Quantum-Implication-Based Equivalents of the Orthomodularity Law in Quantum Logic: Part 3

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA
email: jhorner@cybermesa.com

Abstract

The optimization of quantum computing circuitry and compilers at some level must be expressed in terms of quantum-mechanical behaviors and operations. In much the same way that the structure of conventional propositional (Boolean) logic (BL) is the logic of the description of the behavior of classical physical systems and is isomorphic to a Boolean algebra (BA), so also the algebra, C(H), of closed linear subspaces of (equivalently, the system of linear operators on (observables in)) a Hilbert space is a logic of the descriptions of quantum mechanical systems and is a model of an ortholattice (OL). An OL can thus be thought of as a kind of "quantum logic" (QL). C(H) is also a model of an orthomodular lattice, which is an OL conjoined with the orthomodularity axiom (OMA). The rationalization of the OMA as a claim proper to physics has proven problematic, motivating the question of whether the OMA and its equivalents are required in an adequate characterization of QL. The OMA, it turns out, has strong connections to implication in QL. Here I provide an automated deduction of three implication-based equivalents of the OMA from orthomodular lattice theory. The proofs may be novel.

Keywords: automated deduction, quantum computing, orthomodular lattice, Hilbert space

1.0 Introduction

The optimization of quantum computing circuitry and compilers at some level must be expressed in terms of the description of quantum-mechanical behaviors ([1], [17], [18], [20]). In much the same way that conventional propositional (Boolean) logic (BL,[12]) is the logical structure of description of the behavior of classical physical systems (e.g. "the measurements of the position and momentum of particle P are commutative"), i.e., can be measured in either order, yielding the same results) and is isomorphic to a Boolean lattice ([10], [11], [19]), so also the algebra, C(H), of the closed linear subspaces of (equivalently, the system of linear operators on (observables in)) a Hilbert space H ([1], [4], [6], [9], [13]) is a logic of the descriptions of the behavior of quantum mechanical systems (e.g., "the measurements of the position and momentum of particle P are not commutative") and is a model ([10]) of an ortholattice (OL; [4]). An OL can thus be thought of as a kind of "quantum logic" (QL; [19]). C(H) is also a model of (i.e., isomorphic to a set of sentences which hold in) an orthomodular lattice (OML; [4], [7]), which is an OL conjoined with the orthomodularity axiom (OMA; see Figure 1). The rationalization of the OMA as a claim proper to physics has proven problematic ([13], Section 5-6), motivating the question of whether the OMA is required in an adequate characterization of QL. Thus formulated, the question suggests that the OMA and its equivalents are specific to an OML, and that as a consequence, banning the OMA from QL yields a "truer" quantum logic. The OMA, it turns out, has strong
connections to implication in QL, as demonstrated in the following.

**Lattice axioms**

\[
\begin{align*}
x &= c(c(x)) & \text{(AxLat1)} \\
x \lor y &= y \lor x & \text{(AxLat2)} \\
(x \lor y) \lor z &= x \lor (y \lor z) & \text{(AxLat3)} \\
(x \land y) \land z &= x \land (y \land z) & \text{(AxLat4)} \\
x \lor (x \land y) &= x & \text{(AxLat5)} \\
x \land (x \lor y) &= x & \text{(AxLat6)}
\end{align*}
\]

**Ortholattice axioms**

\[
\begin{align*}
c(x) \land x &= 0 & \text{(AxOL1)} \\
c(x) \lor x &= 1 & \text{(AxOL2)} \\
(x \land y) &= c(c(x) \lor c(y)) & \text{(AxOL3)}
\end{align*}
\]

**Orthomodularity axiom**

\[
y \lor c(y) \land (x \lor y) &= x \lor y & \text{(OMA)}
\]

**Definitions of implications and partial order**

\[
\begin{align*}
&i_1(x, y) = c(x) \lor (x \land y). \\
i_2(x, y) = i_1(c(y), c(x)). \\
i_3(x, y) = (c(x) \land y) \lor c(x) \lor c(y) \lor i_1(x, y). \\
i_4(x, y) = i_3(c(y), c(x)). \\
i_5(x, y) = (x \land y) \lor c(x) \land y \lor c(x) \land c(y). \\
&le(x, y) = (x \land (x \land y)).
\end{align*}
\]

where
- \( x, y \) are variables ranging over lattice nodes
- ^ is lattice meet
- \( \lor \) is lattice join
- \( c(x) \) is the orthocomplement of \( x \)
- \( i_1(x, y) \) means \( x \rightarrow y \) (Sasaki implication)
- \( i_2(x, y) \) means \( x \rightarrow y \) (Dishkant implication)
- \( i_3(x, y) \) means \( x \rightarrow y \) (Kalmbach implication)
- \( i_4(x, y) \) means \( x \rightarrow y \) (non-tollens implication)
- \( i_5(x, y) \) means \( x \rightarrow y \) (relevance implication)
- \( le(x, y) \) means \( x \leq y \)
- \( <-> \) means if and only if
- \( = \) is equivalence (\{12\})
- 1 is the maximum lattice element (= \( x \lor c(x) \))
- 0 is the minimum lattice element (= \( c(1) \))

**Figure 1.** Lattice, ortholattice, orthomodularity axioms, and some definitions.

Consider the following proposition.

\[
\begin{align*}
&((x \rightarrow_i y) = 1) \leftrightarrow le(x, y)
\end{align*}
\]

where \( i = 1, 2, 3, 4, 5 \).

**Figure 2.** Proposition 2.10
Note that there are five implications in QL (BL has only one). Proposition 2.10 is a generalization of the BL definition of implication, sometimes denoted $\rightarrow_0$.

**2.0 Method**

The OML axiomatizations of Megill, Pavičić, and Horner ([5], [14], [15], [16], [21], [22]) were implemented in a prover9 ([2]) script ([3]) configured to derive Proposition 2.10i, for each of $i = 1, 2, 3$ from orthomodular lattice theory, then executed in that framework on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 (clocked @ 2.33 GHz) and 8.00 GB RAM, running under the Windows Vista Home Premium /Cygwin operating environment.

**3.0 Results**

Figure 3 shows the proofs, generated by [3] on the platform described in Section 2.0, that orthomodular lattice theory implies Proposition 2.10i, for each of $i=1,2,3$.
73 \ x \ v \ (c(x) \ v \ y) = 1. \ [\text{back_rewrite(44), rewrite(\{69(5)\})}].
78 \ x \ v \ 0 = x. \ [\text{para(15(a,1), 40(a,1,2,1)), rewrite(\{38(2)\})}].
80 \ x \ v \ x = x. \ [\text{para(38(a,1), 40(a,1,2,1)), rewrite(\{8(5), 67(4)\})}].
91 \ 0 \ v \ x = x. \ [\text{para(78(a,1), 8(a,1)), flip(a)}].
93 \ x \ v \ (x \ v \ y) = x \ v \ y. \ [\text{para(80(a,1), 9(a,1,1)), flip(a)}].
338 \ c(c(c1) \ v \ c(c2)) = c1. \ [\text{para(57(a,1), 29(a,2,1,1)), rewrite(\{29(17), 7(27), 93(26), 7(27), 7(32), 73(31), 38(26), 8(26), 9(126), 8(25), 68(25), 7(11), 38(11), 7(19), 93(18), 7(19), 7(24), 73(23), 38(18), 8(16), 91(18), 91(17) \)), flip(b), merge(b)}].
339 -le(c1, c2). \ [\text{back_rewrite(37), rewrite(\{338(8), 8(4), 15(4)\}), xx(a)}].
340 \ F. \ [ur(54,a,339,a), rewrite(\{8(5), 338(6)\}), xx(a)}].

% Proof 1 at 2.32 (+ 0.08) seconds.
% Length of proof is 35.
% Level of proof is 8.

1 le(x, y) \rightarrow x = x ^ y \ # label("Df: less than") \ # label(non_clause). \ [assumption].
3 i2(x, y) = 1 \rightarrow le(x, y) \ # label("Proposition 2.10i2") \ # label(non_clause) \ # label(goal). \ [goal].
6 x = c(c(x)) \ # label("AxL1"). \ [assumption].
7 c(c(x)) = x. \ [copy(6), flip(a)].
8 x \ v \ y = y \ v \ x \ # label("AxL2"). \ [assumption].
11 x \ v \ (x ^ y) = x \ # label("AxL5"). \ [assumption].
12 x \ v \ (x ^ y) = x \ # label("AxL6"). \ [assumption].
13 x \ v \ x = 0 \ # label("AxOL1"). \ [assumption].
14 c(x) \ v \ x = 1 \ # label("AxOL2"). \ [assumption].
15 x \ v \ c(x) = 1. \ [copy(14), rewrite(\{8(2)\})].
16 x \ v \ c(x) = c(c(x) \ v \ c(y)) \ # label("AxOL3"). \ [assumption].
17 x \ v \ c(x) \ v \ c(y) = x ^ y \ # label("OMA"). \ [assumption].
22 i2(x, y) = c(c(y)) \ v \ (c(y) ^ c(x)) \ # label("Df: i2"). \ [assumption].
23 i2(x, y) = y \ v \ c(y \ v \ x). \ [copy(22), rewrite(\{7(3), 16(4), 7(3), 7(3)\})].
30 \ -le(x, y) | x ^ y = x \ # label("Df: less than"). \ [clausify(1)].
31 \ -le(x, y) | x \ v \ c(x) \ v \ c(y)) = x. \ [copy(30), rewrite(\{16(2)\})].
32 \ le(x, y) | x \ v \ y = x \ # label("Df: less than"). \ [clausify(1)].
33 \ le(x, y) | c(c(x) \ v \ c(y)) = x. \ [copy(32), rewrite(\{16(2)\})].
34 i2(c1, c2) = 1 \rightarrow le(c1, c2) \ # label("Proposition 2.10i2"). \ [deny(3)].
35 c2 \ v \ c(c1 \ v \ c2) = 1 \ # label("Proposition 2.10i2"). \ [deny(3)].
36 i2(c1, c2) = 1 \rightarrow le(c1, c2) \ # label("Proposition 2.10i2"). \ [deny(3)].
37 c2 \ v \ c(c1 \ v \ c2) = 1 \ # label("Proposition 2.10i2"). \ [deny(3)].
38 c1 = 0. \ [\text{back_rewrite(13), rewrite(\{16(2), 7(2), 15(2)\})}].
39 c(c(x) \ v \ c(x \ v \ y)) = x. \ [\text{back_rewrite(12), rewrite(\{16(2)\})}].
40 x \ v \ c(c(x) \ v \ c(y)) = x. \ [\text{back_rewrite(11), rewrite(\{16(1)\})}].
57 c2 \ v \ c(c1 \ v \ c2) = 1 \ # label("Proposition 2.10i2"). \ [resolve(35, b, 31, a)].
62 le(x, y) \ v \ y). \ [resolve(39, a, 33, b)].
78 \ x \ v \ 0 = x. \ [\text{para(15(a,1), 40(a,1,2,1)), rewrite(\{38(2)\})}].
79 \ x \ v \ c(y \ v \ c(x)) = x. \ [\text{para(18(a,1), 40(a,1,2,1))}].
91 \ 0 \ v \ x = x. \ [\text{para(78(a,1), 8(a,1)), flip(a)}].
362 c2 \ v \ c(c1 \ v \ c2) = 1 \ # label("Proposition 2.10i2"). \ [resolve(35, b, 31, a)].
6823 c1 \ v \ c2 = c2. \ [\text{para(362(a,1), 18(a,1,2,1)), rewrite(\{38(8), 8(8), 91(8)\}), flip(b), merge(b)}].
6824 \ -le(c1, c2). \ [\text{back_rewrite(37), rewrite(\{6823(4), 15(4)\}), xx(a)}].
6828 \ F. \ [\text{para(6823(a,1), 62(a,2)), unit_del(\{a, 6824\})}].

% Proof 1 at 3.14 (+ 0.11) seconds.
% Length of proof is 50.
% Level of proof is 9.
6 x = c(c(x)) # label("AxL1"). [assumption].
7 c(c(x)) = x. [copy(6), flip(a)].
8 x v y = y v x # label("AxL2"). [assumption].
9 (x v y) v z = x v (y v z) # label("AxL3"). [assumption].
11 x v (x ^ y) = x # label("AxL5"). [assumption].
12 x ^ (x v y) = x # label("AxL6"). [assumption].
13 x v c(x) = 1. [copy(14), rewrite([8(2)])].
14 x v c(x) = 1 # label("AxOL1"). [assumption].
15 c(x) ^ x = 0 # label("AxOL1"). [assumption].
16 x ^ y = c(c(x) v c(y)) # label("AxOL3"). [assumption].
17 x v (c(x) ^ (y v x)) = y v x # label("OMA"). [assumption].
18 x v c(x v c(y v x)) = y v x. [copy(17), rewrite([16(3), 7(2)])].
24 i3(x, y) = ((c(x) ^ y) v (c(x) ^ c(y))) v (c(x) v (x ^ y)) # label("Df: i3"). [assumption].
25 i3(x, y) = c(x v y) v (c(x v c(y)) v (c(x) v (x ^ y))). [copy(24), rewrite([16(3), 7(3), 16(7), 8(7), 16(9), 14(14)])].
28 i5(x, y) = ((x ^ y) v (c(x) ^ y)) v (c(x) ^ c(y)) # label("Df: i5"). [assumption].
29 i5(x, y) = c(x v y) v (c(x v c(y)) v c(c(x) v c(y))). [copy(28), rewrite([16(2), 16(7), 7(7), 8(9), 16(12), 7(11), 7(11), 8(12)])].
30 le(x, y) | x ^ y = x # label("Df: less than"). [clausify(1)].
31 le(x, y) | c(x v c(y)) = x. [copy(30), rewrite([16(2)])].
32 le(x, y) | x ^ y = x # label("Df: less than"). [clausify(1)].
33 le(x, y) | c(x v c(y)) = x. [copy(32), rewrite([16(2)])].
34 i3(c1, c2) = 1 | le(c1, c2) # label("Proposition 2.10i3"). [deny(3)].
35 c(c1 v c2) v (c(c1 v c(c2)) v (c(c1) v c(c(c1) v c(c2)))) = 1 | le(c1, c2).
36 i3(c1, c2) != 1 | -le(c1, c2) # label("Proposition 2.10i3"). [deny(3)].
37 c(c1 v c2) v (c(c1 v c(c2)) v (c(c1) v c(c(c1) v c(c2)))) != 1 | -le(c1, c2).
38 c(1) = 0. [back rewrite(13), rewrite([16(2), 7(2), 15(2)])].
39 c(c(x) v (x v y)) = x. [back rewrite(12), rewrite([16(2)])].
40 x v c(x v c(y)) = x. [back rewrite(11), rewrite([16(1)])].
43 c(c1) v (c(c1 v c2) v (c(c1 v c(c2)) v c(c(c1) v c(c2)))) = 1 | -le(c1, c2).
44 c(c1) v (c(c1 v c2) v (c(c1 v c(c2)) v (c(c1) v c(c(c1) v c(c2)))) = 1 | -le(c1, c2).
46 x v (c(x) v y) = 1 v y. [para(15(a, 1), 9(a, 1, 1)), flip(a)].
56 le(x, y) | c(c(x) v c(x)) = x. [para(8(a, 1), 33(b, 1, 1)]).
64 c(x) v (c(x v y) = c(x). [para(39(a, 1), 7(a, 1, 1)), flip(a)].
68 c(x v y) = c(x v (c(x v y) = c(x)). [para(39(a, 1), 7(a, 1, 1)), flip(a)].
70 1 v x = 1. [para(38(a, 1), 39(a, 1, 1)), rewrite([68(6)])].
72 x v (c(x) v y) = 1. [back rewrite(46), rewrite([70(5)])].
79 x v 0 = x. [para(15(a, 1), 40(a, 1, 2, 1)), rewrite([38(2)])].
81 x v x = x. [para(38(a, 1), 40(a, 1, 2, 1)), rewrite([8(3), 68(4)])].
90 c(c1) v (c(c1 v c2) v (c(c1 v c(c2)) v (c(c1) v c(c2)))) = 1 | c(c1) v c(c2) = c1.
93 0 v x = x. [para(79(a, 1), 8(a, 1)), flip(a)].
95 x v (x v y) = y v y. [para(81(a, 1), 9(a, 1, 1)), flip(a)].
36 c(x) v (c(x v y) = c(x) v z. [para(64(a, 1), 9(a, 1, 1)), flip(a)].
347 c(c1) v (c(c1) v c(c2)) = 1 | c(c1) v c(c2) = c1.
[back rewrite(90), rewrite([336(20), 336(15)])].
350 c(c1) v (c(c1) v c(c2)) = c1.
[back rewrite(43), rewrite([336(20), 336(15)])].
20075 c(c1) v c(c2) = c1.
[para(347(a, 1), 29(a, 2, 1, 1), rewrite([29(17), 7(27), 95(26), 7(27), 7(32), 74(31), 38(26), 8(26), 93(26), 8(25), 69(25), 7(11), 38(11), 7(19), 95(18), 7(19), 7(24), 74(23), 38(18), 8(18), 93(17)], flip(b), merge(b)].
20076 -le(c1, c2). [back rewrite(350), rewrite([20075(8), 8(4), 15(4)], xx(a)].
20077 SF. [ur(56, a, 20076, a), rewrite([8(5), 20075(6), xx(a)].

Figure 3. Summary of a prover9 (2) proof of Proposition 2.10i, for each of i = 1, 2, 3 from orthomodular lattice theory. The proofs assume the default inference rules of prover9. The general
form of a line in this proof is “line_number conclusion [derivation]”, where line_number is a unique identifier of a line in the proof, and conclusion is the result of applying the prover9 inference rules (such as paramodulation, copying, and rewriting), noted in square brackets (denoting the derivation), to the lines cited in those brackets. Note that some of “logical” proof lines in the above have been transformed to two text lines, with the derivation appearing on a text line following a text line containing the first part of that logical line. The detailed syntax and semantics of these notations can be found in [2]. All prover9 proofs are by default proofs by contradiction.

The total time to produce the proofs in Figure 3 on the platform described in Section 2.0 was approximately 11 seconds.

4.0 Discussion

The results of Section 3.0 motivate several observations:

1. The proofs in Figure 3 for i=1 and i=3 use L1, L2, L3, L5, L6, OL1, OL2, and OL3. In contrast, the proof for i=2 uses L1, L2, L5, L6, OL1, OL2, and OL3. This suggests (but does not prove) that Proposition 2.10, for i = 1,2, may have axiomatic bases that are more closely related to each other than they are to the axiomatic basis for i = 3. Future work will explore this hypothesis.

2. The proofs in Section 3.0 may be novel.

3. Companion papers provide proofs for i = 4, 5, and for Propositions 2.10i, i = 1,2,3,4,5, imply the OMA ([23]).

5.0 Acknowledgements

This work benefited from discussions with Tom Oberdan, Frank Pecchioni, Tony Pawlicki, and the late John K. Prentice, whose passion for foundations of physics inspired those of us privileged to have known him. For any infelicities that remain, I am solely responsible.

6.0 References