Universal logic VŁ4

1. Abstract

This paper demonstrates why logic system VŁ4 is a universal logic composed of any refutation as a non-tautologous fragment. Recent advances are a definitive answer to criticism of logic Ł4, modal equations for lines and angles of the Square of Opposition, confirmation of the 24-syllogisms by updating Modus Cesare and Camestros, and proving that respective quantified and modal operators are equivalent. The parser Meth8 implements VŁ4 as the modal logic model checker Meth8/VŁ4. Over 435 assertions are tested in about 2400 assertions with a refutation rate of 81%.

2. Introduction

2.1. Outline

This paper proves that an exact, bivalent, quaternary logic is *not* a probabilistic, vector space. From the four-values of the 2-tuple, logical assignments are derived for two models in logic system B4. Modal values are further ascribed for system Ł4 with truth tables for connectives. A criticism of Ł4 is answered by trivial proof. The Square of Opposition is corrected with modal equations for vertices and edges. Corrections are made to two of the 24-syllogisms as confirmed. The quantifiers are shown equivalent to the respective modal operators as a distinguishing feature for system VŁ4. The Meth8 parser hosts and implements VŁ4 as a modal logic model checker. Meth8/VŁ4 tested over 435 artifacts in about 2400 assertions for a rate of 19% confirmation and 81% refutation. The seven examples given are for refutations.

2.2. Overview of literature

Universal logic owns a public domain corpus published at encyclopedia web sites with lists of marginal, secondary references. A few primary sources describe non-standard and paraconsistent logic as appropriated by three writers, but traceable to earlier concepts as minimized or suppressed. Until now, there is *no* literature on bivalent, modal, quaternary, universal logic.

3. B4 as a group, ring, module

In (James, 2010), the 2-tuple of logic B4 was described as:

> Four value bit code (4vbc) consists of four dibits that have the semantic meanings of True {01} and False {10} and the syntactic meanings of Bivalent {11} and Not Bivalent {00}. The respective left- and right-bits are further variables for false and true. Two dibits (4-bits) form the basis of PMDL, a universal logic for propositional, modal, and deontic logics. PMDL has three levels of tabular proofs as negation, rotation, and reflection. This paper proves that 4vbc constitutes its own mathematical category as a group, ring, and module.

The outline of the proof was:

4vbc contains unique 8-bit operators that are tabulated into 256 look up tables (LUTs). The additive table for the small finite field $\mathbf{F}_4$ is isomorphic in 4vbc to the LUT of the logical operator “XOR”. 4vbc is not isomorphic to the $\mathbf{F}_4$ multiplicative table which is *bit-inconsistent*. Hence 4vbc is not a vector space. The modulo 2 additive table of the elementary Abelian group $(Z/2Z)^2$ is isomorphic in 4vbc to the LUT of the logical operator “Necessarily XOR”. $(Z/2Z)^2$ is the finite group C_2xC_2 that is a distinct group of order 4 and is not cyclic. A Cayley table as the representation of a multiplicative table of C_2xC_2 table is isomorphic in 4vbc to the LUT of the logical operator “10 EQV(XOR)”. (Another distinct group of order 4 is the C_4 group that is cyclic; in 4vbc that multiplicative table is *bit-inconsistent*.) 4vbc is further isomorphic to multiplicative table of the abstract Vierergruppe or Klein V_4 Group. 4vbc meets the five axioms required for an Abelian group under addition. 4vbc meets the three axioms required for a monoid group under multiplication. 4vbc meets the six axioms required for a ring to include left- and right-distributivity through 12 brute force combinations. 4vbc meets the four axioms required for a left R-module. Because the right and left R-modules are commutative, 4vbc is also an R-module.

The term above *bit-inconsistent* describes non-bivalent, vector spaces.

LET s = sinister (left-handed); d = dexter (right-handed). The x below is connective AND.

Table 4. **F**4 multiplicative table in 4vbc

<table>
<tr><td>s|d</td><td>s|d</td><td>s|d</td><td>s|d</td><td>s|d</td><td></td></tr>
<tr><td>x</td><td>0|0</td><td>0|1</td><td>1|0</td><td>1|1</td><td></td></tr>
<tr><td>0|0</td><td>0|0</td><td>0|0</td><td>0|0</td><td>0|0</td><td>line 1</td></tr>
<tr><td>0|1</td><td>0|0</td><td>0|1</td><td>1|0</td><td>1|1</td><td>line 2</td></tr>
<tr><td>1|0</td><td>0|0</td><td>1|0</td><td>1|1</td><td>0|1</td><td>line 3</td></tr>
<tr><td>1|1</td><td>0|0</td><td>1|1</td><td>0|1</td><td>1|0</td><td>line 4</td></tr>
</table>

Table 4 is bit inconsistent as shown in the left and right bits, respectively, in Table 5.

Table 5. **F**4 bit-inconsistent in 4vbc

ssss		dddd
0011		0011
0101		0101
0001	also	1110

Table 5 is derived many ways, as for example:

By the default of row major, for left bits (s) from Table 4:

↓

0 0 = 0 in line 1 and 0 in line 2
0 1 = 0 in line 1 and 1 in line 2
1 0 = 0 in line 3 and 0 in line 3
1 1 = 1 in line 3 and 0 in line 3

but also for right bits (d) from Table 4:

↓
0 0 = 1 in line 3 and 0 in line 3
0 1 = 1 in line 3 and 0 in line 3
1 0 = 1 in line 4 and 0 in line 4
1 1 = 0 in line 4 and 1 in line 4

4. Two model types on B4 with 2-tuple values

The two model types on B4 are named Model 1 (M1) and Model 2(M2). The logical values of the 2-tuple {00, 10, 01, 11} are described respectively as:

{**F**alse for contradiction; Contingent for falsity;
Non contingent for truthity; Tautology for proof} (M1)

and

{**U**nevaluated; Improper; Proper; Evaluated}. (M2)

The respective values of { F, C, N, T} in M1 are equivalent to { U, I, P, E} in M2. The designated *proof* value is T for tautology and E for evaluated.

5. Modal values on Ł4

Model 2.1 (M2.1) is equivalent to Model 1 (M1) but with { U, I, P, E} instead of { F, C, N, T}. M2.1 is included for completeness. M2 also contains the sub-models of M2.2 and M2.3. These are required for combinations of logical values in B4 to produce modal values in Ł4. The derivation is based on the up-and down-functors of Łukasiewicz below. (The symbols are & for AND and v for OR, and [] for necessity and <> for possibility.)

Łukasiewicz' Up-functor [p]

```
M1     []: { F,C,N,T} & C = { F,C,F,C};  M1     <>: { F,C,N,T} v N = { N,T,N,T}
M2.1   []: { U,I,P,E} & E = { U,I,P,E};  M2.1   <>: { U,I,P,E} v U = { U,I,P,E}
M2.2   []: { U,I,P,E} & U = { U,U,U,U};  M2.2   <>: { U,I,P,E} v E = { E,E,E,E}
M2.3.1 []: { U,I,P,E} & P = { U,U,P,P};  M2.3.1 <>: { U,I,P,E} v I = { I,I,E,E}
M2.3.2 []: { U,I,P,E} & I = { U,I,U,I};  M2.3.2 <>: { U,I,P,E} v P = { P,E,P,E}
```

Łukasiewicz' Down-functor [~p]

```
M1     []: { T,N,C,F} & C = { C,F,C,F};  M1     <>: { T,N,C,F} v N = { T,N,T,N}
M2.1   []: { E,P,I,U} & E = { E,P,I,U};  M2.1   <>: { E,P,I,U} v U = { E,P,I,U}
M2.2   []: { E,P,I,U} & U = { E,E,E,E};  M2.2   <>: { E,P,I,U} v E = { U,U,U,U}
M2.3.1 []: { E,P,I,U} & P = { E,E,I,I};  M2.3.1 <>: { E,P,I,U} v I = { E,E,I,I}
M2.3.2 []: { E,P,I,U} & I = { I,U,I,U};  M2.3.2 <>: { E,P,I,U} v P = { E,P,E,P}
```

The look up tables (LUTs) are stored in binary and decimal as

```
{ 00, 10, 01, 11} and
{  0,  3,  2,  1}
```

```
with substitution LUTs for:

     {  F,  C,  N,  T} and
     {  U,  I,  P,  E}.
```

Rule 1 states that for any expression falling within the scope of a modal operator, only M2.1 applies for all truth constructs of the expression.

Symbols are: & for AND; + for OR; # for [] necessity; and % for <> possibility. The modal results for #p, %p, #~p, and %~p of each model are below:

Row index	Column index Model	0 **p**	1 **#p**	2 **%p**	3 **#~p**	4 **%~p**
0	**B_4** # * 0 % + 2	0 3 2 1	0 3 0 3	2 1 2 1	3 0 3 0	1 2 1 2
1	**B_4** # & 01 % + 10	00 10 01 11	00 10 00 10	01 11 01 11	10 00 10 00	11 01 11 01
2	**M1** # & C % + N	F C N T	F C F C	N T N T	C F C F	T N T N
3	**M2.1** # & E % + U	U I P E	U I U I	P E P E	E P E P	I U I U
4	**M2.2** # & U % + E	U I P E	U U U U	E E E E	E E E E	U U U U
5	**M2.3.1** # & P % + I	U I P E	U U P P	I I E E	E E I I	P P U U
6	**M2.3.2** # & I % + P	U I P E	U I U I	P E P E	E P E P	I U I U

More compact LUTs are described as:

```
VŁ4:      M1  M2      ~VŁ4: ~M1 ~M2
          F   U              T   E
          C   I              N   P
          N   P              C   I
          T   E              F   U

   1        2.1  2.2  2.31 2.32
   # %      # %  # %  # %  # %
F. F C   U. U U  U E  U P  U I
C. F C   I. I I  U E  I E  U I
N. N T   P. P P  U E  U P  P E
T. N T   E. E E  U E  I E  P E
```

< Definitions of the five models.
\# Necessity, All or every;
% Possibility, One or some
(The equivalence of modal and quantified operators is derived in Section 9 below.)

The connectives are from standard logic and in one character as

{and, or, imply, equivalent} for {&, +, >, =};

and with the negated connectives as

{nand; nor; not imply; exclusive-or} for {\, -, <, @}.

The 16-valued look up truth tables are by four rows-major and presented horizontally.

```
1 & . F,F,F,F . F,C,F,C . F,F,N,N . F,C,N,T
1 \ . T,T,T,T . T,N,T,N . T,T,C,C . T,N,C,F
1 + . F,C,N,T . C,C,T,T . N,T,N,T . T,T,T,T
1 - . T,N,C,F . N,N,F,F . C,F,C,F . F,F,F,F
1 < . F,F,F,F . C,F,C,F . N,N,F,F . T,N,C,F
1 = . T,N,C,F . N,T,F,C . C,F,T,N . F,C,N,T
1 > . T,T,T,T . N,T,N,T . C,C,T,T . F,C,N,T
1 @ . F,C,N,T . C,F,T,N . N,T,F,C . T,N,C,F

2 & . U,U,U,U . U,I,U,I . U,U,P,P . U,I,P,E
2 \ . E,E,E,E . E,P,E,P . E,E,I,I . E,P,I,U
2 + . U,I,P,E . I,I,E,E . P,E,P,E . E,E,E,E
2 - . E,P,I,U . P,P,U,U . I,U,I,U . U,U,U,U
2 < . U,U,U,U . I,U,I,U . P,P,U,U . E,P,I,U
2 = . E,P,I,U . P,E,U,I . I,U,E,P . U,I,P,E
2 > . E,E,E,E . P,E,P,E . I,I,E,E . U,I,P,E
2 @ . U,I,P,E . I,U,E,P . P,E,U,I . E,P,I,U
```

6. Answer to an Ł4 objection

This proposition is supposed to be egregious to logic system $Ł_4$:

$$(\Diamond p \& \Diamond q) \rightarrow \Diamond (p \& q). \qquad (6.1.1)$$

If possibly the cat is alive and possibly the cat is dead, then
possibly both the cat is alive and the cat is dead. (6.1.0)

LET p, q: Schrödinger's cat is alive; Schrödinger's cat is dead

(%p&%q)>%(p&q) ; `TTTT TTTT TTTT TTTT` (6.1.2)

Assumptions: ((exists(p) & exists(q))).
Goals (exists(p&q)). Exhausted. (6.1.3)

Prover9 invalidates Eq. 6.1.3 to show $Ł_4$ is untenable as an alethic logic.

If we preload p=~q as the antecedent to Eq. 6.1.0, then:

If possibly the cat is alive is equivalent to Not (the cat is dead), then
if possibly the cat is alive and possibly the cat is dead, then
possibly both the cat is alive and the cat is dead. (6.2.0)

%(p=~q)>(%(p&q)>(%p&%q)) ; `TTTT TTTT TTTT TTTT` (6.2.2)

Assumptions: (exists(p<->-q)).
Goals: (exists(p)&exists(q))->(exists(p&q)).
Exhausted. (6.2.3)

Prover9 invalidates Eq. 6.2.3 to show $Ł_4$ is untenable as an alethic logic.

Remark 6.2.3: Eq. 6.2.3 shows Prover9 also does not distribute the existential quantifier.

We rewrite Eq. 6.2.1 using one variable and its negation as respectively *alive* and *not alive*:

$$(\Diamond p \& \Diamond \sim p) \rightarrow \Diamond (p \& \sim p). \qquad (6.3.1)$$

If possibly the cat is alive and possibly the cat is not alive, then possibly both the cat is alive and the cat is not alive. (6.3.0)

(%p&%~p)>%(p&~p) ;   TTTT TTTT TTTT TTTT   (6.3.2)

Assumptions: (exists(p)&-exists(p)).
Goals: (exists(p&-p).   Theorem.   (6.3.3)

Prover9 validates Eq. 6.3.3 to show Ł4 is tenable as an alethic logic.

We explain Eqs. 6.1.2, 6.2.2, and 6.3.2 as rendered as tautologous in Meth8/VŁ4, but 6.1.3 as exhausted in Prover9 in this way. For more than one variable, the vector space for arity with Prover9 diverges from the bivalance inherent in $VŁ_4$, in which modal operators and quantifiers are distributive. This speaks to Meth8/VŁ4, based on the *corrected* modern Square of Opposition for an exact bivalent system, as opposed to Prover9, based on the uncorrected modern Square of Opposition for an inexact probabilistic vector space.

Remark 6.3.2: Meth8/VŁ4 also distinguishes between Eqs. 2.2 and 3.2 by protasis and apodosis as:

%p&%q ;   CCCT CCCT CCCT CCCT   (6.1.2.1.2)
%(p&q)=(p=p) ;   CCCT CCCT CCCT CCCT   (6.1.2.2.2)

and

%p&%~p ;   CCCC CCCC CCCC CCCC   (6.3.2.1.2)
%(p&~p)=(p=p) ;   CCCC CCCC CCCC CCCC   (6.3.2.2.2)

7. Corrected Square of Opposition

We include the Square of Opposition as corrected by Meth8 and confirmation of the Łukasiewicz Square of Opposition via logic VŁ4, including the Seuren Cube of Opposition which vindicates the mistaken criticism of it (although it still was *not* tautologous).

7.1. Square of Opposition Meth8 corrected

The modern revision of the square of opposition is not validated as tautologous by the Meth8 logic model checker, as based on system variant VŁ4. Consequently we redefine the square so that it is validated as tautologous my Meth8. Instead of definientia using implication for universal terms or conjunction for existential terms, we adopt the equivalent connective for all terms. The modal modifiers necessity and possibility map quantifiers as applying to the entire terms rather than to the antecedent within the terms.

The Meth8 symbols here are: ~ Negation ; \ Nand ; > Imply ; + Or ; # modal necessity for universal quantifier ; % modal possibility for existential quantifier ; ? unspecified connective.

Sources Type	Definientia	* Modern Revision Script	Valid as	** Meth8 Correction Script	Valid as
Corner	A	#s> p		#(s= p)	
	E	#s>~p		#(s=~p)	
	I	%s&p		%(s= p)	
	O	%s&~p		%(s=~p)	
Contraries	AE	(#s>p) + (#s>~p)	A + E	#(s= p) \ #(s=~p)	A \ E
Subalterns	AI	(#s>p) ? (%s&p)		#(s= p) > %(s= p)	A > I
Contradictories	AO	(#s>p) + (%s&~p)	A + O	#(s= p) \ %(s=~p)	A \ O
Contradictories	EI	(#s>~p) + (%s&p)	E + I	#(s=~p) \ %(s= p)	E \ I
Subalterns	EO	(#s>~p) ? (%s&~p)		#(s=~p) > %(s=~p)	E > O
Subcontraries	IO	(%s&p) \ (%s&~p)	I \ O	%(s= p) + %(s=~p)	I + O

* The quantifier may refer to the entire term as #(p=q) or to the antecedent of the term as (#p=q). In Meth8 there is a difference. We adopt the latter because it returns more validated connectives. For example from the traditional square: #(A?E), #(I?O) versus (A+E), (I\O).

The modern revision of the square of opposition is not validated as tautologous by the Meth8 logic checker in five models for all expressions. This leads us to consider that any logic system based on the square of opposition is spurious. What follows then is that a first order predicate logic based on the square of opposition is now suspicious.

** The Meth8 validated square of opposition redefines A, E, I, O to match the words more clearly. For example on A, "All S is P" is mapped as "#(s=p)", not as in the note above with "#s=p" because the connective of equivalence is stricter than that of implication and consistent for all definiens. By changing the connective in the term from implication or conjunction to equivalence makes the Meth8 validated square of opposition suitable as a basis for other logics such as first order predicate logic.

We note the validating connectives for the edges on the square are: \ Nand for the Contraries and Contradictories; > Imply for the Subalterns; and + Or for the Subcontraries.

7.2. Confirmation of the Łukasiewicz Square of Opposition via logic VŁ4

We evaluate the existential import of the Revised Modern Square of Opposition. We confirm that the Łukasiewicz syllogistic was intended to apply to *all* terms. What follows is that Aristotle was mistaken in his mapping of vertices, which we correct and show fidelity to Aristotle's intentions. We also evaluate the Cube of Opposition of Seuren. Two final claims are not tautologous, hence refuting the Cube, which also contradict criticism of Seuren that was not based on those claims.

See: Read, S. (2015). Aristotle and Łukasiewicz on Existential Import.
st-andrews.ac.uk/~slr/Existential_import.pdf slr@st-andrews.ac.uk

We map vertices of the first Square of Opposition on page 4 with its words below.

(A)	Every S is P.	#(s= p)=(p=p) ;	N**F**N**F** N**F**N**F** **F**N**F**N **F**N**F**N	(7.2.0.1.2)
(E)	No S is P.	#(s=~p)=(p=p) ;	**F**N**F**N **F**N**F**N N**F**N**F** N**F**N**F**	(7.2.0.3.2)
(I)	Some S is P.	%(s= p)=(p=p) ;	TCTC TCTC CTCT CTCT	(7.2.0.5.2)
(O)	Not every S is P.	%(~s=p)=(p=p) ;	CTCT CTCT TCTC TCTC	(7.2.0.7.2)

Remark 7.2.0: The above is from our *revised* Modern Square of Opposition as in Section 7.1.

We map the relations which Aristotle accepts as preserved here.

A- and E-propositions are contrary (cannot both be true) [(A)=T & (E)=T] (7.2.1.1.1)

(#(s= p)=(p=p))&(#(s=~p)=(p=p)) ; **FFFF FFFF FFFF FFFF** (7.2.1.1.2)

and I- and O-propositions are subcontrary (cannot both be false)
[(I)=**F** & (O)=**F**] (7.2.1.2.1)

(%(s= p)=(p@p))&(%(s=~p)=(p@p)) ; **FFFF FFFF FFFF FFFF** (7.2.1.2.2)

A- and O-propositions are contradictories, [(A)&(O)] (7.2.2.1.1)

#(s= p)&%(s=~p) ; **FFFF FFFF FFFF FFFF** (7.2.2.1.2)

as are I- and E-propositions [(I) & (E)] (7.2.2.2.1)

%(s= p)&#(s=~p) ; **FFFF FFFF FFFF FFFF** (7.2.2.2.2)

A-propositions imply their subaltern I-proposition, [(A) > (I)] (7.2.3.1.1)

#(s= p)>%(s= p) ; TTTT TTTT TTTT TTTT (7.2.3.1.2)

and E-propositions their subaltern O-proposition [(E) > (O)] (7.2.3.2.1)

#(s=~p)>%(s=~p) ; TTTT TTTT TTTT TTTT (7.2.3.2.2)

I- propositions convert simply ‘Some *S* is *P* ’ implies ‘Some *P* is *S*’, (7.2.4.1.1)

%(s= p)>%(p= s) ; TTTT TTTT TTTT TTTT (7.2.4.1.2)

and E-propositions ‘No *S* is *P* ’ implies ‘No *P* is *S*’ (7.2.4.2.1)

#(~s=p)>#(~p=s) ; TTTT TTTT TTTT TTTT (7.2.4.2.2)

A-propositions convert accidentally (‘Every *S* is *P* ’ implies ‘Some *P* is *S*’) (7.2.5.1.1)

#(s= p)>%(p= s) ; TTTT TTTT TTTT TTTT (7.2.5.1.2)

and O-propositions don’t convert at all.
[Some S is not P implies Every P is not S.] (7.2.5.2.1)

%(s=~p)>#(p=~s) ; NNNN NNNN NNNN NNNN (7.2.5.2.2)

We present these six equations for the six directed rays in the Square, as in Section 7.1.

(A\E) #(s= p) \ #(s=~p) ; TTTT TTTT TTTT TTTT (7.2.6.1.2)
(A>I) #(s= p) > %(s= p) ; TTTT TTTT TTTT TTTT (7.2.6.2.2)
(A\O) #(s= p) \ %(s=~p) ; TTTT TTTT TTTT TTTT (7.2.6.3.2)
(E\I) #(s=~p) \ %(s= p) ; TTTT TTTT TTTT TTTT (7.2.6.4.2)
(E>O) #(s=~p) > %(s=~p) ; TTTT TTTT TTTT TTTT (7.2.6.5.2)
(I+O) %(s= p) + %(s=~p) ; TTTT TTTT TTTT TTTT (7.2.6.6.2)

Remark 7.2.6: The new connective distribution is as follows with count. The mappings above allow for replication and confirmation of the 24-syllogisms and with our claim of a minor correction each to Modus Camestros and Modus Cesare.

(1) Contraries Not And (\);
(1) Subcontraries Or (+);
(2) Subalterns Imply (>); and
(2) Contradictories Not And (\)

We conclude that Łukasiewicz was not mistaken in his rendition of the Square of Opposition.

We now turn to the criticism of the Cube of Opposition of Seuren to map and interleave the additional vertices from the diagram on page 8. While * marks predicate negation with the term "-P", we use $ to mark copula negation with the term "not P", and mark the negation of $ using !.

(A) Every S is P. #(s= p) =(p=p) ;
N**F**N**F** N**F**N**F** **F**N**F**N **F**N**F**N (7.2.7.1.1)
(A*) Every S is not-P. ~(#(s=p)=(p=p))=(p=p) ;
as Not (Every S is P.)
CTCT CTCT TCTC TCTC (7.2.7.1.2)

(A$) Every S is not P. #(s=~p)=(p=p) ;
FN**F**N **F**N**F**N N**F**N**F** N**F**N**F** (7.2.7.1.3)

(A!) Not (Every S is not P.) ~(#(s=~p)=(p=p))=(p=p) ;
TCTC TCTC CTCT CTCT (7.2.7.1.4)

(E) No S is P. #(s=~p)=(p=p) ;
FN**F**N **F**N**F**N N**F**N**F** N**F**N**F** (7.2.7.2.1)

(E*) No S is not-P. ~(#(s=~p)=(p=p))=(p=p) ;
as Not (No S is P.) TCTC TCTC CTCT CTCT (7.2.7.2.2)

(E$) No S is not P. #(~s=~p)=(p=p) ;
N**F**N**F** N**F**N**F** **F**N**F**N **F**N**F**N (7.2.7.2.3)

(E!) Not (No S is not P.) ~(#(~s=~p)=(p=p))=(p=p) ;
CTCT CTCT TCTC TCTC (7.2.7.2.4)

(I) Some S is P. %(s= p)=(p=p) ;
TCTC TCTC CTCT CTCT (7.2.7.3.1)

(I*) Some S is not-P. ~(%(s= p)=(p=p))=(p=p) ;
as Not (Some S is P). **F**N**F**N **F**N**F**N N**F**N**F** N**F**N**F** (7.2.7.3.2)

(I$) Some S is not P. %(s=~p)=(p=p) ;
CTCT CTCT TCTC TCTC (7.2.7.3.3)

(I!) Not (Some S is not P.) ~(%(s=~p)=(p=p))=(p=p) ;
N**F**N**F** N**F**N**F** **F**N**F**N **F**N**F**N (7.2.7.3.4)

(O) Not every S is P. %(~s=p)=(p=p) ;
CTCT CTCT TCTC TCTC (7.2.7.4.1)

(O*) Not every S is not-P. ~(%(~s=p)=(p=p))=(p=p) ;
as Not(Not every S is P.) N**F**N**F** N**F**N**F** **F**N**F**N **F**N**F**N (7.2.7.4.2)

(O$) Not every S is not P. %(~s=~p)=(p=p) ;
TCTC TCTC CTCT CTCT (7.2.7.4.3)

(O!) Not (Not every S is not P.) ~(%(~s=~p)=(p=p))=(p=p) ;
FN**F**N **F**N**F**N N**F**N**F** N**F**N**F** (7.2.7.4.4)

The following are supposed to hold:

~I* = *E: ~(~(%(s= p)=(p=p))=(p=p)) = (~(#(s=~p)=(p=p))=(p=p)) ;
TTTT TTTT TTTT TTTT (7.2.8.1.1)

~A* = O*: ~(~(#(s=p)=(p=p))=(p=p)) = (~(%(~s=p)=(p=p))=(p=p)) ;
TTTT TTTT TTTT TTTT (7.2.8.1.2)

A* > E: (~(#(s=p)=(p=p))=(p=p)) > (#(s=~p)=(p=p)) ;
NNNN NNNN NNNN NNNN (7.2.9.1.1)

A > E*: (#(s= p) =(p=p)) > (~(#(s=~p)=(p=p))=(p=p)) ;
TTTT TTTT TTTT TTTT (7.2.9.1.2)

I > O*: (%(s= p)=(p=p)) > (~(%(~s=p)=(p=p))=(p=p)) ;
NNNN NNNN NNNN NNNN (7.2.9.1.3)

I* > O: (~(%(s= p)=(p=p))=(p=p)) > (%(~s=p)=(p=p)) ;
TTTT TTTT TTTT TTTT (7.2.9.1.4)

Eqs. 7.2.9.1.1 (A* > E) and 7.2.9.1.3 (I > O*) are *not* tautologous, albeit truthities. This means that the final claims of Seuren's Cube of Opposition are mistaken, but also that the criticism of Seuren as based not on those claims is also mistaken.

8. Corrected syllogisms

The original Square of Opposition produced four combinations for each corner A, I, E, O for 4 ^ 4 = 256 syllogisms. Medieval scholars determined 24 of the 256 syllogisms were tautologous deductions. Of those, 9 were made tautologous but only after additional *known* assumptions were applied as fix ups. Meth8/VŁ4 found tautologous none of the 24 syllogisms *before* fix ups. Meth8 also *discovered* correct additional assumptions to render the other 15 syllogisms found as tautologous. The fix ups in bold were verified independently by Prover9 (2007).

From: en.wikipedia.org/wiki/Syllogism

LET q, r, s: M, P, S.

Original syllogisms in Meth8 script:

Code	**Name**	**Assumptions: 1, 2,**	**3**	**Conclusion**	**Test**	**Comments**
AAA-1	Modus Barbara	((#q&r)&(#s&q))		>(#s&r)	tautologous	
AAI-1	Modus Barbari	(((#q&r)&(#s&q))	&%s)	>(%s&r)		* not needed
		((#q&r)&(#s&q))		>(%s&r)	tautologous	
AAI-4	Modus Bamalip	(((#r&q)&(#q&s))	&%r	>(%s&r)		* not needed
		((#r&q)&(#q&s))		>(%s&r)	tautologous	
EAE-1	Modus Celarent	((~q&r)&(#s&q))		>(~s&r)	tautologous	
EAE-2	**Modus Cesare**	((~r&q)&(#s&q))		>(~s&r)	~ tautologous	* Mistake
		(((~r&q)&(#s&q))	**&%r)**	>(~s&r)	tautologous	*** Meth8 fix**
AEE-2	Modus Camestres	((#r&q)&(~s&q))		>(~s&r)	tautologous	
AEE-4	Modus Calemes	((#r&q)&(~q&s))		>(~s&r)	tautologous	
EAO-1	Modus Celaront	(((~q&r)&(#s&q))	&%s)	>(~s&r)		* not needed
		((~q&r)&(#s&q))		>(~s&r)	tautologous	
EAO-2	Modus Cesaro	(((~r&q)&(#s&q))	&%s)	>(%s&~r)		* not needed
		((~r&q)&(#s&q))		>(%s&~r)	tautologous	

Code	Name	Assumptions: 1, 2,	3	Conclusion	Test	Comments
AEO-2	**Modus Camestros**	(((#r&q)&(~s&q))	**&%s)**	>(%s&~r)	tautologous	* **needed**
		((#r&q)&(~s&q))		>(%s&~r)	~ tautologous	* Mistake
AEO-4	Modus Calemos	(((#r&q)&(~q&s))	&%s)	>(%s&~r)		* not needed
		((#r&q)&(~q&s))		>(%s&~r)	tautologous	
AII-1	Modus Darii	((#q&r)&(%s&q))		>(%s&r)	tautologous	
AII-3	Modus Datisi	((#q&r)&(%q&s))		>(%s&r)	tautologous	
IAI-3	Modus Disamis	((%q&r)&(#q&s))		>(%s&r)	tautologous	
IAI-4	Modus Diamatis	((%r&q)&(#q&s))		>(%s&r)	tautologous	
EIO-1	Modus Ferio	((~q&r)&(%s&q))		>(%s&~r)	tautologous	
EIO-2	Modus Festino	((~r&q)&(%s&q))		>(%s&~r)	tautologous	
EIO-3	Modus Ferison	((~q&r)&(%q&s))		>(%s&r)	tautologous	
EIO-4	Modus Fresison	((~r&q)&(%q&s))		>(%q&~r)	tautologous	
AOO-2	Modus Baroco	((#r&q)&(%s&~q))		>(%s&~r)	tautologous	
OAO-3	Modus Bocardo	((%q&~r)&(#q&s))		>(%s&~r)	tautologous	
AAI-3	Modus Darapti	(((#q&r)&(#q&s))	&%q)	>(%s&r)		* not needed
		((#q&r)&(#q&s))		>(%s&r)	tautologous	
EAO-3	Modus Felapton	(((~q&r)&(#q&s))	&%q)	>(%s&~r)		* not needed
		((~q&r)&(#q&s))		>(%s&~r)	tautologous	
EAO-4	Modus Fesapo	(((~r&q)&(#q&s))	&%q)	>(%s&~r)		* not needed
		((~r&q)&(#q&s))		>(%s&~r)	tautologous	

9. Quantifiers equivalent to modal operators

The rationale for rendering quantifiers as modal operators in Meth8/VŁ4 has arguments from reproducability of formulas for vertices and edges in modal logic for the Square of Opposition in Section 7, reproducability of evaluating syllogisms as tautologous (with two corrections) in Section 8, and from satisfiability (contra Kuhn) below.

From: Kuhn, S.T. (1979). "Quantifiers as modal operators". Studia Logica. 39.2-3/80: 147. faculty.georgetown.edu/kuhns/supp_files/quantifiers.pdf

"Either [with Montague's approach as first order models or with Prior's approach as "sequences of individuals"], there is a problem. The atomic formulas of predicate logic cannot all be treated as atoms in the modal language. If we regard *Pxy* and *Pyx*, for example, as distinct sentence letters of the modal language then

$$\exists x\ \exists y\ Pxy\ \&\ -\exists \boldsymbol{x}\ \exists \boldsymbol{y}\ Pyx \tag{9.1.1}$$

LET p, q, r: p, x, y

(p&(%q&%r))&~(p&(%r&%q)) ;   **FFFF FFFF FFFF FFFF**   (9.1.2)

will be satisfiable.

Remark 9.1.2: Eq. 9.1.2 is *not* tautologous and a contradiction.

If we regard them as identical sentence letters then

$$\exists x\ \exists y\ (Pxy\ \&\ -Pyx) \qquad (9.2.1)$$

((p&(%q&%r))&~(p&(%r&%q))) = (p=p) ; **FFFF FFFF FFFF FFFF**   (9.2.2)

will be unsatisfiable."

Remark 9.2.2: Eq. 9.2.2 is *not* tautologous, is a contradiction, and is identical to Eq. 9.1.2.

Because Eqs. 9.1.2 and 9.2.2 are identical as contradictions, so that rendition of the satisfiability for quantifiers to modal operators is contradictory. For Meth8/VŁ4 to show that the contradictions are equivalent implies Meth8/VŁ4 is consistent in finding those definitions as equivalent.

What follows is that there is no reason to rely on

> "the variable-free formulations of logic by Tarski, Bernays, Halmos, Nolin and Quine ... [for] the effect of arbitrary permutations and identifications of the variables occurring in a formula."

We further show that Eq. 9.1.1 (or 9.2.1) is *not* a fragment contained within the universally quantified variables of p&(#q&#r):   (9.3.1)

((p&(%q&%r))&~(p&(%r&%q)))<(#q&#r) ;
FFFF FFFF FFFF FFFF   (9.3.2)

10. Meth8/VŁ4 implementation

The Meth8 script uses literals and connectives in one-character. Propositions are p-z, and theorems are A-B. The connectives for {and, or, imply, equivalent} are {&, +, >, =}. The negated connectives for {nand; nor; not imply; exclusive-or} are {\, -, <, @}. The operators for {not; possibility ◇∃; necessity □∀} are {~, %, #}. Expressions are adopted for clarity as: (p=p) for tautologous; (p@p) for contradiction; and (x<y) for x∈y. The expression x≤y as "x less than or equal to y" is rendered in the negative as ~(y<x) or as (~x>~y). Variables are defined as:

Definition	Axiom	Symbol	Name	Meaning	Binary	Decimal
1	p=p	T	tautology	proof	11	3
2	p@p	F	contradiction	absurdum	00	0
3	%p>#p	N	non-contingency	truthity	01	1
4	%p<#p	C	contingency	falsity	10	2

Note the meaning of (%p>#p): a possibility of p implies the necessity of p; and some p implies all p. In other words, if a possibility of p then the necessity of p; and if some p then all p.

This shows equivalence of respective modal operators and quantified operators as in Section 9 above.

Meth8 contains recent advances in parsing technology named sliding window. It is written in 7,100 lines of industrial grade code in True BASIC, the educator's language, and ANSI standard. The novel installation wrapper is for one user per seat per CPU, and licensed by number of logical LUT accesses at run time. The is no internet access, and no asymmetric key encryption. Hence Meth8 is ITAR compliant and exportable.

Meth8 use variables for 4 propositions, 4 theorems, and 11 propositions. The size of truth tables are respectively for 16-, 256-, and 2048- truth values, using recent advances in look up table indexing. In RAM look up tables (LUTs) are for 4 theorems (16 result tables), 4 propositional variables (1 result table), 11 propositional variables (128 result tables). Larger numbers of variables scale via LUTs on external media.

11. Notable refutations

We evaluate 419 artifacts in 2286 assertions to confirm 445 as tautology and 1841 as *not* (80.5%). We use Meth8, a modal logic checker in five models. The mapping of formulas in Meth8 script was performed by hand, checked, and tested for accuracy of intent.

We assume the method and apparatus of Meth8/VŁ4 with Tautology as the designated proof value, **F** as contradiction, N as truthity (non-contingency), and C as falsity (contingency). The 16-valued truth table is row-major and horizontal, or repeating fragments of 128-tables, sometimes with table counts, for more variables. (See ersatz-systems.com.)

LET ~ Not, ¬ ; + Or, ∨, ∪; - Not Or; & And, ∧, ∩; \ Not And;
> Imply, greater than, →, ↦, ≻, ⊃, ⊢, ⊨, ↠; < Not Imply, less than, ∈, ≺, ⊂, ⊬, ⊭, ↞;
= Equivalent, ≡, :=, ⟺, ↔, ≜, ≈; @ Not Equivalent, ≠;
% possibility, for one or some, ∃, ◊, M; # necessity, for every or all, ∀, □, L;
(z=z) T as tautology, ⊤, ordinal 3; (z@z) **F** as contradiction, ∅, Null, ⊥, zero;
(%z<#z) C as contingency, Δ, ordinal 1; (%z>#z) N as non-contingency, ∇, ordinal 2;
~(y < x) (x ≤ y), (x ⊆y); (A=B) (A~B).

Note: For clarity we usually distribute quantifiers on each variable as designated.

Seven refutations are discovered as non-tautologous fragments of VŁ4.

11.1. Refutation of Bell's inequality

From: Maccone, L. (2013). "A simple proof of Bell’s inequality". arxiv.org/pdf/1212.5214.pdf

The summation of the respective probabilities for q equivalent to r, r equivalent to s, and q equivalent to s is equal to or greater than one, and hence is equivalent to a theorem. (11.1.1.1)

~((((p&q)=(p&r)) + (((p&r)=(p&s)) + ((p&q)=(p&s)))) < (%p>#p))
= (p=p) ; NNNN NNNN NNNN NNNN (11.1.1.2)

Remark 11.1.1.1: For further qualification to strengthen Eq. 11.1.1.1, we rewrite it as:

If the respective probabilities for q, r, s are equivalent to and equal to one, then the summation of the respective probabilities for q equivalent to r, r equivalent to s, and q equivalent to s is equal to or greater than one. (11.1.2.1)

(((p&q)=((p&r)=(p&s)))=(%p>#p)) >
~((((p&q)=(p&r)) + (((p&r)=(p&s)) + ((p&q)=(p&s)))) < (%p>#p)) ;
NNNT TTNN TTNN NNTT (11.1.2.2)

Eqs. 11.1.1.2 and 11.1.2.2 as rendered are *not* tautologous. Hence, Bell's inequality as Eqs. 11.1.1.1 or 11.1.2.1 is refuted.

11.2. Refutation of the Gödel-Löb axiom

This example replicates the proof for provability logic of the Gödel-Löb axiom GL as

$\Box(\Box p\rightarrow p)\rightarrow\Box p$. (11.2.1.1)

If p is "*choice*", this transcribes in words to:
"The necessity of *choice*, as always implying *a choice*, implies always *a choice*."
(11.2.1.0)

#(#p>p)>#p ; CTCT CTCT CTCT CTCT (11.2.1.2)

To coerce the GL axiom to be a tautology, the expression is rewritten as

$\Box(\Box p\rightarrow p)\leftrightarrow(p\vee\neg p)$, TTTT TTTT TTTT TTTT (11.2.2.1)

in words: "The necessity of *choice*, as always implying *a choice*, is equivalent to always *a choice* or *no choice*." (11.2.2.0)

A simpler rendition of a tautologous GL-type axiom is either

$\Box(\Box\neg p\rightarrow p)\leftrightarrow\Box p$, or (11.2.3.1)

$\Box(\Box p\rightarrow\neg p)\leftrightarrow\Box\neg p$ (11.2.4.1)

as respectively in words: "The necessity of *no choice*, as always implying *a choice*, is equivalent to always *a choice*."; or (11.2.3.0)

"The necessity of *choice*, as always implying *no choice*, is equivalent to always *no choice*." (11.2.4.0)

Remark 11.2: If GL fails, then so also does Zermelo-Fraenkel set theory and the axiom of choice (ZFC) as the basis of modern mathematics.

11.3. Refutation of the Löb theorem and Gödel incompleteness by substitution of contradiction

From: Gross, J. et al. (2016). Löb's Theorem. jasongross.github.io/lob-paper/nightly/lob.pdf
jgross@mit.edu, jack@gallabytes.com, benya@intelligence.org

This, in a nutshell, is Löb's theorem: to prove *X*, it suffices to prove that *X* is true whenever *X* is provable. If we let $\Box X$ denote the assertion "*X* is provable," then, symbolically, Löb's theorem becomes: $\Box(\Box X\rightarrow X)\rightarrow\Box X$. (11.3.1.1)

LET p: *X*.

#(#p>p)>#p ; `CTCT CTCT CTCT CTCT` (11.3.1.2)

Remark 11.3.1.2: Eq 11.3.1.2 as rendered is *not* tautologous, thus refuting Löb's theorem.

Note that Gödel's incompleteness theorem follows trivially from Löb's theorem: by instantiating *X* with a contradiction [⊥], we can see that it's impossible for provability to imply truth for propositions which are not already true. (11.3.2.1)

#(#(p@p)>(p@p))>#(p@p) ; `CCCC CCCC CCCC CCCC` (11.3.2.2)

Remark 11.3.2.2: Eq. 11.3.2.2, rendered as Eq. 11.3.1.2 with p substituted by (p@p), is *not* tautologous but consistently falsity as `C` for contingency. Hence Gödel's incompleteness theorem, as following trivially, is also refuted.

This means that the type of Löb's theorem becomes either $\Box(\Box X\rightarrow X)\rightarrow\Box X$ [Eq. 11.3.1.1], which is not strictly positive, or $\Box(X\rightarrow X)\rightarrow\Box X$, (11.3.3.1)

#(p>p)>#p ; CTCT CTCT CTCT CTCT (11.3.3.2)

which, on interpretation, must be filled with a general fixpoint operator. Such an operator is well-known to be inconsistent.

> **Remark on Fn. 2:** Eq. 11.3.3.2 as rendered produces the same truth table result as Eq. 11.3.1.2 and as another trivial refutation.

11.4. Refutation of the Löwenheim–Skolem theorem

From: en.wikipedia.org/wiki/Löwenheim–Skolem_theorem

In its general form, the Löwenheim–Skolem theorem states that for every signature σ, every infinite σ-structure *M,* and every infinite cardinal number κ ≥ |σ|, (11.4.1.1)

LET p, q, r, s: κ, M, N, σ ; (p@p) 0, zero;
(s>(p@p)) |σ|; (q>(p@p)) |*M*|; (r>(p@p)) |*N*|; ~(p<q) (p≥q).

#(s&((s&q)&(~(p<(s>(p@p)))))) ; FFFF FFFF FFNF FFNF (11.4.1.2)

there is a σ-structure *N* (11.4.2.1)

%(s&r) ; CCCC CCCC CCCC TTTT (11.4.2.2)

such that |*N*| = κ and
if κ < |*M*| then *N* is an elementary substructure of *M*; [and/or]
if κ > |*M*| then *N* is an elementary extension of *M*. (11.4.3.1)

(((r>(p@p))=p)&(((p<(q>(p@p)))>(q<r)) [&,+] ((p>(q>(p@p)))>(q>r)))) ;
FTFT TFTF FTFT TFTF (11.4.3.2)

Eq. 11.4.1.1 implies 11.4.2.1. (11.4.4.1)

#(s&((s&q)&~(p<(s>(p@p)))))>%(s&r) ; TTTT TTTT TTCT TTTT (11.4.4.2)

Eq. (11.4.4.1 = 11.4.1.1 implies 11.4.2.1) implies 11.4.3.1. (11.4.5.1)

(#(s&((s&q)&~(p<(s>(p@p)))))>%(s&r)) >
(((r>(p@p))=p)&(((p<(q>(p@p)))>(q<r))+((p>(q>(p@p)))>(q>r)))) ;
FTFT TFTF FTFT TFTF (11.4.5.2)

Eq. 11.4.1.2 as rendered is *not* tautologous, and not contradictory. Eq. 11.4.11.4.4.1 is *not* tautologous due to one C falsity value. Eq. 11.4.4.2 is *not* tautologous, and the same result table as Eq. 11.4.5.2. This means the Löwenheim–Skolem theorem is refuted.

11.5. Refutation of Peirce's abduction and induction, and confirmation of deduction

From: iep.utm.edu/peir-log/

C.S. Peirce originally defined the three forms of inference in logic as:

Abduction: (Q is S) and (Q is P) imply (S is P) (11.5.1.1.1)

LET p, q, s: P, Q, S.

((q=s)&(q=p))>(s=p) ; TTTT TTTT TTTT TTTT (11.5.1.1.2)

Induction: (S is Q) and (P is Q) imply (S is P) (11.5.2.1.1)

((s=q)&(p=q))>(s=p) ; TTTT TTTT TTTT TTTT (11.5.2.1.2)

Deduction: (S is Q) and (Q is P) imply (S is P) (11.5.3.1.1)

((s=q)&(q=p))>(s=p) ; TTTT TTTT TTTT TTTT (11.5.3.1.2)

Peirce described Eqs. 11.5.1 - 11.5.3 as inversions of the same.

Remark 11.5.1.1.1: If the word "is" is taken to mean the word "implies" then the connective = is replaced with the connective > below.

Abduction: (Q implies S) and (Q implies P) imply (S implies P) (11.5.1.2.1)

((q>s)&(q>p))>(s>p) ; TTTT TTTT **F**TTT **F**TTT (11.5.1.2.2)

Induction: (S implies Q) and (P implies Q) imply (S implies P) (11.5.2.2.1)

((s>q)&(p>q))>(s>p) ; TTTT TTTT TT**F**T TT**F**T (11.5.2.2.2)

Deduction: (S implies Q) and (Q implies P) imply (S implies P) (11.5.3.2.1)

((s>q)&(q>p))>(s>p) ; TTTT TTTT TTTT TTTT (11.5.3.2.2)

Eqs. 11.5.1.2.2 - 11.5.2.2.2 as rendered for abduction and induction are *not* tautologous, but Eq. 11.5.3.2.2 is tautologous. This means that abduction and induction are not inversions of deduction, leaving deduction as the only form of tautologous inference in logic.

11.6. Erwin Schrödinger's cat thought-experiment

From: en.wikipedia.org/wiki/Schrödinger's_cat

If the monitor is tautologous, that is not activated, along with the box, cat, and poison apparatus in place, then there is no death. (11.6.H_0.1)

LET p, q, r, s t: box, cat, poison, monitor, death

((s=s)&((p&q)&r)) > ~t ; **FF**N**F** **FF**N**F** **FF**N**F** **FF**N**F** (11.6.H_0.2)

If the monitor is contradictory, that is activated, along with the box, cat, and poison apparatus in place, then there is death. (11.6.H_1.1)

((s@s)&((p&q)&r)) > t ; TTTT TTTT TTTT TTTT (11.6.H_1.2)

Hence when opening the box at any time, the cat is either still alive or dead, but not "entangled" as both dead and alive (a contradiction). Therefore the experiment is *not* a paradox from Eq. 11.6.H_1.2 but a contradiction.

11.7. Refutation of the ZF axiom of the empty set

From: en.wikipedia.org/wiki/Axiom_of_empty_set

In the formal language of the Zermelo–Fraenkel axioms, the axiom reads ... in words:

There is a set such that no element is a member of it: $\exists x \forall y \neg(y \in x)$ (11.7.1.0)

We distribute the quantifiers to the respective variables as:
Not(necessarily y as a member of possibly x). (11.7.1.1)

(#q>%p) = (p=p) ; TTCT TTCT TTCT TTCT (11.7.1.2)

From: plato.stanford.edu/entries/set-theory/ZF.html by Joan Bagaria (joan.bagaria@icrea.cat)

The null set, equivalent to the empty set, is defined as: $\exists x \neg \exists y(y \in x)$ (11.7.2.0)

We distribute the quantifiers to the respective variables as:
Not(possibly y as a member of possibly x). (11.7.2.1)

(%q>%p) = (p=p) ; TTCT TTCT TTCT TTCT (11.7.2.2)

Eqs. 11.7.1.2 and 11.7.2.2, with the same truth table result, are *not* tautologous. This refutes the ZF axiom of the empty set.

12. Conclusion

This paper:

1. Introduces the bivalent logic B4;
2. Adopts a four-valued system based on the 2-tuple in two models M1 and M2;
3. Derives modal values in Ł4;

4. Answers an objection by trivial proof;
5. Corrects the Square of Opposition with modal equations for lines and angles;
6. Confirms the 24-syllogisms by modifying two;
7. Shows respective quantified and modal operators are equivalent;
8. Describes the Meth8 software implementation of VŁ4;
9. Tests 2200 assertions for a refutation rate of 80%;
10. Provides seven worked examples of refutation;
11. Classifies refutations as non tautologous fragments of VŁ4; and
12. Concludes that VŁ4 is a universal logic.

13. Future research

Continued testing of artifacts burgeons the table of contents of results, with details usually as one or two paged papers. The mapping of sentences into script for Meth8/VŁ4 could be automated for repetitive testing, however there is no substitute for hand-coding as best-by-test for catching most errors of symbolic assignments. The parsing component of Meth8 is mature enough to rapidly detect incorrect grammatical for the input script. For Meth8 an immediate further application is mapping sentences of natural language into logical formulas, so a semi-automation of that linguistic process is proceeding.

Acknowledgments

Thanks are due for: helpful discussion from G. Goodwin; and useful comments from L. Humberstone and A. Jovanovich.

References [other than those embedded in the text]

James, C. (2010). Proof of four valued bit code (4vbc) as a group, ring, and module. Third world congress and school on universal logic. Estoril, Portugal.

James, C. (2015a). Theorem prover Meth8 applies four valued Boolean logic for modal interpretation. First World Conference: Analogy. Beneméita Universidad Autónoma de Puebla, Mexico, November 4-6, 2015, Handbook, ISBN 978-83-65273-01-1, 50-51.

James, C. (2017). Meth8 on Karl Popper Ex(Gx). Second world congress on religion and logic. University of Warsaw.

Łukasiewicz, J. (1953). A system of modal logic. The Journal of Computing Systems, 1, 111-149.

Łukasiewicz, J. (1954). On a controversial problem of Aristotle's modal syllogistic. Dominican Studies. 7, 114-128.

Łukasiewicz, J. (1957). Aristotle's syllogistic logic (Second Edition). Clarendon Press, Chapter VII.