First, a review of the progress is presented for the development of conditional event algebras. Following this, a new canonical bijection of isomorphisms is derived. This is an extension of the usual indicator function mapping to that between all possible truth-functional three-valued logics and all possible choices of conditional event operators extending unconditional boolean ones. Relations between the conditional event algebra proposed by Goodman & Nguyen and $\mathcal{E}^3$, as well as that proposed by Schay, Adams, and Calabrese and $\mathcal{S}^3_3$ are derived, among other isomorphic correspondences.

Published in Proceedings of ISUMA '90, Dec 1990.
<table>
<thead>
<tr>
<th>21a NAME OF RESPONSIBLE INDIVIDUAL</th>
<th>21b TELEPHONE (Include Area Code)</th>
<th>21c OFFICE SYMBOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. R. Goodman</td>
<td>(619) 553-4014</td>
<td>Code 421</td>
</tr>
</tbody>
</table>
Three-Valued Logics and Conditional Event Algebras

I. R. Goodman

Code 421
Command & Control Department
Naval Ocean Systems Center
San Diego, CA 92152-5000

Abstract

First, a review of the progress is presented for the development of conditional event algebras. Following this, a new canonical bijection of isomorphism is derived. This is an extension of the usual indicator function mapping to that between all possible truth-functional three-valued logics and all possible choices of conditional event operators extending unconditional boolean ones. Relations between the conditional event algebra proposed by Goodman & Nguyen and $\mathcal{F}_3$, as well as that proposed by Schay, Calabrese and Sob, are derived, among other isomorphic correspondences.

Note on General Notation, Conventions

In addition to the usual use of equality $=$, set inclusion $\in$, set membership $\in$ and class of all subsets of or power class $\mathcal{P}(\cdot)$, null set $\emptyset$, etc., we introduce $\updownarrow$ to mean "is defined to be," emphasizing the difference between provable, as in $\updownarrow$ and the former. Throughout, $\mathbb{R}$ (as opposed to $\mathbb{R}$ for the real line) stands for an arbitrary but fixed nontrivial boolean algebra of events or sets $\{a, a_0, \ldots, b, b_0, \ldots, c, \ldots\}$. When $\mathbb{R}$ is considered (via the Stone Representation Theorem or directly) to be such that $\mathbb{R}$ is isomorphic to that between all possible derivable, this, a new development of conditional event algebras. Followini

Introduction

Conditional events have been developed in order to provide a systematic way to determine evaluations of arbitrary logical combinations of conditional or implicational statements with differing antecedents, so that each is consistent with conditional probability. Thus, when one seeks to obtain the probability of a compound statement such as "((if $b$ then $a$) or (if $d$ then not($c$)) but not $e$)", traditional methods are inadequate in dealing with this. For example, if the well-known material implication is used to interpret the conditionals so that ordinary boolean algebra and properties of probability can be used for the full evaluation, before proceeding one should note that the probabilities do not match the corresponding conditional probability forms:

\[ p(b \rightarrow a) \neq p(a) \cdot p(b) / p(b) \quad \text{for} \quad p(b) \neq 0, \]

and similarly for "if $d$ then not($c$)". In fact, it can be seen ([1], p. 201) that

\[ p(b \rightarrow a) = 1 - p(b) + p(ab) \cdot p(a') / p(b) \cdot (p(b) \neq 0, \]

with strict inequality holding in general. Going further, Calabrese ([1], ch. 22) showed no binary boolean function $g: \mathbb{R} \rightarrow \mathbb{R}$ exists (of the 16 possible ones) for which

\[ p(g(a, b)) = p(ab), \quad p(a, b) \cdot p(b, a); \quad \text{all} \quad a, b \in \mathbb{R}, \text{ all} \quad p: \mathbb{R} \quad \text{in} \quad \text{g} \quad \text{with} \quad \text{strictly satisfying} \]

(3)

or "conditional" events, see Goodman & Nguyen ([2]). Briefly, one should mention the original contribution of Boole ([5], ch. 6.4), Haller's rigorizing of Boole's attempts ([6], DeFinetti's work [7], Schay's efforts [8], Adams' work ([9], chp. II), and more recently, Calabrese [1] and Bruno & Gilio [10], among others. In all of the above, only DeFinetti and Schay considered conditional events through extensions of the usual indicator function, with only Schay developing a full conditional event algebra. Adams proposed extensions of the usual boolean operators to conditional forms, but did not give any real interpretation to what conditional events meant, nor did he investigate to the depth that Calabrese carried out in the latter's fully developed conditional event algebra.

Conditional Events Identified as Principal Ideal Cosets

In response to the previous unconnected efforts, Goodman [11] and Goodman & Nguyen [12],[13],[14] developed a fresh approach to conditional event algebra. Recall the basic concept of the principal ideal
in \( R \) generated by any \( b' \in R \) as \( Rb' = (rb' : x \in R) \subseteq R \), leading to the boolean quotient algebra \( R/Rb' = (Rb' : a \in R) \) with the usual well-defined coset operations for the cosets \( Rb' : a, b \in R \). Denote the class of all \( r \in R \) principal ideal cosets of \( R \) as \( R \rceil (Rb') \) and the natural mapping \( R \rceil Rb' \). 

If \( g : R \rightrightarrows S \) is to be a reasonable candidate for a conditional event of the form \( g(a,b) \), for any \( a, b \) events, then Lewis’ result shows that at least \( S g R \), when range(\( g \)) = \( S \). In addition, one should assume that:

(i) Antecedent-consequent invariance:

\[
\forall (a,b) \in R, (a,b) = (b,a) 
\]

(ii) Unique global representation:

\[
g(a,b) = g(b,a) 
\]

(iii) Allows for the definition:

\[
p(g(a,b)) = p(ab) 
\]

(iv) Note the relations for all \( a, b, c, d \):

\[
(a \equiv b) = (c \equiv d) \iff (c \equiv d) = (a \equiv b) \iff abc'd = a'b'd = c'd = a'b 
\]

(v) Note the relations for all \( a, b, c, d \):

\[
(a = b) = (c = d) \iff (c = d) = (a = b) \iff a = b 
\]

Theorem 1. Goodman & Nguyen [3], chp. 2.

(i) \( g \) is a feasible candidate for forming conditional events and for each fixed \( b, g(-, b) : R \rightrightarrows R/\mathbb{R}b' \) is a homomorphism wrt coset operations. This allows for the definition:

\[
p(g(a,b)) = p(ab) 
\]

(ii) If \( g : R \rightrightarrows S \) is any feasible candidate for forming conditional events, \( g \) globally isomorphic to \( n \). That is, \( R \) exists bijection \( \mathbb{S} R \mathbb{S} \) over \( R/\mathbb{R}b' \), which is of unity type.

Remark.

(i) The above theorem justifies naturally the chose of principal ideal cosets of \( R \) for its conditional events, so that one defines for all \( a, b \in R \), \( (a,b) = (b,a) \in R \), \( R \rightrightarrows R/\mathbb{R}b' \), and the natural mapping \( R \rightrightarrows R/\mathbb{R}b' \).

(ii) If \( g : R \rightrightarrows S \) is any feasible candidate for forming conditional events, then \( g \) is globally isomorphic to \( n \). That is, \( R \) exists bijection \( \mathbb{S} R \mathbb{S} \) over \( R/\mathbb{R}b' \), which is of unity type.

(iii) Special conditional events: un condolitional events \( a = (a) \), whence \( R \subseteq R(R) \); the unconditional conditional event \( (a) = (0) \); the unity-type conditional event \( (b) = (b) \in R/\mathbb{R}b' \); principal filter generated by \( b \) in \( R, \mathbb{F}_b \); the zero-type conditional event \( (0) = (0) \in R/\mathbb{R}b' \); principal generator by \( b \) in \( R, \mathbb{P}_b \).

(iv) Note the relations for all \( a, b, c, d \in R \):

\[
(a \equiv b) = (c \equiv d) \iff (c \equiv d) = (a \equiv b) \iff abc'd = a'b'd = c'd = a'b 
\]

(v) Note the relations for all \( a, b, c, d \):

\[
(a = b) = (c = d) \iff (c = d) = (a = b) \iff a = b 
\]

Conditional Events Identified with Three-Valued Indicator Functions

De Finetti [7] and, independently, Schay [g] extended the ordinary indicator functions of sets to three values to represent conditional events by the mapping \( \phi : R \rightarrow (0, a, \epsilon) \), assuming \( R \subseteq \mathbb{P}(R) \), where \( w(g) \), we also assume here the third value is \( a \) (entire unit interval). From now on denote \( (0, a, \epsilon) \) by \( Q_0 \). Then, for all \( a, b \in R \),

\[
\phi(ab)(w) = \begin{cases} 1, & \text{if } w = ab \\ 0, & \text{if } w \neq ab \\ \epsilon, & \text{if } w \neq a'b \\ \end{cases} 
\]

The following theorem can help motivate the choice of operators over \( (R) \) extending the boolean ones over \( R \), assuming \( R \) is atomic and noting \( \mathbb{S} = \{a \} \).

Theorem 2. Goodman & Nguyen [3], chp. 5.

Let \( \mathbb{S} = \{a \} \). For any \( (a, b, c) \subseteq (R) \),

\[
\phi(ab) \neq (a) \iff (a, b) \neq (a) \iff ab \neq a \\
\phi(ab) \neq (b) \iff (a, b) \neq (b) \iff ab \neq b \\
\phi(ab) \neq (c) \iff (a, b) \neq (c) \iff ab \neq c \\
\phi(ab) \neq (d) \iff (a, b) \neq (d) \iff ab \neq d \\
\phi(ab) \neq (e) \iff (a, b) \neq (e) \iff ab \neq w \\
\phi(ab) \neq (f) \iff (a, b) \neq (f) \iff ab \neq \epsilon \\
\phi(ab) \neq (g) \iff (a, b) \neq (g) \iff ab \neq 0 \\
\phi(ab) \neq (h) \iff (a, b) \neq (h) \iff ab \neq \epsilon \\
\phi(ab) \neq (i) \iff (a, b) \neq (i) \iff ab \neq \epsilon \\
\phi(ab) \neq (j) \iff (a, b) \neq (j) \iff ab \neq \epsilon \\
\phi(ab) \neq (k) \iff (a, b) \neq (k) \iff ab \neq \epsilon \\
\phi(ab) \neq (l) \iff (a, b) \neq (l) \iff ab \neq \epsilon \\
\phi(ab) \neq (m) \iff (a, b) \neq (m) \iff ab \neq \epsilon \\
\phi(ab) \neq (n) \iff (a, b) \neq (n) \iff ab \neq \epsilon \\
\phi(ab) \neq (o) \iff (a, b) \neq (o) \iff ab \neq \epsilon \\
\phi(ab) \neq (p) \iff (a, b) \neq (p) \iff ab \neq \epsilon \\
\phi(ab) \neq (q) \iff (a, b) \neq (q) \iff ab \neq \epsilon \\
\phi(ab) \neq (r) \iff (a, b) \neq (r) \iff ab \neq \epsilon \\
\phi(ab) \neq (s) \iff (a, b) \neq (s) \iff ab \neq \epsilon \\
\phi(ab) \neq (t) \iff (a, b) \neq (t) \iff ab \neq \epsilon \\
\phi(ab) \neq (u) \iff (a, b) \neq (u) \iff ab \neq \epsilon \\
\phi(ab) \neq (v) \iff (a, b) \neq (v) \iff ab \neq \epsilon \\
\phi(ab) \neq (w) \iff (a, b) \neq (w) \iff ab \neq \epsilon \\
\phi(ab) \neq (x) \iff (a, b) \neq (x) \iff ab \neq \epsilon \\
\phi(ab) \neq (y) \iff (a, b) \neq (y) \iff ab \neq \epsilon \\
\phi(ab) \neq (z) \iff (a, b) \neq (z) \iff ab \neq \epsilon \\
\end{cases} 
\]

Functional Image Approach to Extending Boolean Operators over \( R \) to \( \mathbb{R} \)

As mentioned before, Adams, Schay, and Calabrese have independently proposed extensions of boolean operators to \( (R) \), details of which will be shown later. These operators were based upon empirically appealing, but ad hoc, considerations. The thinking of Goodman & Nguyen has been, on the other hand to use the natural way one extends "point"-valued functions to set valued ones: \( R \subseteq Y \) extends the well-known functional image approach to simply \( g : (R \subseteq X) \rightarrow Y \), via \( g(A) = g(x) \in x \in A \). Since \( (R \subseteq X) \subseteq (R) \), it seems reasonable to attempt to extend the ordinary boolean operators over \( R \) to the functional image approach restricted to \( R \), with the expectation that closure holds not just for \( P(R) \) (trivially), but for \( (R \subseteq R) \) itself. This is indeed so:

Theorem 3. [12], [13].

For all \( a, b, c, d, a, b, c, d, j, l \in \mathbb{R} \), arbitrary:

\[
(a, b) = (c, d) \iff (a, b) = (c, d) \iff ab = cd & b = d 
\]

Remarks.

(i) The indeterminate element is the only \( (a, b) \) for which \( \phi(ab) \equiv (a) \equiv (b) \).

(ii) It is desirable to obtain a conditional event algebra of operations yielding a partial order over \( \mathbb{R} \), extending the unconditional counterpart \( \mathbb{R} \) over \( \mathbb{R} \), compatible with Theorem 2. This is seen to be the case as presented in the next section.
Remarks.
(i) Theorem 3 shows that any finite logical combination of logical connectors of conditional statements can be evaluated compatible with all probability evaluations, thus addressing the motivating problem for developing conditional event algebras.

(ii) Applying Theorem 3(ii) to Theorem 2 answers in the affirmative the remark, part (ii) following Theorem 2: the extended lattice or partial order \( \preceq \) over \((R[R])^2\) yields the compatibility, for all \((a,b),(c,d)\in(R[R])^2\) with \((a,b)\) not zero-type, \((c,d)\) not unity type: \(\psi(a,b)\psi(c,d)\) over \(\preceq\) if and only if \((a,b)c(RJR)\) iff for all \(p\in P\) \(p(a)p(b)p(d)>0\), \(p(a)b(p(c))p(d)\). (16)

(iii) A third type of justification for employing the conditional event algebra proposed here is provided by the next theorem, where it is seen: that this conditional event algebra has almost all the properties of a boolean algebra that it can be evaluated compatible with all probability \( p:R\rightarrow R\). In turn, if the standard Stone representation mapping is denoted as \(M:R\rightarrow P(R)\), for any boolean algebra \(R\), an injective isomorphism, it can be shown that the mapping \((m\in P(R))=P(R)\) is also an injective isomorphism, extending \(m\), where \((m\in P(R)\) is an injective isomorphism, providing a concrete representation for any such abstract conditional event algebra.

Basic Isomorphism between All 3-Valued Truth-Functional Logics and All Boolean-Extended Conditional Event Algebras

In the last section a compact detailed structural analysis of the Goodman & Nguyen [abbreviated from now on as GH] conditional event algebra was given. Much remains to be analyzed for the other leading candidate conditional event algebras, including the independently considered, but commonly structured, proposal of Schay (alternative choice one of two proffered [5]), Adams' [9], and Calabrese [1] (abbreviated from now on as SAC), and another of Schay's (alternative choice two- see again [8] [abbreviated from now on as simply S]). However, Schay (5, Theorem 5) has derived Stone-like representations for, in effect, both SAC and S, corresponding to part of Theorem 4(ii) above.

For completeness, the basic operators for SAC and S are given below, with appropriately subscripted letters for all \((a,b),(c,d)\in(R[R])\). Once more, it is emphasized that GH, SAC, and S all agree on the essential structure of \((R[R])\)-sans any algebraic operations, other than the classical coset ones for each fixed antecedent principal ideal boolean quotient algebra of parent boolean algebra \(R\):

\[
(a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = \ldots (26)
\]

\[
(a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = \ldots (27)
\]

\[
(a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = \ldots (28)
\]

\[
(a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = \ldots (29)
\]

\[
(a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = (a\cdot b)'SAC = \ldots (30)
\]

In addition, recently, Dubois & Prade [16], [17] have expressed interest in the development of the candidate condition event algebras. In [16], pp. 1112, 1113 and [17], pp. 31-34, they have pointed out that the following correspondences hold between the three basic candidates and certain three-valued logics (although this was previously also indicated in [18] in preliminary form), using an informal argument:

\[
SAC \leftrightarrow SOb_3; \quad S \leftrightarrow B_3; \quad GH \leftrightarrow S_3.
\]

where \(SOb_3\) indicates Sobociński's three-valued logic (see [19] or Rescher [20], pp. 70, 342), \(B_3\) is Bouchvar’s internal three-valued logic ([20], pp. 29-34), and \(S_3\) is Łukasiewicz’ three-valued logic ([20], pp. 22-28 and 335).
In this section a general theorem will be fully derived which constructively establishes an isomorphism between any choice of three-valued truth functional logical operator and any extended boolean conditional event operator (for definition, see below). First, some additional notation for multiple variables, as well as other concepts must be introduced (R ∈ R(R)). Let n be any positive integer and a,b,a₁,a₂,...,aᵢ ∈ R arb: a = (a₁,...,aᵢ), b = (b₁,...,bᵢ) ∈ (RIR)n c Rn ;
Let n be any positive integer and a,b,a₁,a₂,...,aᵢ ∈ R arb:
- *(a) ᵐa₁...aᵢ a ∈ R; (a)b = ((a₁)b₁,...,(aᵢ)bᵢ) c (RIR)n and extend the three-valued indicator function for any wεΩ as
- φ(a₁b₁,...,aᵢbᵢ) c Rn (32)
Define the mappings w₁ : (RIR) → Ω, for (RIR)n, by
- w₁(a₁b₁,...,aᵢbᵢ) = φ(a₁b₁,...,aᵢbᵢ) (33)
and extending this, for any j = (j₁,...,jₙ) ∈ Qn, j ≠ 0,
- w₁(a₁b₁,...,aᵢbᵢ) = (w₁(a₁b₁),...,w₁(aᵢbᵢ)) c Rn (34)
Also, booln(R) ⊂ Ω, for g : R → Ω is a boolean function
and for any pair g, c ∈ booln(R), define the extended boolean function over (RIR)n, (g₁,g₂) : Ω × Ω → (RIR)n, where for any (g₁,g₂) = ((g₁g₂),(g₁g₂)) c (RIR)n ;
- (g₁g₂)(a₁b₁,...,aᵢbᵢ) = (g₁(a₁b₁),g₂(aᵢbᵢ)) (35)
Lemma 1. (i) For any (a₁b₁)...(aᵢbᵢ) ∈ (RIR)n, there is a unique partitioning of R, and more generally, for all i ∈ Qn, φ(a₁b₁,...,aᵢbᵢ) is a partitioning of R, so that
- φ(a₁b₁,...,aᵢbᵢ) = (w₁(a₁b₁),...,w₁(aᵢbᵢ) d w(a₁b₁,...,aᵢbᵢ) d w(a₁b₁,...,aᵢbᵢ) d w(a₁b₁,...,aᵢbᵢ)
(ii) φ(a₁b₁,...,aᵢbᵢ) = φ(a₁b₁,...,aᵢbᵢ), all i ∈ Qn , and more generally, for all i ∈ Qn, φ(a₁b₁,...,aᵢbᵢ) = φ(a₁b₁,...,aᵢbᵢ)
Lemma 2. For each ε ∈ booln(R), there is a minimal classwise nonvacuous index set j₂ ≤ j₁ ≤ Qn such that
g(a₁b₁,...,aᵢbᵢ) = ε(w₁(a₁b₁),...,w₁(aᵢb₁)) (37)
Proof: Use normal disjunctive form for boolean functions.
Theorem 5. Let g : (RIR)n → (RIR) be arbitrary in (booln(R)mod booln(R)) d g = (g₁,g₂) ext booln, over (RIR)n
Then there is a unique function φ : Ω × Ω → Ω such that for all i ∈ R, ε ∈ Ω, φ(g(a₁b₁,...,aᵢbᵢ)) = φ(a₁b₁,...,aᵢbᵢ) (38)
Proof: Direct result of combining Theorems 5 and 6.
Corollary 1. Referring to Theorems 5 and 6:
(i) φ : booln(R) → booln(R) × Qn is a bijection which makes any ε ∈ booln(R) commute with the three-valued indicator mapping φ : RIR → Qn in the sense
- φ(g) = φ(g)φ (50)
(i.e., for all (a₁b₁,...,aᵢbᵢ) ∈ (RIR)n, φ is an isomorphism relative to (booln(R)mod booln(R)) over (RIR)n and Qn relative to Qn)
algebras via three-valued logics and vice versa. The next sections show how Corollary 1 (or Theorems 5 or 6) can be used to compare and contrast properties for various candidate conditional event algebras in addition to the three discussed earlier.

Further Results Using the Basic Isomorphism

Example illustrating conditional event algebra operations converted to 3-valued logic.

As an example how the constructive proof in Theorem 5 can be used, consider again the operator $g$ as in Theorem 3 applied to $\mathbf{g}(\mathbf{a'b,})$ but converted to 3-valued logic. As an example how the constructive proof in Theorem 5 can be used, consider again the operator $g = \mathbf{g}(\mathbf{a'b,})$, as in Figure 1.4, where

$$g_1(\mathbf{a'b,b}) = abd', b'cd'vabcd = \mathbf{w}_1(\mathbf{a'b})\mathbf{w}_2(\mathbf{c'd}) \vee$$

$$\mathbf{w}_3(\mathbf{a'b})\mathbf{w}_4(\mathbf{c'd}) \vee \mathbf{w}_5(\mathbf{a'b})\mathbf{w}_6(\mathbf{c'd}),$$

whence $g_1 = ((1,1), (1,1), (1,1), (1,1), (1,1), (1,1))$.

Then, from eq. (20),

$$g_1 = g_1, g_2= g_2, g_2 = g_2, g_2 = g_2,$$

whence $g_2 = (1,1), (1,1), (1,1), (1,1), (1,1), (1,1))$.

Thus, for all $j=1,2$ we have $g_j = g_j, g_j = g_j, g_j = g_j$.

Figure 1. Partitioning of values for the 3-valued logic operator corresponding to $\mathbf{g}(\mathbf{a'b,})$ via procedure of Theorem 5.

Example illustrating 3-valued logic operators converted to conditional event algebra.

As an example how the constructive proof in Theorem 6 can be used, consider the three-valued logical operator given in Figure 1. We will show how the original conditional event operator - in this case $\mathbf{g}(\mathbf{a'b,})$ - can be recovered, knowing only the entries in the table.

First, obtain from the table, denoted as 3-valued logical operator $h$ (replacing $\mathbf{g}(\mathbf{a'b,})$),

$$h^{-1}(0) = (0,0,0,0,0,0,0) ; h^{-1}(1) = (1,1,1,1,1,1,1).$$

Next, obtain for any $(\mathbf{a'b,b})(\mathbf{a'b,b})$, $\mathbf{c'd,}$

$$c_1(\mathbf{a'b,b})\mathbf{w}_1(\mathbf{a'b})\mathbf{w}_2(\mathbf{c'd}) \vee$$

$$\mathbf{w}_3(\mathbf{a'b})\mathbf{w}_4(\mathbf{c'd}) \vee \mathbf{w}_5(\mathbf{a'b})\mathbf{w}_6(\mathbf{c'd}),$$

which of course checks with $\mathbf{g}(\mathbf{a'b,})$ via procedure of Theorem 5.

Applications to Comparing/Contrasting Conditional Event Algebras

Using the procedure in the examples, one can verify rigorously Dubois & Prade’s conclusions in (31):

Corollary 2. $\psi: (\mathbf{RIR})^n \to \mathbf{B}_3$ is an isomorphism relative to:

(i) SAC-conditional event algebra over $(\mathbf{RIR})^n$ and $\mathbf{B}_3$ logic over $\mathbf{B}_3^n$.

(ii) $\mathbf{S}$-conditional event algebra over $(\mathbf{RIR})^n$ and $\mathbf{B}_3$ logic over $\mathbf{B}_3^n$.

(iii) $\mathbf{G}$-conditional event algebra over $(\mathbf{RIR})^n$ and $\mathbf{B}_3$ logic over $\mathbf{B}_3^n$.}

* Next, consider a number of desirable properties that a conditional event algebra should possess. By use of the transfer technique above, in general it will be more convenient to analyze the candidate conditional event algebras for these properties via the three-valued logic form, rather than in the original form. However, these properties will be given in the latter form initially with a circle about the corresponding ordinary boolean operator to indicate the generic form.

Details are not required for the standard concepts of associativity, commutativity, and idempotence for $\mathbf{SAC}$, involutiveness for $\mathbf{SAC}$, $\mathbf{G}$, $\mathbf{S}$, being orthocomplemented (i.e., law of excluded middle holds) or being a De Morgan triple, or, finally, for $\mathbf{G}$ being mutually distributive. In addition, define the following by the associated equations for all $\mathbf{a'b,c'd}$ in $(\mathbf{RIR})$:

monotonicity $(\mathbf{a'b,c'd}) \leq \mathbf{a'b,c'd},$ $(\mathbf{a'b,c'd}) \geq \mathbf{a'b,c'd},$

zero-unity $(\mathbf{a'b,c'd}) \leq \mathbf{a'b,c'd},$ $(\mathbf{a'b,c'd}) \geq \mathbf{a'b,c'd},$

common antecedent $(\mathbf{a'b,c'd}) \leq (\mathbf{a'b,c'd}),$ $(\mathbf{a'b,c'd}) \geq (\mathbf{a'b,c'd}),$

chaining $1. (\mathbf{a'b,c'd}) \leq (\mathbf{a'b,c'd}), (\mathbf{a'b,c'd}) \geq (\mathbf{a'b,c'd}),$

full lattice $(\mathbf{a'b,c'd}) = (\mathbf{a'b,c'd}),$ $(\mathbf{a'b,c'd}) = (\mathbf{a'b,c'd}),$

full compatibility $(\mathbf{a'b,c'd}) \leq (\mathbf{a'b,c'd}), (\mathbf{a'b,c'd}) \geq (\mathbf{a'b,c'd}),$

$(\mathbf{a'b,c'd})$ not zero-, $(\mathbf{a'b,c'd})$ not unity-types.
The candidate conditional algebras to dates satisfying the required constraints. These are all presented in Table 2 below:

In turn, Table 1 allows only four possible candidates satisfying the required constraints. These are all presented in Table 2 below:

Clearly, the fourth suitable above is the same as conjunction, i.e., which already has been introduced as corresponding to GN. It should also be noted that the second suitable above corresponds to the important connector cop, the smallest possible copula, where, for all s, t ∈ \(\mathbb{A}\),

\[
\text{cop}_0(s, t) \equiv \max(s + t - 1, 0) \quad (a = \text{'and'}, b = \text{'or'})
\]  

(51)

[See [21] for background.] cop also plays a role in the operation for a Chang or MV algebra, where cop is the De Morgan dual of \(\text{cop}_0\) (and hence the maximal such one)

\[
\text{cop}_m(s, t) \equiv \min(s + t + 1, 1) \quad (a = \text{'and'}, b = \text{'or'})
\]  

(52)

[See [22, p. 473 et passim for further details.]

Also, for completeness, the 3-valued logical tables corresponding to the three leading candidates will now be displayed for the conjunction operators:

Clearly, the fourth suitable above is the same as conjunction, i.e., which already has been introduced as corresponding to GN. It should also be noted that the second suitable above corresponds to the important connector cop, the smallest possible copula, where, for all s, t ∈ \(\mathbb{A}\),

\[
\text{cop}_0(s, t) \equiv \max(s + t - 1, 0) \quad (a = \text{'and'}, b = \text{'or'})
\]

(51)

[See [21] for background.] cop also plays a role in the operation for a Chang or MV algebra, where cop is the De Morgan dual of \(\text{cop}_0\) (and hence the maximal such one)

\[
\text{cop}_m(s, t) \equiv \min(s + t + 1, 1) \quad (a = \text{'and'}, b = \text{'or'})
\]

(52)

[See [22, p. 473 et passim for further details.]

Also, for completeness, the 3-valued logical tables corresponding to the three leading candidates will now be displayed for the conjunction operators:

Applying the transfer procedure of the second example, yields the following conditional event algebra correspondences to the conjunction operators in Table 2, for all \((a|b), (c|d) \in (R|R)\) the disjunction being just the De Morgan dual:

\[
(a|b)_1(c|d) = abcd
\]

(53)

\[
(a|b)_2(c|d) = (abcd|a'b v c'd v abcd v b'd')
\]

(54)

\[
(a|b)_3(c|d) = (abcd|v v).
\]

(55)

Finally, as a check with eq.(10),

\[
(a|b)_4(c|d) = (abcd|a'b v c'd v abcd).
\]

(56)

Thus, in summary, the candidate conditional event algebras considered are represented by their conjunction operators, which are given in eqs.(28),(30),(10), and (53)-(55), while their corresponding 3-valued logical conjunction operators are given in Tables 3 and 2. All of this leads to the next table providing a comparison and contrasts for the above 6 systems, again obtained via the transfer technique, based upon Theorems 5 and 6:

Table 3. Conjunctions for 3-valued event algebra

<table>
<thead>
<tr>
<th>(\text{cop}_0)</th>
<th>(\text{cop}_1)</th>
<th>(\text{cop}_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\text{cop}_0(s, t) \equiv \max(s + t - 1, 0) \quad (a = \text{'and'}, b = \text{'or'})
\]

(51)

\[
\text{cop}_1(s, t) \equiv \min(s + t + 1, 1) \quad (a = \text{'and'}, b = \text{'or'})
\]

(52)

Table 4. Comparisons of properties for 6 candidate conditional event algebras.

<table>
<thead>
<tr>
<th>Conditional Event Algebra</th>
<th>SAC</th>
<th>S</th>
<th>GN</th>
<th>cmD</th>
<th>cmD</th>
<th>cmD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\ominus) associative</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(\ominus) commutative</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(\ominus) idempotent</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>(\ominus) involutive</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(\ominus) orthocycle</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(\ominus) De Morgan</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>(\ominus) mut. distrib.</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>zero-unity</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>comm. ante. homomorph.</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>chaining prop. 1</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>chaining prop. 2</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>full (\ominus) lattice</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>full compatible</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>logical ent. pres.</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>logical equiv. pres.</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>rel. pseudocompl.</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

\(t\), if YES, only if the consequent of material implication \(((c|d)|v (a|b))\), i.e., \(c'd v ab\) (using eq.(10)), is in place of the usual \(\rightarrow\) implication, which by applying Theorem 6 to [22], p.23 is in fact in the form \(b\rightarrow c\) \(\leftrightarrow ((c|d)|v (a|b))\) = \(\text{ab} \leftrightarrow \text{cd}\). This equivalence, which by applying Theorem 6 to [22], p.23 is in fact in the form \(b\rightarrow c\) \(\leftrightarrow \text{ab} \leftrightarrow \text{cd}\). The response of SAC and the partial YES of GN (see \(t_1, t_2\) above) are due to a characterization that these are the only possible systems preserving logical entailment and logical equivalence tautologically. (See [3].)
(ii) Additional properties of GI can be found in [3] where higher order conditional events and their homomorphic reductions are considered, as well as development of a conditional probability logic of propositions and the issue of relating the classical assignment of conditional probability to conditional events as functional image extensions. Furthermore, relations are developed between conditional random variables and CSs (conditional events) as well as between qualitative conditional probability and CSs with interpretations for their outcomes through \( \phi \).

(iii) In an alternative direction, McCarthy has developed a three-valued logic responsive to the spirit of flow diagrams "if then, else..." [23], which has been greatly expanded and analyzed by Guzman & Squier [24], relating to a Kleene regular extension of classical logic. However, none of this has been related to probability computations in the sense discussed in this paper. It is of some interest, however, to be able to convert this non-commutative logic into a conditional event algebra. In particular, the proposed conjunction operator is given by the table

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Using Theorem 6, it readily follows that

\[ (a|b|c|d) = \text{abcd} (a|b|d|c) \]

\[ (a|b|c|d) = \text{abcd} (a|b|d|c) \]

\[ (a|b|c|d) = \text{abcd} (a|b|d|c) \]

Acknowledgments

The author wishes to acknowledge the support of the line management at NOSC, especially, H.K. Kudurian, Head of Advanced Concepts and Development Branch, and J.A. Saltzmann, Head of Ashore Command and Intelligence Centers Division. In addition, acknowledgments are given here for the direct support of the Office of Naval Research, Code 1123 (under the sponsorship of Dr. R. Wachter) and the NOSC Program for Research and Technology, Code 014.

References