\vec{\wedge}\, Q) \otimes (Q \,\vec{\wedge}\, P) \\ P \vee^w Q & = & (P \,\vec{\vee}\, Q) \otimes (Q \,\vec{\vee}\, P) \end{array}$$

The family of consistent members must be closed under these connectives. It follows from the monotonicity properties of the guard connective and the interlacing conditions that $\vec{\wedge}$, $\vec{\vee}$, $\wedge^w$, and $\vee^w$ are all monotonic with respect to $\leq_k$. As we remarked earlier, in $\mathcal{FOUR}$ this monotonicity was what made Kripke's truth revision operator monotonic, and lay at the heart of his semantic treatment of truth in languages allowing self-reference.

10 Bilinear bilattices

Among the family of distributive bilattices there is a subcollection of independent interest that behaves just like $\mathcal{FOUR}$ with respect to the connectives we have considered.

Definition 10.1 A bilattice is *bilinear* if any pair of members is comparable, under at least one of the bilattice orderings.

Proposition 10.2 *Let $\mathcal{B}$ be a distributive bilattice, and hence isomorphic to $L_1 \odot L_2$ for some distributive lattices L_1 and L_2. $\mathcal{B}$ is bilinear if and only if each of L_1 and L_2 is linear.*

The proof of this is straightforward, and is omitted.

We claim that, if $\mathcal{B}$ is a bilinear distributive bilattice with negation then an equivalence, $X = Y$, is valid in $\mathcal{B}$ if and only if it is valid in $\mathcal{FOUR}$. (Validity means both X and Y have the same values in $\mathcal{B}$ under every valuation, where the notion of valuation is the obvious one.) The chief tool we need is the following Definition and Lemma, in which we identify $\mathcal{FOUR}$ with $\mathcal{TWO} \odot \mathcal{TWO}$.

Definition 10.3 Let L be a lattice. We first define an operation from L to $\{false, true\}$ as follows:

$$x \to y = \begin{cases} true & \text{if } x \leq y \\ false & \text{otherwise} \end{cases}$$

Next, let $a \in L$. We define a mapping $\theta_a : L \odot L \to \mathcal{FOUR}$ as follows:

$$\theta_a(\langle x_1, x_2 \rangle) = \langle a \to x_1, a \to x_2 \rangle$$

Lemma 10.4 *Let L be a lattice that is linearly ordered, and let a be an arbitrary member of L. Then θ_a is a homomorphism in the following sense. For any $p, q \in L \odot L$:*

1. $\theta_a(\neg p) = \neg\theta_a(p)$;
2. *if $\circ$ is any of $\wedge$, $\vee$, $\otimes$, $\oplus$, $:$, $\vec{\wedge}$, $\vec{\vee}$, $\wedge^w$, or $\vee^w$, $\theta_a(p \circ q) = \theta_a(p) \circ \theta_a(q)$.*

Proof Since L is a lattice, in it $a \leq x \wedge y$ if and only if $a \leq x$ and $a \leq y$, and consequently $a \to (x \wedge y) = (a \to x) \wedge (a \to y)$, where the $\wedge$ on the right is the usual Boolean operation of $\{false, true\}$. Also $a \leq x$ or $a \leq y$ implies $a \leq x \vee y$. But also, since L is linear it follows easily that $a \leq x \vee y$ implies $a \leq x$ or $a \leq y$. Consequently $a \to (x \vee y) = (a \to x) \vee (a \to y)$. Now:

$$\begin{aligned} \theta_a(\langle p_1, p_2 \rangle \wedge \langle q_1, q_2 \rangle) &= \theta_a(\langle p_1 \wedge q_1, p_2 \vee q_2 \rangle) \\ &= \langle a \to (p_1 \wedge q_1), a \to (p_2 \vee q_2) \rangle \\ &= \langle (a \to p_1) \wedge (a \to q_1), (a \to p_2) \vee (a \to q_2) \rangle \\ &= \langle a \to p_1, a \to p_2 \rangle \wedge \langle a \to q_1, a \to q_2 \rangle \\ &= \theta_a(\langle p_1, p_2 \rangle) \wedge \theta_a(\langle q_1, q_2 \rangle) \end{aligned}$$

The other cases are similar, and we leave them to the reader. ∎

Now we have the main result of the section.

Theorem 10.5 *Let $\mathcal{B}$ be a distributive bilattice with negation that is bilinear. Let X and Y be formulas built up from propositional letters using $\wedge$, $\vee$, $\neg$, $\otimes$, $\oplus$, $:$, $\vec{\wedge}$, $\vec{\vee}$, $\wedge^w$, and $\vee^w$. Then the equivalence $X = Y$ is valid in $\mathcal{B}$ if and only if $X = Y$ is valid in $\mathcal{FOUR}$.*

Proof One half is quite trivial. Suppose $X = Y$ is not valid in $\mathcal{FOUR}$. Let v be a valuation under which X and Y map to different members of $\mathcal{FOUR}$. Now think of v as a valuation in $\mathcal{B}$ instead of $\mathcal{FOUR}$. Since $\top$, $\bot$, *false*, *true* behave the same in $\mathcal{B}$ as in $\mathcal{FOUR}$ under the operations we are considering, v will map X and Y to different members of $\mathcal{B}$ as well, so $X = Y$ is not valid in $\mathcal{B}$ either.

Next, suppose $X = Y$ is not valid in $\mathcal{B}$; where the valuation v establishes this. Using the Representation Theorems from Section 8, and Proposition 10.2, we can assume $\mathcal{B}$ is of the form $L \odot L$, where L is linear. Say $v(X) = \langle x_1, x_2 \rangle$ and $v(Y) = \langle y_1, y_2 \rangle$. Since $v(X) \neq v(Y)$, $x_1 \neq y_1$ or $x_2 \neq y_2$, say the first. Also since L is linear, $x_1 < y_1$ or $y_1 < x_1$. Again say the first; the other possibilities are treated similarly.

Now define a valuation v' in $\mathcal{FOUR}$ by: $v'(P) = \theta_{y_1}(v(P))$, for P a propositional letter. Using Lemma 10.4, for any formula Z, $v'(Z) = \theta_{y_1}(v(Z))$. In particular, $v'(X) = \theta_{y_1}(v(X)) = \theta_{y_1}(\langle x_1, x_2\rangle) = \langle y_1 \rightarrow x_1, y_1 \rightarrow x_2\rangle = \langle \mathit{false}, y_1 \rightarrow x_2\rangle$. But $v'(Y) = \theta_{y_1}(v(Y)) = \theta_{y_1}(\langle y_1, y_2\rangle) = \langle y_1 \rightarrow y_1, y_1 \rightarrow y_2\rangle = \langle \mathit{true}, y_1 \rightarrow y_2\rangle$. Since X and Y differ under v', $X = Y$ is not valid in $\mathcal{FOUR}$. ■

11 Conclusion

Bilinear bilattices include some of obvious interest, such as $[0,1] \odot [0,1]$. On the other hand, Theorem 10.5 says that as logics bilinear bilattices are not very interesting at all. We encourage others to investigate the family of distributive bilattices that are not bilinear. We feel all these bilattice structures are of intrinsic interest and have potential applicability. Encouraging further work is the main point of this paper.

References

[1] Belnap, Jr., N. D. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logic*, J. M. Dunn and G. Epstein, Eds. D. Reidel, 1977.

[2] Dunn, J. M. Natural language versus formal language. presented at the Joint APA-ASL symposium, New York, dec 1969.

[3] Dunn, J. M. Intuitive semantics for first-degree entailments and coupled trees. *Philosophical Studies 29* (1976), 149–168.

[4] Dunn, J. M. Relevance logic and entailment. In *Handbook of Philosophical Logic, Volume III*, D. Gabbay and F. Guenthner, Eds. D. Reidel, 1986, ch. 3, pp. 117–224.

[5] Fitting, M. C. Bilattices and the theory of truth. *Journal of Philosophical Logic 18* (1989), 225–256.

[6] Fitting, M. C. Negation as refutation. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science* (1989), R. Parikh, Ed., IEEE, pp. 63–70.

[7] Fitting, M. C. Bilattices and the semantics of logic programming. *Journal of Logic Programming 11* (1991), 91–116.

[8] Fitting, M. C. Kleene's logic, generalized. *Journal of Logic and Computation 1* (1992), 797–810.

[9] Fitting, M. C. The family of stable models. *Journal of Logic Programming 17* (1993), 197–225.

[10] Fitting, M. C. On prudent bravery and other abstractions. In preparation, 1993.

[11] Ginsberg, M. L. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence 4* (1988), 265–316.

[12] Kleene, S. C. *Introduction to Metamathematics*. D. Van Nostrand, Princeton, NJ, 1950.

[13] Kripke, S. Outline of a theory of truth. *The Journal of Philosophy 72* (1975), 690–716. Reprinted in *New Essays on Truth and the Liar Paradox*, R. L. Martin, ed., Oxford (1983).

[14] Smullyan, R. M. *First-Order Logic*. Springer-Verlag, 1968.

[15] Visser, A. Four valued semantics and the liar. *Journal of Philosophical Logic 13* (1984), 181–212.