Notre Dame Journal of Formal Logic
Volume 31, Number 1, Winter 1990

Motivation and Demotivation of a Four-Valued Logic

JOHN FOX*

Abstract Belnap offers two arguments for the usefulness of four-valued logic. I argue that one of them, which rests on interpreting valuations as states of our information, when taken seriously collapses into an argument for two-valued logic in which relevance is lost, and that the other, resting on Scott's thesis, is not an argument for its usefulness.

In this note I argue that Belnap's argument for the usefulness of four-valued logic, taken seriously, subsides into an argument for two-valued logic.

That fragment of the Ackermann-Anderson-Belnap logic of entailment in whose theorems the principal connective is the conditional arrow flanked only by truth functions is called the system of tautological entailment (TE). It can be axiomatized in various ways, all of which share the feature that the addition of any one of the "irrelevance" axioms

$$\begin{gathered} (A \vee B) \mathbin{\&} {\sim}A \rightarrow B \\ A \rightarrow B \vee {\sim}B \\ A \mathbin{\&} {\sim}A \rightarrow B \end{gathered}$$

yields the system, sometimes called that of tautological implication (TI), in which "$A \rightarrow B$" exactly captures "It is truth-functionally impossible that "$A \mathbin{\&} {\sim}B$" is true".

*Most of this note was read at the AAL Conference in Melbourne in October 1977. The final paragraphs on the "meaning" of the connectives were added in reply to a letter from Belnap on the earlier part. Two points of clarification I made in the conference discussion should also be appended. First: for use, I think the computer needs only the T, F and valuation-rules, not the arrow formulas which systematize for us its inference patterns. Second: I am not here criticizing either *relevance* as a motivation, or its Pittsburgh reading; I take it that the interpretation considered here was meant to motivate a relevance logic independently of considerations of relevance.

Received December 14, 1987; revised July 13, 1988

Smiley first discovered that TE was characterized by a four-valued matrix:

A	$\sim A$
t	f
b	b
n	n
f	t

		B			
	&	t	b	n	f
	t	t	b	n	f
A	b	b	b	f	f
	n	n	f	n	f
	f	f	f	f	f

		B			
	v	t	b	n	f
	t	t	t	t	t
A	b	t	b	t	b
	n	t	t	n	n
	f	t	b	n	f

$$\begin{tikzcd}
 & t & \\
b \arrow[ur] & & n \arrow[ul] \\
 & f \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Given the partial order on the four elements such that "$\rightarrow$" reads up the Hasse diagram, an expression of the form "$A \rightarrow B$" is valid exactly when for every assignment of elements to the sentence letters of A and B, $\mathrm{v}(A) \leq \mathrm{v}(B)$. It has been known for some years—first, I think, by Dunn—that reading "t" as "true", "f" as "false", "b" as "both true and false", and "n" as "neither true nor false" rendered the formal moves plausible; but since even in Pittsburgh nobody thought there were four such exemplified truth values, this wasn't widely taken as showing the usefulness of the system.

Belnap has interpreted the matrices in a way that he claims well motivates the system for some practical purposes. I will present a simple version that I hope captures this interpretation.

Let us assume a language L whose formulas express propositions, so that each is either true or false, and none is both. Let us also assume that we, or our computer, have various fallible sources of information about the truth values of formulas of L, and that this information is restricted to the atomic formulas (a.f.) of L. The first assumption is made in almost all uses of logic, and we attempt to ensure it by regimentation where this is needed; the second assumption is realistic; the third is a reasonable simplification, for the sake of working out the logic appropriate for the simpler case before advancing to the more complex.

Just as there are exactly two *truth* values, "true" and "false", of which each formula has exactly one, so there are exactly two *told* values, "T" and "F". An a.f. has T iff we have been informed that it is true, and F iff we have been informed that it is false. However, there are four *information-states*: {T}, {T,F}, $\varnothing$, {F}, which I relabel respectively t, b, n, f. It is essential not to confuse true, T, and t.

Given our told values for a.f.'s, which told values should we assign to truth-functional compounds?

Surely $\sim A$ should be assigned T exactly when A has been assigned F; letting "$\mathrm{v}(A)$" represent the information state for A, we have intuitively

(i) $\mathrm{T} \in \mathrm{v}(\sim A) \leftrightarrow \mathrm{F} \in \mathrm{v}(A)$
(ii) $\mathrm{F} \in \mathrm{v}(\sim A) \leftrightarrow \mathrm{T} \in \mathrm{v}(A)$
(iii) $\mathrm{T} \in \mathrm{v}(A \ \& \ B) \leftrightarrow \mathrm{T} \in \mathrm{v}(A) \ \& \ \mathrm{T} \in \mathrm{v}(B)$
(iv) $\mathrm{F} \in \mathrm{v}(A \ \& \ B) \leftrightarrow \mathrm{F} \in \mathrm{v}(A) \vee \mathrm{F} \in \mathrm{v}(B)$
(v) $\mathrm{T} \in \mathrm{v}(A \vee B) \leftrightarrow \mathrm{T} \in \mathrm{v}(A) \vee \mathrm{T} \in \mathrm{v}(B)$
(vi) $\mathrm{F} \in \mathrm{v}(A \vee B) \leftrightarrow \mathrm{F} \in \mathrm{v}(A) \ \& \ \mathrm{F} \in \mathrm{v}(B)$.

This motivates exactly the Smiley matrices. This is immediately clear in most instances, and soon clear in the odd ones, such as "b v n = t". For if $\mathrm{v}(A) =$ b, i.e. {T,F}, we have been told that it is true, and so that at least one of A and

B is true, and so that $(A \vee B)$ is true, so $T \in v(A \vee B)$; and if $v(B) = n$, i.e. $\varnothing$, we have been told neither that it is true nor that it is false, so we have not been told that it is false, nor, therefore, that (A v B) is false: so $F \notin v(A \vee B)$, so $v(A \vee B) = \{T\}$, i.e. t.

A schema "$A \rightarrow B$" is valid if the inference from a formula of the form A to the corresponding formula of the form B always preserves T. Equivalently, by the obvious duality of the lattice, it is valid if that inference never introduces F. The validity of a schema is ascertained effectively by an information-table calculation: A formula is valid iff it is an instance of a valid schema.

This interpretation yields TE; that it rejects disjunctive syllogism is shown by the following assignations, of which the first introduces F and the second fails to preserve T.

$$\begin{array}{cccccc} (A & \vee & B) & \& & {\sim}A & B \\ \mathrm{n} & \mathrm{n} & \mathrm{f} & \mathrm{n} & \mathrm{n} & \mathrm{f} \\ \mathrm{b} & \mathrm{b} & \mathrm{f} & \mathrm{b} & \mathrm{b} & \mathrm{f}. \end{array}$$

Such, I take it, is Belnap's argument: a lovely and plausible one (see pp. 41–42 of [1]).

I think it fails. We have motivated the valuation rules by bivalence and the truth-functionality of ~, v, and &. Strictly speaking, when our initial information is simply about a.f.'s, we are always *telling ourselves* what we can conclude about compounds on the basis of all our information plus what we know about truth functions. So, for instance, if we are given any information at all about A, the rules allow us to tell ourselves that $(A \vee {\sim}A)$ is true and that $(A \;\&\; {\sim}A)$ is false. But we sell ourselves—or our computer—short if we allow ourselves only selective use of what we know about truth functions. Since the truth of $(A \vee {\sim}A)$ and the falsehood of $(A \;\&\; {\sim}A)$ are thus independent of information, we can tell ourselves that they are true and false respectively, even when we have been given no information about A. So where $B = {\sim}A$, the left-to-right implications of (iv) and (v) should fail.

Since we know of each formula that it is either true or false but not both, we know that the truth about it is expressed either by its being t (told true and not told false), or by its being f (told false and not told true). So if a formula gets the same value in both these cases, that is its true value. Realizing this, we (or our computer) can assign information-values as follows. First, assign to the compound that one value, if any, which it gets on all uniform reassignments of t and f to components assigned b or n; failing that, assign values according to rules (i)–(vi), i.e., the Smiley matrices. On this suggestion, tautologies always emerge with value t and contradictions with value f.

Belnap's concern was of course not with which *formulas* emerge as always T (told-true), but with which rules of inference preserve T. The new suggested assignment is crucially different in this respect. For example, disjunctive syllogism comes out valid; the former counterexamples yield:

$$\begin{array}{cccccc} (A & \vee & B) & \& & {\sim}A & B \\ \mathrm{t} & \mathrm{t} & \mathrm{f} & \mathrm{f} & \mathrm{f} & \mathrm{f} \\ \mathrm{f} & \mathrm{f} & \mathrm{f} & \mathrm{f} & \mathrm{t} & \mathrm{f}. \end{array}$$

The "paradoxes" of "strict implication" recur; the conditional collapses into the strict conditional and the system collapses into TI; relevance is lost.

The motivation I have given here is neither the only nor the main one Belnap gives in [1] and [2]. The main argument he gives for the told-tables for the truth-functional connectives is this: Rotate the Hasse diagram through a right angle so that "b" is at the top. It is now an approximation lattice, in which each information state approximates to any above it. Belnap appeals to an idea inspired by Dana Scott, which he calls Scott's thesis, that the only important functions in an approximation lattice are the continuous ones, and remarks that in finite lattices only monotonic functions are continuous. Demand then monotonicity; demand also that the told-tables be classical (that is, that those parts involving only t and f mimic the truth-tables for True and False); and the told-table for negation is already determined. The tables for "v" and "&" are not; but if we treat them as informational joins and meets, demanding also that

$$A \vee B = B \text{ just if } A \mathbin{\&} B = A$$
$$A \vee B = A \text{ just if } A \mathbin{\&} B = B$$

then their tables are also determined (see [1], pp. 37–40, [2], pp. 13–14).

This is attractive, but I do not see how it constitutes any argument that programming a computer with the logic resulting from these tables would be useful for evaluating any inferences correctly.

In later parts of [1] and [2] Belnap offers a recursive account of what changes should be made to a computer's epistemic state if it is offered nonatomic data as input, and describes "the recursive clauses" as "a way of giving meaning to the connectives". The arguments that had yielded the told-tables and the system TE dealt solely with atomic data and depended on assumptions of bivalence and on our already understanding the connectives as truth-functional. Could it be argued that once these recursive clauses are offered as explicating the connectives, the earlier arguments for TE should be reinterpreted in their light, and that so reinterpreted, they become immune to my demotivation?

No. As is only appropriate if they are offered as giving the meaning of the connectives, the clauses do not use connectives themselves. They spell out the meaning of molecular data as input in terms of atomic data as input; thus, in the simplest cases, inputting "$\sim A$" as told true is equivalent to inputting A as told false, and inputting "A & B" as told true is equivalent to inputting first B and then A as told true. The recursion says nothing about the meaning of the connectives in output. If an account of the connectives' meaning in output is to be motivated by the account of their meaning in terms of atomic sentences as input, there is no suggestion about how it is to be done. Perhaps the told-tables do it; but the only motivation for these (and so for TE) remains that one presented earlier.

Perhaps the inputting and outputting accounts should be converse. But the only logic I know in which introduction and elimination rules for logical constants are both converses and are spelled out in terms of constant-free formulations is intuitionistic logic, not TE.

REFERENCES

[1] Belnap, N. D., "How a computer should think," pp. 30–56 in *Contemporary Aspects of Philosophy*, edited by G. Ryle, Oriel, Stocksfield, 1976.

[2] Belnap, N. D., "A useful four-valued logic," pp. 8–37 in *Modern Uses of Multiple-Valued Logics*, edited by J. M. Dunn and G. Epstein, Reidel, Dordrecht, 1977.

Department of Philosophy
La Trobe University
Bundoora Victoria
Australia 3083