Algebraic modelling of quantum mechanical equations in the finite- and infinite-dimensional Hilbert spaces

Megill, Norman Dwight

University of Zagreb

PHYSICS DEPARTMENT OF THE
FACULTY OF SCIENCE

Norman Dwight Megill

ALGEBRAIC MODELLING OF QUANTUM MECHANICAL EQUATIONS IN THE FINITE- AND INFINITE-DIMENSIONAL HILBERT SPACES

DOCTORAL THESIS

Zagreb, 2011

Sveučilište u Zagrebu

FIZIČKI ODSJEK
PRIRODOSLOVNO-MATEMATIČKOG FAKULTETA

Norman Dwight Megill

ALGEBARSKO MODELIRANJE KVANTNO-MEHANIČKIH JEDNADŽBI U KONAČNO I BESKONAČNO DIMENZIONALNIM HILBERTOVIM PROSTORIMA

DOKTORSKI RAD

Zagreb, 2011

University of Zagreb

PHYSICS DEPARTMENT OF THE
FACULTY OF SCIENCE

Norman Dwight Megill

ALGEBRAIC MODELLING OF QUANTUM MECHANICAL EQUATIONS IN THE FINITE- AND INFINITE-DIMENSIONAL HILBERT SPACES

DOCTORAL THESIS

Supervisor:
Prof. Mladen Pavičić

Zagreb, 2011

Sveučilište u Zagrebu

FIZIČKI ODSJEK
PRIRODOSLOVNO-MATEMATIČKOG FAKULTETA

Norman Dwight Megill

ALGEBARSKO MODELIRANJE KVANTNO-MEHANIČKIH JEDNADŽBI U KONAČNO I BESKONAČNO DIMENZIONALNIM HILBERTOVIM PROSTORIMA

DOKTORSKI RAD

Mentor:
Prof. Mladen Pavičić

Zagreb, 2011

Committees

This thesis, entitled *Algebraic Modelling of Quantum Mechanical Equations in the Finite- and Infinite-Dimensional Hilbert Spaces* and written by Norman Dwight Megill, has been approved for the Department of Physics of the Faculty of Science of the University of Zagreb by the following committee: Prof. Hrvoje Šikić (Department of Mathematics, Faculty of Science, University of Zagreb), Prof. Mladen Pavičić (Chair of Physics, Faculty of Civil Engineering, University of Zagreb), Prof. Barry C. Sanders (Institute for Quantum Information Science, University of Calgary, Canada), Prof. Aleksa Bjeliš (Department of Physics, Faculty of Science, University of Zagreb), and Prof. Mirko Primc (Department of Mathematics, Faculty of Science, University of Zagreb).

The final copy of this thesis has been examined by the signatories, and they found that both the content and the form meet acceptable presentation standards for a scholarly work in the discipline of mathematical quantum physics. The Council of the Department of Physics accepted their opinion on November 8, 2011.

The committee for the defense, approved by the Council of the Department of Physics, consisted of Prof. Hrvoje Šikić (Department of Mathematics, Faculty of Science, University of Zagreb), Prof. Mladen Pavičić (Chair of Physics, Faculty of Civil Engineering, University of Zagreb), Prof. Slobodan Bosanac (Division of Physical Chemistry, Institute Ruđer Bošković, Zagreb), Prof. Denis Sunko (Department of Physics, Faculty of Science, University of Zagreb), and Prof. Mirko Primc (Department of Mathematics, Faculty of Science, University of Zagreb). The defense took place on December 14, 2011.

Acknowledgments

I am greatly indebted to Prof. Pavičić, without whose inspiration and guidance this thesis would not be possible. I also wish to thank Susan Cass for her encouragement and patience.

Naći iglu u plastu sijena.

Abstract

The Hilbert space of quantum mechanics has a dual representation in lattice theory, called the Hilbert lattice. In addition to offering the potential for new insights, the lattice-theoretical approach may be computationally efficient for certain kinds of quantum mechanics problems, particularly if, in the future, we are able to exploit what may be a "natural" fit with quantum computation. The equations that hold in the Hilbert space lattice representation are not completely known and are poorly understood, although much progress has been made in the last several years. This work contributes to the development of these equations, with special attention to the so-called generalized orthoarguesian equations. Many new results that do not appear in the literature are given, along with their detailed proofs. In addition, possible approaches for work towards answering some remaining open questions are discussed.

Keywords

Hilbert space, Hilbert lattice, orthoarguesian property, strong state, quantum logic, quantum computation, Godowski equations, orthomodular lattice

Prošireni sažetak

Pozadina. Stanja u kvantnoj mehanici mogu se modelirati kao vektori u Hilbertovom prostoru. Skup zatvorenih podprostora konačno ili beskonačno dimenzionalnog Hilbertova prostora član je klase čestica koje se zovu Hilbertove rešetke (Hilbert lattice, HL). (Rešetka je djelomično uređen skup u kojemu svaka dva člana imaju najmanju gornju i najveću donju granicu. Ovaj i svi drugi ovdje korišteni termini formalno su definirani u disertaciji). Obratno, moguće je izvesti Hilbertov prostor polazeći od HL. Zbog ovog dvostrukog odnosa razumijevanje svojstava HL-a može dovesti do boljeg razumijevanja svojstava Hilbertova prostora. Osim što nudi mogućnost novih uvida, teorijski pristup rešetki može biti računski efikasan za neke vrste kvantno mehaničkih problema, naročito ako, u budućnosti, budemo mogli koristiti ono što bi mogao biti "prirodno" odgovarajući dio za kvantno računanje. Jednadžbe koje u Hilbertovu prostoru podržavaju prikaz rešetke nisu u potpunosti poznate i nedovoljno ih se razumije, premda je tijekom nekoliko posljednjih godina učinjen veliki napredak.

Familija svih HL-ova definirna je (aksiomatizirana) skupom uvjeta prvoga reda koji uključuju (egzistencijalne) kvantifikatore. Za određeni broj uvjeta nultog reda odnosno jednadžbi bez kvantifikatora, može se pokazati da vrijede u svakom HL-u. Najočigledniji od njih su jednadžbe koje definiraju bilo koju rešetku (te posebno svaku orto-rešetku), koje su dio skupa aksioma. Godine 1937. Husimi je otkrio ortomodularni zakon (koji je sada također dio HL definicije), koji je bio intenzivno obrađen u literaturi o klasi ortomodularnih rešetki (OML), kojih je HL podklasa.

Za razliku od uvjeta prvoga reda, jednadžbe nam omogućavaju da direktno baratamo objektima u podprostoru Hilbertova prostora i dobijemo vrstu računske "algebre" za rad s tim objektima. Jednadžbe su posebno prikladne za efikasne računske tehnike. Klasa rešetki definirana samo jednadžbama, kao što je OML, naziva se jednadžbenim varijetetom. Klasa HL-a sama po sebi nije jednadžbeni varijetet (za što je dokaz naveden u disertaciji). Usprkos tome, klasa rešetki koju je generirao (tj. koja zadovoljava) skup jednadžbi koje vrijede u HL-u može se proučavati odvojeno kao superklasa od HL-a i svi rezultati su automatski primjenjivi na HL kao poseban slučaj.

Jedan važan neriješen problem je pronaći sve moguće jednadžbene zakone koji vrijede u HL-u. S jačim jednadžbama moguće je proučiti više karakteristika HL-a korištenjem samo jednadžbenih varijeteta.

Ovdje je kratki pregled napretka postignutog do sada. Trebalo je nekoliko desetljeća nakon Husimieva OML zakona da se pronađe drugi, a to je bio ortoarguesiev zakon kojega je otkrio Alan Day 1975.

1981. g. Godowski je otkrio nezavisnu beskonačnu familiju HL jednadžbi, baziranu na kvantnim probabilističkim stanjima. Te jednadžbe nazivano “Godowski-eve jednadžbe” ili *n*-Gos. Godine 1986. Mayet je našao algoritam za generiranje većeg skupa jednadžbi (nazvan MGEs), koji je također utemeljen na stanjima, čiji su podskup bile Godowski-eve jednadžbe), premda se na početku nije znalo da li je ikoja od njih nezavisna od *n*-Go jednadžbi. Od 2006. do 2009., Megill i Pavičić pronašli su nove jednadžbe utemeljene na Mayet-ovu algoritmu za koje se pokazalo da se ne daju izvesti iz Godowski-evih.

U 2000. g. Megill i Pavičić otkrili su novu familiju jednadžbi koje vrijede u HL-u—generalizirane ortoarguesieve jednadžbe, nazvane *n*OA zakoni ($n \geq 3$). OA zakon Alan-a Day-a je drugi član ove serije, zakon 4OA, a Greechie/Godowski-eve jednadžbe izvedene iz OA su ekvivalentne prvome članu, zakonu 3OA. Dok je otvoren problem da li se obitelj *n*OA sastoji od uzastopno jačih jednadžbi, mi smo dokazali (obimnim kompjuterskim traženjem protuprimjera) da su zakoni 3OA, 4OA, 5OA i 6OA uzastopno jači. 2011. g. uspjeli smo dokazati da je zakon 7OA jači od zakona 6OA.

Godine 1995. Maria Solèr je dokazala da je dodavanjem dva dodatna HL aksioma, moguće iz HL-a izvesti Hilbertov prostor čije je polje jedno od “klasičnih” polja kvantne mehanike (realno, kompleksno ili kvaternionsko). Solèr-in teorem upotpunio je dugo neostvareni cilj da se Hilbertov prostor kvantne mehanike izvede iz nekog HL-a pokazujući da su oni dualni. Godine 2006. Mayet je opisao novu obitelj jednadžbi, nazvanu *E* jednadžbe, utemeljenu na jednom svojstvu Hilbertova prostora koje se naziva vektorski-valuiranim stanjem. Važno je reći da te jednadžbe ne vrijede za svako moguće polje koje se može dovesti u vezu s Hilbertovim prostorom već samo za ona polja s karakteristikom 0, koja uključuju klasična polja kvantne mehanike. To nam daje jednadžbeni uvjet koji je u stvari ovisan o (te ih tako djelomično i opisuje) Solèr-inim dodatnim uvjetima (prvoga reda) dodanim HL-u.

Ovdje ćemo ukratko sumirati ključne teme pokrivene u disertaciji koje se odnose na traženje novih HL jednadžbi.

Ortomodularne rešetke. Veliki broj uvjeta koji vrijede u OML-u prikupljen je u poglavlju 3, za kasniju uporabu. Oni koji se nisu ranije pojavili u literaturi popraćeni su detaljnim dokazima. Određeni broj novih rezultata naveden je za takozvanu Sasaki hook operaciju, koja postaje koristan alat u kasnijim poglavljima.

Ortoarguesieve jednadžbe. Poglavlje 4 predstavlja ekstenzivnu studiju generaliziranih ortoarguesievih rešetki (jednadžbeni varijeteti *n*OA). Prezentiran je revidirani dokaz tih zakona i razmotreni su poznati rezultati neovisnosti (sve do 7OA). Nekoliko sustava označavanja, korisnih u različitim situacijama, uvedeno je kako bi se kompaktno reprezentirale te jednadžbe. Mnoge jednadžbe koje su ekvivalentne i koje su posljedice zakona *n*OA, koje su gotovo sve nove, izvedene su uz detaljne dokaze.

Važan neriješen, otvoren problem je "pretpostavka ortoarguesievog identiteta," koja propituje da li je uvjet poznat kao zakon ortoarguesievog identiteta ekvivalentan ortoarguesianskom zakonu. Ako ova pretpostavka vrijedi, bila bi moćan alat za dokazivanje teorema. Jedna ekstenzivna studija koja je posljedica ove pretpostavke, jednako kao i drugih pretpostavki koje je impliciraju, predstavlja središnji dio posljednjeg odjeljka poglavlja 3.

Ostale jednadžbe Hilbertove rešetke. Poglavlje 5 razmatra druge gore spomenute jednadžbene varijetete. Posebno je predstavljeno 16 novih Mayet-Godowski-evih jednadžbi (MGEs), otkrivenih kao dio ove disertacije.

Poglavlje 6 istražuje svojstva superpozicije prvoga reda i modularnu simetriju, od čega niti jedno do sada nije dovelo do nove jednadžbe. Prezentirana je pretpostavljena jednadžba izvedena iz modularne simetrije, ali je otvoreni problem da li njen izvod (počevši od modularne simetrije) vrijedi u svim OML-ovima.

Jednadžbe rešetke za konačno dimenzionalne Hilbertove prostore. Konačno dimenzionalni Hilbertovi prostori važni su za mnoge probleme u kvantnoj mehanici, uključujući većinu eksperimenata koji uključuju čestična stanja i većinu pristupa kvantnom računanju. Poglavlje 7 razmatra modularni zakon i Arguesiev zakon koji vrijedi u zatvorenim podprostorima konačno dimenzionalnih Hilbertovih prostora. Izvedena je nova serija Arguesievih zakona višeg reda. Prodiskutirane su moguće primjene Pappusova zakona projektivne geometrije.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION AND OVERVIEW

At the very end of his book Quantum Computation and Quantum Communication, Mladen Pavičić [99] lays forth a bold vision for a possible future of quantum computing, one in which a universal quantum "algebra" is discovered that will finally turn the search for quantum algorithms—of which less than half a dozen exist today, in spite of intense work by hundreds of researchers—from a mysterious black art into a science. The goal of such an algebra would be to provide a quantum analog, in some sense that is still unknown, to the Boolean algebra used by classical computation. A possible clue, he believes, may be provided by uncovering and understanding the link between the lattice equations of Hilbert space and quantum computation. Today such a link is almost completely unknown, other than the fact that both are independently derived starting from Hilbert space. This thesis will investigate and continue with the ideas envisioned by Pavičić, building on the foundation that he and this author have developed over the last several years in the various papers that we have co-authored [102] [103] [104] [76] [73] [77] [78] [80] [79] [109].

> The main idea behind representing Hilbert space by an orthomodular lattice is to add additional strengthening axioms which are still weak enough so as not to make it modular. These axioms will give us the so-called Hilbert lattices
>
> . . .
>
> Thus we do arrive at a full Hilbert space, but the axioms for the Hilbert lattices that we used for this purpose are too involved to reveal a possible transition to its finite-dimensional representation. This is because in the past, the axioms were simply read off from the Hilbert space structure and were formulated as predicative statements of the first and second order that cannot be implemented by a quantum Turing machine

—M. Pavičić [99, pp. 195–196]

Towards this goal, this work involves, in particular, an extensive study of the equations that hold in the Hilbert lattice, that is, the lattice of closed subspaces of infinite- or finite-dimensional Hilbert space. Primary emphasis has been placed on making progress towards the following subgoals, all of which are currently open problems about which very little is known: (1) extending the known set of equations with new discoveries, (2) determining the minimum set of additional first-order logical properties (i.e. those involving quantifiers) that are needed to re-derive Hilbert space from existing and new equations, (3) determining what fragment of Hilbert space it is possible to describe by the equations alone (without the additional first-order properties), and (4) connecting these results, starting from this Hilbert space fragment, to equations or conditions (such as such as Schrödinger's equation) that hold in ordinary quantum mechanics, especially those related to quantum computation and qubits. Our ultimate goal is to achieve, to whatever extent possible, a way of "talking about" Hilbert space, or at least some fragment of it, using only (zeroth-order) equations, in the hope of eventually arriving at a practical computational "algebra" for quantum algorithms.

1.1 Background and history

In this section, we will review some terminology, then we will summarize the history and the present state of knowledge concerning Hilbert lattices. The following brief definitions are meant to assist this informal discussion and will be developed in more detail in Ch. 2.

A **lattice** is an algebra $\langle L, \vee, \wedge \rangle$ in which the following equations hold: $a \vee b = b \vee a, a \wedge b = b \wedge a$, $(a \vee b) \vee c = a \vee (b \vee c)$, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$, $a \wedge (a \vee b) = a$, and $a \vee (a \wedge b) = a$. Partial ordering $a \leq b$ is defined by $a \vee b = b$. An **ortholattice** (OL) is a lattice with an **orthocomplement** operation $'$ such that $a \leq b \Rightarrow b' \leq a'$ and $a'' = a$. An **orthomodular lattice** (OML) is an ortholattice in which the **orthomodular law** $b \leq a$ & $c \leq a' \Rightarrow a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ holds. (The terms OL, OML, etc. refer to the proper classes of all lattices obeying the respective equations.) An **orthocomplemented modular lattice** (MOL) is an ortholattice in which the **modular law** $b \leq a \Rightarrow a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ holds [99, p. 192]. These two laws are weakened, but successively stronger, forms of the **distributive law** $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ that holds in a **Boolean (classical) lattice** (BA). Indeed, the failure of the distributive law is the key feature that distinguishes these lattices from Boolean lattices.

The set of closed subspaces of a finite- or infinite-dimensional Hilbert space is called $\mathscr{C}(H)$ and is a member of a class of lattices called **Hilbert lattices** (HL). The family of all HLs is defined by first-order conditions involving quantifiers [Def. 2.3.1 below (p. 19)]. The importance of a Hilbert lattice is that a Hilbert space can be derived from it, meaning that it serves as a dual representation for Hilbert space (and thus quantum mechanics). A loose analogy is the way that the frequency domain serves as a dual representation for the time domain via the Fourier transform, although the reconstruction of a Hilbert space from a Hilbert lattice is far more complicated.

Certain zeroth-order conditions, i.e. equational laws not involving first-order quantifiers, hold in a Hilbert lattice in addition to the basic equations holding in any ortholattice. The earliest known equational condition, the OML law discovered by Husimi in 1937 [49, p. 7], is normally part of the Hilbert lattice definition, and other, stronger equations can be derived from the definition. Unlike first-order conditions, equational laws allow us to manipulate the subspace objects in Hilbert space directly and provide a kind of computational "algebra" for working with those objects. An important unsolved problem is to find the strongest possible equational laws for Hilbert lattices [64], shrinking the size of the OML class towards the class of all BAs and as a consequence allowing more and more classical techniques to become useable.

In finite-dimensional Hilbert spaces, a condition stronger than the OML law, called the modular law, also holds. Ordinarily, the modular law is not considered part of the Hilbert lattice definition, since HL is meant to encompass both finite- and infinite-dimensional Hilbert spaces. We will study conditions that hold in finite-dimensional Hilbert spaces separately, in Ch. 7 (p. 111).

Before 1975, it was known only that orthomodular lattice equations hold in infinite-dimensional Hilbert space and that the modular law holds in finite-dimensional Hilbert space. This fact alone led to a vast body of research, papers, and books on the subject of orthomodular lattices (as well as modular lattices, but to a lesser extent) [96] [49] [6] [86].

In 1975, Alan Day discovered that a stronger equation, the 6-variable **orthoarguesian law** (OA), holds in infinite-dimensional Hilbert space (cf. [27] [31]) Perhaps because the equation was complicated and there were no tools available to work with it conveniently, it remained more or less a quiet curiosity for many years. However, it provided the first clue that the Hilbert lattice embodied a much richer equational structure than was previously thought. The first study of Day's equation was done in 1984, when Godowski and Greechie derived 3- and 4-variable consequences of OA, although their relationship to the original OA remained unclear [27]. (Later, Megill and Pavičić showed that these were strictly weaker than the original OA, although stronger than the orthomodular law [76].)

In 1981, Godowski discovered an unrelated infinite series of stronger equations, based on quantum probability states, that also hold in infinite-dimensional Hilbert space [26]. We call these "Godowski's equations" [*n*-Go, Th. 5.1.3 (p. 79)]. In 1986, Mayet gave an algorithm for generating a larger variety of equations, also based on states, of which Godowski's were a subset [65]. Although Mayet exhibited some sample equations he found with his algorithm, Megill and Pavičić showed that all of his examples were derivable from Godowski's [76], so it was unclear if Mayet had discovered anything new i.e. if any such equations stronger than Godowski's exist. However, in 2006–2009, Megill and Pavičić found new equations based on Mayet's algorithm that were shown not to be derivable from Godowski's [81] [105] [82]. We will show some additional equations in this family that have been discovered [Sec. 5.2 (p. 81)].

In 1995, a remarkable and very significant breakthrough was achieved by Maria Solèr [115] [40] [112]. She proved that with a small number of additional first-order conditions (atomicity, irreducibility, completeness, lattice height 4, and an infinite set of mutually orthogonal atoms satisfying a "harmonic conjugate" condition), an infinite-dimensional Hilbert space can be recovered from from an orthomodular lattice, with the only ambiguity being that its field[1] may be real, complex, or quaternionic. Mayet [66] extended this result with additional conditions that uniquely determine the complex field of the Hilbert space used by quantum mechanics, although an equivalent condition to add to a Hilbert lattice is still unknown. The importance of Solèr's work should not be underestimated, as it provides the key missing piece that, before 1995, would have made goal of this thesis impossible.

Although it is defined independently, HL is in effect the collection (up to isomorphism) of all $\mathscr{C}(H)$s of all (generalized) finite- and infinite-dimensional Hilbert spaces on any (skew) field. The phrases "in any HL" and "in any $\mathscr{C}(H)$" in effect say the same thing, although we typically use the latter to indicate a result derived from the properties of the $\mathscr{C}(H)$ of a Hilbert space H as opposed to properties derived directly from the axioms defining HL.

By adding the infinite orthogonal and harmonic conjugate sequences [Def. 2.3.4 (p. 20)] required by Solèr's theorem, we restrict HL to include only the collection (up to isomorphism) of $\mathscr{C}(H)$s of those Hilbert spaces where the field is real, complex, or quaternionic.

It should be noted that the Solèr/Mayet conditions do not make use of any of the newer Hilbert lattice equations described above, but instead add first-order (quantified) conditions on top of the standard orthomodular lattice equations to achieve their goal. An open problem is whether these first-order conditions can be replaced by weaker first-order ones together with zeroth-order equations to make up the difference.

In 2000, Megill and Pavičić discovered a new infinite series of equations that we called

[1]This usage of "field" conflicts with the standard mathematical definition in which multiplication is commutative (which is not the case for quaternions), and more properly we should use "division ring" or "skew field." However, we adopt the literature usage e.g. Ref. [40, p. 205] where "field" implicitly means "skew field."

nOA, with $n \geq 3$, that hold in the Hilbert lattice and are strictly stronger than the OML law [76]. Alan Day's OA law is the second member of this series, 4OA, and Greechie/Godowski's OA-derivative equations [27] are equivalent to the first member, 3OA. With a massive computer search involving a 192-CPU Linux cluster, we proved that third [76] and fourth [105, p. 766, Th. 11] equations, 5OA and 6OA, are strictly stronger than Day's 4OA, and also that 6OA is strictly stronger than 5OA. In 2011, we were able to prove that 7OA is strictly stronger than 6OA [84]. We will review these results in Ch. 4. We conjecture that the nOA series provides an infinite progression of successively stronger members.

The 2000 paper of Megill and Pavičić was the first comprehensive study of both OA-related and GO-related equations, uncovering many new results and interrelationships [76]. Previously, very little was known about these equations, in part because their size made them extremely difficult to work with. The development of new computer programs by Megill, along with powerful new notation introduced by Pavičić, enabled a practical study of these equations.

In 2006, Mayet [67, 68] described an algorithm for generating a series of equations, called E_A, that hold in HL and include the nOA family. While he provided an example of such an equation that was apparently different from the nOA series, in 2009, Megill and Pavičić showed that this example could in fact be derived from the 3OA law [82]. It remains an open problem whether any of the E_A is independent from the nOA series.

Also in 2006, Mayet [67, 68] described an algorithm for generating so-called "E" equations that are based on a property of Hilbert space called "vector-valued states." Importantly, these equations do not hold for every possible field (division ring) that can be associated with a (general) Hilbert space, but require that the field be what is called "characteristic 0," a property possessed by, among other fields, the real, complex, and quaternion "classical" fields of quantum mechanics. Thus these equations do not hold in every HL, but they do hold in every HL that is supplemented with the infinite orthogonal sequence and harmonic conjugate axioms [Def. 2.3.4 (p. 20)] that imply Solèr's theorem.

Beyond the above results, very little is known about Hilbert lattice equations. While the theory of ortholattices (OLs) is decidable (Brun's algorithm [13]), it is unknown even whether the theory of OMLs is decidable [49]. (Regarding the latter problem, Pavičić and Megill discovered an equational variety called called WOML or weakly orthomodular lattices, that is smaller than OL but larger than OML and that is isomorphic to all of OML [102]. Thus OML is decidable iff WOML is decidable.) Much less is known about the set of all equations that hold in Hilbert lattices (which include the OA and GO equations). It is not even known if these equations are recursively enumerable [64].

A major focus in this thesis is on orthoarguesian lattices nOA and their equations, to which we devote Ch. 4. Normally we do not give proofs for known theorems but simply make the appropriate literature references. Unless otherwise indicated, all theorems with explicit proofs have not been published to this author's knowledge.

1.2 Overview of chapters

In this section, we give an overview of the topics covered in subsequent chapters.

Ch. 2 (p. 12)—In this chapter, we review the prerequisites for later chapters. The review is brief, and it is best if the reader has some prior acquaintance with the material, but references are provided should that not be the case.

Ch. 3 (p. 27)—This chapter begins with a brief review of the orthomodular law as well the related operations and notation we will use later. A list of properties of OMLs, most of which will be used in later chapters, are presented. Whenever an equation or other condition is known to have appeared in the literature, a reference is given. Unless otherwise indicated, all theorems accompanied by proofs are believed to have not appeared previously in the literature. In Sec. 3.2 (p. 32), we focus on one type of conditional, called the Sasaki hook, which frequently occurs in the study of the orthoarguesian laws and other equations that hold in the lattice of closed subspaces $\mathscr{C}(H)$ for a Hilbert space H. To this author's knowledge, none of the theorems in Sec. 3.2 have previously been published, and all of them are accompanied by detailed proofs.

Ch. 4 (p. 37)—This chapter provides an extensive study of generalized orthoarguesian lattices (the equational varieties *n*OA). Sec. 4.1 (p. 37) repeats the proof (correcting some minor typos from an earlier published version) that the *n*OA laws hold in any $\mathscr{C}(H)$, which was discovered in 2000 by Megill and Pavičić [76]. Sec. 4.2 (p. 43) provides three different notations for compactly expressing *n*OA-related equations, all of which are useful in different situations. Sec. 4.3 (p. 48) reviews the known independence results for the *n*OA laws. Sec. 4.4 (p. 50) proves many equivalents for the 3OA law, almost all of which have not been published before. Finally, in Sec. 4.5 (p. 62), we define the "orthoarguesian identity laws" and present work towards the still unsolved conjecture [Conjecture 4.5.2 (p. 63)] that they are equivalent to the *n*OA laws.

Ch. 5 (p. 78)—This chapter reviews what is known about three classes of equations that hold in $\mathscr{C}(H)$: the Godowski equations, the Mayet-Godowski equations (MGEs), and Mayet's E-equations. The relationships among these and other known $\mathscr{C}(H)$ equations is summarized in Fig. 1.1 (p. 9). In Sec. 5.2.1 (p. 89), we present 18 examples of MGEs, 16 of which are new and haven't been published before. These are summarized in Tables 5.1 (p. 90) through 5.5 (p. 94).

Ch. 6 (p. 100)—In this chapter, we describe two properties that hold in $\mathscr{C}(H)$, M-symmetry (along with the related O-symmetry) and superposition. These are first-order properties described using quantifiers. An open problem is whether equations can be derived from these quantified conditions. In the case of M-symmetry, we show how equational candidates can in principle be derived from the M-symmetry law. In particular, the method produces an equation [Eq. (6.27), p. 107] which, if it could be proved to hold in all OMLs, would result in a (most likely) new equation holding in $\mathscr{C}(H)$. The problem thus reduces to the conjecture that Eq. (6.27) holds in all OMLs, which is currently unknown.

In Sec. 6.2 (p. 108), we describe how the superposition condition in a Hilbert lattice relates to the superposition of quantum states in a Hilbert space. We also show, in Fig. 6.4 (p. 110), the smallest 3-atom-per-block Greechie diagram in which the superposition principle holds.

Ch. 7 (p. 111)—In this chapter, we study four properties that hold (or in the last case may hold) in the lattice $\mathscr{C}(H)$ of (closed) subspaces of a finite-dimensional Hilbert space: the modular law, the Arguesian law, the higher-order Arguesian laws, and Pappus's postulate. A summary of this chapter is given in Sec. 1.4 (p. 10) below.

1.3 Summary of known Hilbert lattice equations

The families of lattices OL, OML, MOL, and BA are completely characterized by identities, i.e. equational conditions. Such families are called **equational varieties**. Equations, as opposed to quantified conditions, offer many advantages, such as being amenable to fast algorithms for testing finite lattice examples as well as tools and techniques from propositional calculus. At the very least, the manipulation of identities is much simpler both conceptually and practically than the use of predicate calculus, which requires working with quantified conditions.

As we mentioned in the Introduction, before 1975 it was thought that the equations defining OML were the only ones holding in HL. Then Alan Day discovered the orthoarguesian law, which is an equation that holds in any Hilbert lattice but not in all OMLs [31]. Since then, much progress has been made in finding many new equations that hold in HL and are independent from the others.

By Birkhoff's HSP theorem [45, p. 2], the family HL is not an equational variety, since a finite sublattice is not an HL. A goal of studying equations that hold in HL is to find the smallest variety that includes HL, so that the fewest number of non-equational (quantified) conditions such as those in Def. 2.3.1 (p. 19) will be needed to complete the specification of HL.

A summary of the equations known so far is given in Table 1.1. The equations fall into three major categories: geometry-related, state-related, and vector-state-related. The last hold in the $\mathscr{C}(H)$ of all "quantum" Hilbert spaces, i.e. those with real, complex, or quaternion fields but not necessarily with other fields.

Table 1.1: Summary of known equations holding in the in the $\mathscr{C}(H)$ of all (quantum) Hilbert spaces.

Equation	Variety	Based on	Definition
Orthoarguesian	4OA	geometry	Eq. (4.30) (p. 45)
Generalized OA	nOA, $n \geq 3$	geometry	Eq. (4.24) (p. 44)
Mayet's $\mathscr{E}_A$	$\mathscr{E}_A$	geometry	Ref. [82]
Godowski	nGO, $n \geq 3$	states	Th. 5.1.3 (p. 79)
Mayet-Godowski	MGO	states	Def. 5.2.1 (p. 82)
Mayet's E equations	E_n, $n \geq 3$	vector states	Eqs. (5.43), (5.44) (p. 98)

The relationships among the above lattice classes (equational varieties) that satisfy them is shown in Fig. 1.1 (p. 9). In addition, we show the modular law [Sec. 7.2 (p. 113)], the Arguesian law [Sec. 7.3 (p. 123)], and the n-Arguesian laws (n >2) [Sec. 7.4 (p. 140)] that hold for finite-dimensional Hilbert spaces. We also include M- and O-symmetric lattices [Sec. 6.1 (p. 100)] for comparison, although currently they are not known to be equational varieties.

The geometry-related equations are derived using the properties of vectors and subspace sums that hold in a Hilbert space. They include Day's original orthoarguesian equation, the generalized orthoarguesian equations, and Mayet's $\mathscr{E}_A$ equations.

In Ch. 4, we explore the nOA-related equations in much detail and obtain many new results. Although it still has not been solved, we show what progress has been achieved towards answering the **orthoarguesian identity conjecture**, Conjecture 4.5.2 (p. 63).

The state-related equations are derived by imposing states (probability measures) onto Hilbert lattices, and include Godowski's equations and Mayet-Godowski equations. [The justification for doing so is that such states can be defined in Hilbert space, and we map them back to HL via the ortho-isomorphism of Th. 2.3.3 (p. 20).] These equations are derived by finding finite OMLs that do not admit the "strong set of states" condition [Def. 2.4.3 (p. 22)] that Hilbert lattices do admit, then analyzing the strong set of states failure in a prescribed way in order to derive an equation holding in HL but failing in the finite OML.

Vector-state-related equations are derived by imposing "states" onto HLs that map to Hilbert-space vectors instead of real numbers (again, justified by the fact that such "states" can

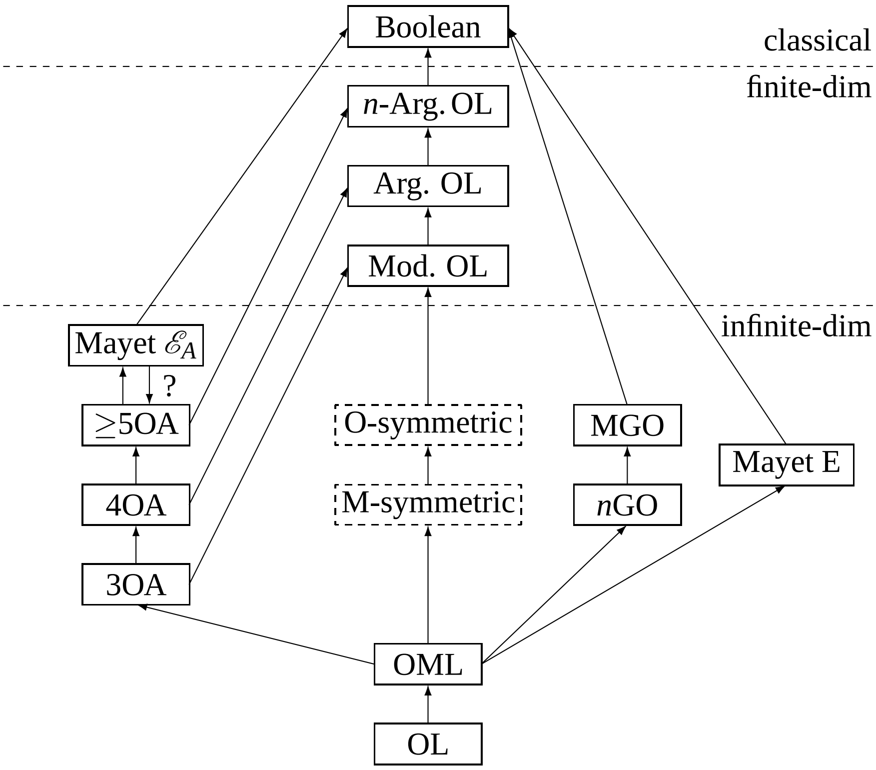


Figure 1.1: Relationship between known equational varieties holding in the closed subspaces of finite- and infinite-dimensional Hilbert spaces. (M- and O-symmetric lattices, Sec. 6.1 (p. 100), are not currently known to determine varieties.) Arrows point to smaller classes of lattices. There may be other relationships between these classes (inclusions) that are currently unknown and thus not shown. See Sec. 7.4 (p. 140) for the *n*-Arguesian laws.

be defined in Hilbert space). They do not always hold when the Hilbert-space field implied by the representation theorem (Th. 2.3.3) does not have characteristic 0. (“Characteristic 0” means, roughly, that the number 1 added to itself repeatedly grows without limit.) This remarkable property narrows down, from the equation alone, the possible fields for the Hilbert space. The real, complex, and quaternion fields of quantum mechanics have characteristic 0, so vector-state-related equations do hold in all “quantum” HLs that have the additional properties demanded by Solèr’s theorem in Th. 2.3.5 (p. 21). The vector-state-related equations known to date are Mayet’s E equations.

1.4 Finite-dimensional Hilbert space

The equations discussed above hold in the closed subspaces of all Hilbert spaces, whether finite or infinite. (We almost always use the adjective “infinite” to mean either finite or infinite.) Finite-dimensional Hilbert spaces are important for many problems in quantum mechanics, including most experiments involving particle states and in particular most approaches to quantum computation.[2] In Ch. 7, we discuss several laws that hold in finite-dimensional Hilbert spaces (but not necessarily in infinite-dimensional ones), starting with the modular law.

In Sec. 7.2.2 (p. 118), we study an inference from the modular law, found by von Neumann, which is closely connected to the orthoarguesian identity law, Eq. (4.90) p. 62, and therefore which may shed some light on the orthoarguesian identity conjecture. We prove that von Neumann’s inference [Th. 7.2.6 (p. 118)] is strictly weaker than the modular law in a lattice, but whether it is strictly weaker than the modular law in an OL remains an open problem. In that section, we also prove that if a condition fails in in a pentagon sublattice (which is the standard characterization for whether or not a lattice is modular), it does not necessarily imply that the condition is as strong as the modular law.

Sec. 7.3 (p. 123) collects known equivalents for the Arguesian law. We review a 184-node lattice that satisfies the modular law but fails the Arguesian law. This lattice, discovered in 1907 by Veblen and MacLagan-Wedderburn [121], seems to have been overlooked in subsequent literature, but apparently it is the only explicit finite lattice that has been published with this property. We also describe a procedure, starting from the skeleton of the standard infinite lattice (projective space) proof of Arguesian law independence, that could be used in a search for a smaller finite lattice counterexample. The technique is related to so-called MMPL diagrams

[2] There are also approaches to quantum computation using continuous variables i.e. infinite-dimensional Hilbert spaces [12] [55] [11] [10], although most of this work is in its infancy.

proposed by Pavičić [101, p. 102103-20, Def. III.2] that extend a finite lattice to satisfy certain additional conditions.

In Sec. 7.4 (p. 140), we show that higher-order Arguesian equations follow as a special case of the Hilbert space theorem from which the *n*OA equations are derived. An open problem is whether these are equivalent to the higher-order Arguesian laws mentioned in Ref. [34].

Finally, in Sec. 7.5 (p. 140), we discuss the law of Pappus that holds in projective planes and review work that has been done towards finding an equation that expresses this law. Such an equation could be useful because a division ring constructed from a Pappian geometry is necessarily commutative. A related goal would be to find a modification of the equation which would hold in infinite dimensions. In conjunction with Solèr's theorem, such an "orthopappian" equation would allow us to narrow down the field of the Hilbert space constructed from a Hilbert lattice to either the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$, eliminating quaternions as a possibility.

Chapter 2

PREREQUISITES

In this chapter, we summarize the necessary background for Chapters 3 through 7.

2.1 Hilbert spaces

We assume that the reader has some familiarity with the basic concepts of set theory. See, for example, Ref. [119]. Here we present a review of the necessary concepts, followed by the definitions needed for a complex Hilbert space.

A **set** is any mathematical object or collection of mathematical objects. The terms **element**, **member**, and set are synonymous. When a is an element of b, denoted $a \in b$, we say that a **belongs** to b and that b **contains** a. We will assume the axioms of ZFC set theory (Zermelo-Fraenkel set theory with the Axiom of Choice), wherein a **class** is an arbitrary collection of elements, and a **set** is a class which belongs to some other class. A **proper class** is one which is not a set. For example, the universe V containing every set is a proper class. The terms **collection** and **family** (such as the family of all algebras) often, but not necessarily, refer to proper classes.

A set (class) a is a **subset** (**subclass**) of another set (class) b, denoted $a \subseteq b$, when every member of a also belongs to b. In this case we say that b **includes** a.

A finite set with (not necessarily distinguished) elements $a_1, \ldots, a_n$ ($n \geq 0$) is denoted $\{a_1, \ldots, a_n\}$; the order is not important. $\{a, b\}$ is called an **unordered pair**, $\{a\}$ is called a **singleton**, and $\{\}$ or $\varnothing$ is the **empty set**. Note that $\{a, b\} = \{b, a\}$ and $\{a, a\} = \{a\}$.

An **ordered pair** $\langle a, b\rangle$ can be defined as $\{\{a\}, \{a, b\}\}$. An **ordered** n**-tuple** $\langle a_1, \ldots, a_n\rangle$ can be defined recursively, for $n \geq 3$, as the ordered pair $\langle\langle a_1, \ldots, a_{n-1}\rangle, a_n\rangle$. For our purposes, the precise definition is unimportant as long as we can talk unambiguously about the first member, second member, and so on.

A **relation** is a class of ordered pairs. The classes of first and second members of the ordered pairs of a relation are called its **domain** and **range** respectively. A **function** or **mapping** f is a relation such that the first member of each pair occurs exactly once. If the domain of f is A, we say that f is a function **on** A. If, in addition, B includes the range of f as a subclass, we say that f maps from A **into** (or just **to**) B, which we denote by $f : A \longrightarrow B$. When B equals the range of f, we say that f maps **onto** B and that f is **surjective**. When the second member of each pair of a function occurs exactly once, we say that the function is **one-to-one** or **injective**. A function that is both surjective and injective is called **bijective**. In general, following Ref. [119], relations and functions may be proper classes as well as sets.

A k-**Cartesian product** $A_1 \times \ldots \times A_k$ ($k \geq 2$) is the class of all k-tuples $\langle a_1, \ldots, a_k \rangle$ where $a_1 \in A_1, \ldots, a_n \in A_n$. Let A be a nonempty set and A^k be the k-Cartesian product $A \times \ldots \times A$ (k factors). An **operation** on A of **arity** k ($k \geq 2$), also called a k-**ary** or k-**place** operation on A, is a function (mapping) from A^k to A. For the special case $k = 1$, a 1-place operation is a mapping from A to A and is called a **unary**[1] operation. For the special case $k = 0$, a 0-**ary** or **nullary** operation is simply a member of A (a **constant operation**) rather than a function. A 2-place operation is usually called a **binary** operation. The arity of an operations is also called the number of **operands** or **arguments**.

An **algebra** is an ordered pair $A = \langle A_{\mathrm{O}}, F \rangle$ where A_{O} is a nonempty set (called the **base set** of the algebra) and F is a set of operations on A_{O}, which for us will always be finite in number [6, pp. 15]. When the (finite) set of operations is $F = \{f_1, \ldots, f_n\}$, we may express the algebra alternately as the ordered $(n+1)$-tuple $\langle A_{\mathrm{O}}, f_1, \ldots, f_n \rangle$, which also imposes an order on the set of operations; which notation is being used should be clear from context. For brevity, we may refer to the base set A_{O} of an algebra by the symbol A for the algebra itself, when it is clear from context.

The arity of the operands of an algebra $A = \langle A_{\mathrm{O}}, f_1, \ldots, f_n \rangle$ forms an ordered n-tuple of non-negative integers $\langle k_1, \ldots, k_n \rangle$, called the **type** of the algebra.

Let S_{O} be a subset of A_{O}, and let f be a k-ary operation on A_{O}. We say that an algebra S is **closed** under f if $f(a_1, \ldots, a_k) \in S_{\mathrm{O}}$ for all $a_1, \ldots, a_k \in S_{\mathrm{O}}$.

Definition 2.1.1. *[6, pp. 18] If S_{O} is a nonempty subset of A_{O}, then $S = \langle S_{\mathrm{O}}, F \rangle$ is called a* **subalgebra** *of the algebra $A = \langle A_{\mathrm{O}}, F \rangle$ iff S is closed under all $f \in F$.*

Subalgebras have the following property that we will use later [Th. 2.5.8 (p. 26)].

[1] Called **singulary** in Ref. [42, p. 39]

Lemma 2.1.2. *If M is a subalgebra of L, then any equation (identity) that holds in L will continue to hold in M. Equivalently, if an equation fails in M but holds in L, then M cannot be a subalgebra of L.*

Proof. See Ref. [101, Lemma II.2]. □

Note that the above lemma does not necessarily apply to quantified conditions. A quantified condition, such as superposition [Def. 2.3.1(3); Eq. (6.33)], that holds in a lattice may not hold in a sublattice. As a trivial example, the quantified condition "has more than two elements" does not hold in the two-element subalgebra consisting of 0 and 1.

A **group** is an algebra $\langle G, *\rangle$ where $*$ is a binary that is associative: $(a*b)*c = a*(b*c)$ for each a,b,c (in G); has a left identity element e: there is a e such that $e*a = a$ for all a; and has a left inverse for every element: for each a, there is a b such that $b*a = e$.

An **Abelian group** is a group whose operation is commutative: $a*b = b*a$.

A **complex vector space** is a set V, whose members are called **vectors**, together with an Abelian group operation $+$ (**vector sum**) on V and a function $\cdot$ (**scalar product**) from $\mathbb{C}\times V$ to V, where $\mathbb{C}$ denotes the set of complex numbers. The scalar product satisfies the identity law $1\cdot a = a$, the associative law $(x\cdot y)\cdot a = x\cdot(y\cdot a)$, and the distributive laws $x\cdot(a+b) = (x\cdot a)+(x\cdot b)$ and $(x+y)\cdot a = (x\cdot a)+(y\cdot a)$, where x and y are complex numbers and a and b are vectors. The symbol $+$ denotes either complex number addition or vector sum depending on context, which is never ambiguous; similarly, $\cdot$ denotes either complex number product or scalar product, either of which we may also denote using juxtaposition. We use 0 (the **zero vector**) to denote the group identity element of the vector sum and unary minus, $-$, to denote a vector's inverse. A vector a plus the inverse of a vector b, $a+-b$, is denoted $a-b$ and is called **vector difference**.

A **normed complex vector space** is a complex vector space V together with a map $\parallel\cdot\parallel$ (**norm**) from V to the real numbers $\mathbb{R}$. The norm is (the real number) 0 only when its vector argument is 0, it satisfies the multiplicative law $\parallel x\cdot a\parallel = \mid x\mid\cdot\parallel a\parallel$ (where $\mid x\mid$ denotes the absolute value of complex number x), and it satisfies the triangle inequality $\parallel a+b\parallel \leq \parallel a\parallel + \parallel b\parallel$.

A **metric space** is an ordered pair $\langle M, D\rangle$ where M is a set and D, a **distance function**, is a mapping from $M\times M$ to $\mathbb{R}$ (the set of real numbers) with the following properties for each x,y,z in M: $D(x,x) = 0$; $D(x,y) = D(y,x)$; $D(x,z) \leq D(x,y)+D(y,z)$; and $D(x,y) > 0$ whenever $x \neq y$.

The **induced metric space** of a normed complex vector space is the metric space whose base set is the vector space V and whose distance function for vectors x,y is $D(x,y) = \parallel x-y\parallel$.

A **sequence** is a function x_i on $\mathbb{N}$ (with values $x_1, x_2, \ldots$). A **Cauchy sequence** is a sequence x_i on a metric space such that for any $r \in \mathbb{R}$, there is a $k \in \mathbb{N}$ such that for all $m, n > k$, $D(x_m, x_n) < r$. A sequence x_i **converges** to a point y in a metric space iff for any $r \in \mathbb{R}$, there is a $k \in \mathbb{N}$ such that for all $n > k$, $D(y, x_n) < r$. A **complete** metric space is one in which all Cauchy sequences converge to a point in the metric space.

A **complex Banach space** is a normed complex vector space whose induced metric space is complete.

A **complex pre-Hilbert space** (also called a **complex inner product space**) is a normed complex vector space whose norm satisfies the **parallelogram law** for vectors x, y:

$$\| x+y \|^2 + \| x-y \|^2 = 2(\| x \|^2 + \| y \|^2) \tag{2.1}$$

A **complex Hilbert space** is a pre-Hilbert space that is also a Banach space.

We define the **inner product**[2] of vectors a and b as the complex number

$$(a,b) = \sum_{k=1}^{4} \frac{i^k}{4} \| a + i^k b \|^2 \tag{2.2}$$

where $i = \sqrt{-1}$.

The inner product has the following properties, which (if we chose consider it primitive rather than defined in terms of the norm) can also be considered its definition. For vectors x, y, z and complex number α, where $+$ denotes vector or complex number addition depending on context, juxtaposition represents scalar product or complex number product depending on context, and * denotes complex conjugate, we have [54, p. 129]:

$$(x+y,z) = (x,z)+(y,z) \tag{2.3}$$

$$(\alpha x,y) = \alpha(x,y) \tag{2.4}$$

$$(x,y) = (y,x)^* \tag{2.5}$$

$$(x,x) \geq 0 \tag{2.6}$$

$$(x,x) = 0 \quad \Leftrightarrow \quad x = 0 \tag{2.7}$$

A **subspace** of a vector space V is a subset S which contains the zero vector $0 = x - x$ (where

[2] By defining the inner product in terms of the norm, the norm becomes a primitive operation on a Hilbert space. The advantage of doing this is that Hilbert spaces become a subclass of Banach spaces, and both will have the same operations. The standard inner product properties can be recovered from this definition; see, for example, [111, p. 361].

x is any vector) and such that for any $x, y \in S$ and $\alpha \in \mathbb{C}$, $x + \alpha y \in S$.

The **orthogonal complement** (also called **orthocomplementation**) $S^\perp$ of a subspace S of a vector space V is the set of all vectors in V orthogonal to all vectors in S.

A subspace S is a **closed subspace** when all Cauchy sequences converge to a point in the subspace. Equivalently, a closed subspace is any subset S which equals its **closure** (double orthogonal complement) i.e. $S = S^{\perp\perp}$.

The **subspace sum** of two subspaces S and T, denoted $S + T$, is the set of all vectors $x + y$ where $x \in S$ and $y \in T$. The **join** of two subspaces S and T, denoted $S \vee T$, is the closure of their subspace sum i.e. $(S + T)^\perp$.

In quantum mechanics, the complex Hilbert spaces (i.e. Hilbert spaces over the field of complex numbers) described above are the ones of most practical importance. There also exist more general Hilbert spaces over general division rings (skew fields), in particular the three **classical fields** of real numbers, complex numbers, and quaternions, and more generally over any division ring (skew field). As a technicality, any such division ring must be accompanied by an additional unary operation "$*$" called **involution**, with the properties $(x + y)^* = x^* + y^*$, $(xy)^* = x^* y^*$, and $x^{**} = x$. Such a division ring is called a $*$**-field** (star field). In the case of complex numbers, the involution is the complex conjugate. For more information, the reader may consult Ref. [40, p. 205].

2.2 Lattice structures

We briefly recall the lattice theory definitions we will need. For further information, see Refs. [6], [76], [107], and [105].

Definition 2.2.1. [8] *A* **lattice** (Lat) *is an algebra $L = \langle L_{\mathrm{O}}, \wedge, \vee \rangle$ such that the following conditions are satisfied for any $a, b, c \in L_{\mathrm{O}}$:*

$$a \vee b = b \vee a \qquad a \wedge b = b \wedge a \tag{2.8}$$

$$(a \vee b) \vee c = a \vee (b \vee c) \qquad (a \wedge b) \wedge c = a \wedge (b \wedge c) \tag{2.9}$$

$$a \wedge (a \vee b) = a \qquad a \vee (a \wedge b) = a \tag{2.10}$$

In the above definition, Lat denotes the equational variety (class of all algebras) determined by the defining equations. When we say "L is a lattice" or "L is a Lat," we mean that it is a member of the (proper) class Lat. Similarly, in subsequent definitions, OL, etc. will denote the corresponding equational varieties.

Theorem 2.2.2. [8] *The binary relation* $\le$ *defined on* L *as*

$$a \le b \stackrel{\text{def}}{\Leftrightarrow} a = a \wedge b \tag{2.11}$$

is a partial ordering.

Definition 2.2.3. [9] *An* **ortholattice** (OL) *is an algebra* $L = \langle L_O, ', \wedge, \vee, 0, 1\rangle$ *such that the triple* $\langle L_O, \wedge, \vee\rangle$ *is a lattice and* $'$ *is a unary operation called* **orthocomplementation** *that satisfies the following conditions for* $a, b \in L_O$ *(*a' *is called the* **orthocomplement** *of* a*):*

$$a \vee a' = 1, \qquad a \wedge a' = 0 \tag{2.12}$$

$$a \le b \ \Rightarrow\ b' \le a' \tag{2.13}$$

$$a'' = a \tag{2.14}$$

Definition 2.2.4. *We define the* **classical implication** $a \to_0 b$ *and the* **quantum implications** $a \to_i b$ *(*$i = 1, ..., 5$*) as follows:*[3]

$$a \to_0 b \stackrel{\text{def}}{=} a' \vee b \qquad \text{(classical)} \tag{2.15}$$

$$a \to_1 b \stackrel{\text{def}}{=} a' \vee (a \wedge b) \qquad \text{(Sasaki)} \tag{2.16}$$

$$a \to_2 b \stackrel{\text{def}}{=} b \vee (a' \wedge b') \qquad \text{(Dishkant)} \tag{2.17}$$

$$a \to_3 b \stackrel{\text{def}}{=} ((a' \wedge b) \vee (a' \wedge b')) \vee \big(a \wedge (a' \vee b)\big) \qquad \text{(Kalmbach)} \tag{2.18}$$

$$a \to_4 b \stackrel{\text{def}}{=} ((a \wedge b) \vee (a' \wedge b)) \vee \big((a' \vee b) \wedge b'\big) \qquad \text{(non-tollens)} \tag{2.19}$$

$$a \to_5 b \stackrel{\text{def}}{=} ((a \wedge b) \vee (a' \wedge b)) \vee (a' \wedge b') \qquad \text{(relevance)} \tag{2.20}$$

The classical implication $\to_0$ is the only one of the six that does not satisfy the **Birkhoff-von Neumann requirement** [49, p. 238] in all OMLs:

$$a \le b \quad \Leftrightarrow \quad a \to_i b = 1, \qquad i = 1, \ldots, 5 \tag{2.21}$$

Th. 3.1.1 below (p. 27) shows that, in any OL, the Birkhoff-von Neumann requirement is equivalent to the OML law for quantum implications $i = 1 \ldots 5$. If we set $i = 0$ in Eq. (2.21), we end up with a condition equivalent to the distributive law, which is why we call $\to_0$ "classical".

[3] These are the names given in [95], except that $a \to_1 b$ was called "Mittelstaedt." In other literature, it is called "quasi-implication" [85, Eq. (3.4) on p. 1361] and the "Sasaki hook" [37, p. 322]. The relevance implication $\to_5$ has also been called the "Kotas-Kalmbach hook" [37, p. 322].

The Sasaki implication (also called the **Sasaki hook** [37, pp. 312,322]) is frequently used, and we will often omit its subscript.

Definition 2.2.5.

$$a \to b \overset{\text{def}}{=} a \to_1 b \tag{2.22}$$

Definition 2.2.6. *The following operation is called* **equivalence**.

$$a \equiv b \overset{\text{def}}{=} (a \wedge b) \vee (a' \wedge b') \tag{2.23}$$

Definition 2.2.7. [97, 98] *An* **orthomodular lattice** (OML) *is an* OL *in which the following condition (the* **orthomodular law**, *also called the* OML **law**) *holds:*

$$a \equiv b = 1 \quad \Rightarrow \quad a = b. \tag{2.24}$$

The equivalence of this definition to the other definitions in the literature follows from the fact that Eq. (2.24) holds in all OMLs and fails in the non-OML lattice O6, which we show in the form of a Hasse diagram [Def. 2.5.1 below (p. 22)] in Fig. 2.1a). This means that it implies the OML law by Theorem 2 of [49, p. 22]. There are many other equivalent formulations of the OML law, which can proved by showing that they hold in all OMLs and that they fail in lattice O6. Some of these are given in Theorem 3.1.1 below.

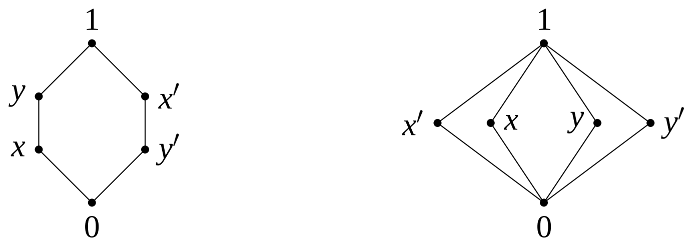


Figure 2.1: (a) Lattice O6; (b) Lattice MO2.

Definition 2.2.8. [125] *We say that a and b* **commute** *in an* OML, *and write aCb, when the following equation holds:*

$$a \wedge (a' \vee b) \le b \tag{2.25}$$

We call C the **commutativity relation**.

For later use, we define modular lattices and Boolean algebras.

Definition 2.2.9. *An* **orthocomplemented modular lattice** (MOL) *is an ortholattice in which the* **modular law** $b \le a \Rightarrow a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ *holds [99, p. 192]. A* **Boolean algebra** (BA) *is an ortholattice in which* **distributive law** $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ *holds.*

The above classes satisfy the proper subclass relations BA $\subset$ MOL $\subset$ OML $\subset$ OL $\subset$ Lat. In a BA, all six implications of Def. 2.2.4 (p. 17) equal each other.

2.3 Hilbert lattices

Our primary interest is in the subclass of OML called HL (Hilbert lattices).

Definition 2.3.1. *An orthomodular lattice that satisfies the following conditions is a* **Hilbert lattice** (HL).

1. Completeness: *The meet and join of any subset of an* HL *exist.*

2. Atomicity: *Every non-zero element in an* HL *is greater than or equal to an atom. (An* **atom** *a is a non-zero lattice element with* $0 < b \le a$ *only if* $b = a$*.)*

3. Superposition principle: *(The atom c is a* **superposition** *of the atoms a and b if $c \neq a$, $c \neq b$, and $c \le a \vee b$.)*

   (a) *Given two different atoms a and b, there is at least one other atom c, $c \neq a$ and $c \neq b$, that is a superposition of a and b.*

   (b) *Given atoms a and b and a lattice element c such that $a \wedge c = 0$, $a \le b \vee c$ implies $b \le a \vee c$. In particular, if a is a superposition of b and (atom) c, then b is a superposition of a and c.*

4. Minimum height: *The lattice contains at least three elements a, b, c satisfying:* $0 < a < b < c < 1$.

These conditions imply an infinite number of atoms in HL, as shown by Ivert and Sjödin [44].

With suitably defined operations, the set of closed subspaces of a Hilbert space H, $\mathscr{C}(H)$, can be shown to be a Hilbert lattice (a member of HL). The meet operation $a \wedge b$ corresponds to the set intersection $H_a \cap H_b$ of subspaces H_a, H_b of H; the ordering relation $a \le b$ corresponds to $H_a \subseteq H_b$; the join operation $a \vee b$ corresponds to the smallest closed subspace of H containing the set union $H_a \cup H_b$; and the orthocomplementation operation a' corresponds to $H_a^{\perp}$, the set of vectors orthogonal to all vectors in H_a. Within Hilbert space there is also an operation which has no parallel in the Hilbert lattice: the sum of two subspaces $H_a + H_b$, which is defined as the set of sums of vectors from H_a and H_b. We also have $H_a + H_a^\perp = H$ i.e. the subspace that equals the whole of Hilbert space itself. One can define all the lattice operations on a Hilbert space itself following the above definitions ($H_a \wedge H_b = H_a \cap H_b$, etc.). Thus we have $H_a \vee H_b = \overline{H_a + H_b} = (H_a + H_b)^{\perp\perp} = (H_a^\perp \cap H_b^\perp)^\perp$ [43, p. 175], where $\overline{H_c}$ is the closure of H_c, and therefore $H_a + H_b \subseteq H_a \vee H_b$. When H is finite-dimensional or when the closed subspaces H_a and H_b are orthogonal to each other then $H_a + H_b = H_a \vee H_b$ [35, pp. 21-29] [49, pp. 66,67] [86, pp. 8-16].

Using these operations, it is straightforward to verify that closed subspaces $\mathscr{C}(H)$ of a finite- or infinite-dimensional Hilbert space H form an OML [49, pp. 66,67] and more specifically an HL [4, pp. 105–108,166,167]. (In the case of a finite Hilbert space, $\mathscr{C}(H)$ is also an MOL [4, p. 107].) Specifically, we have the following theorem.

Theorem 2.3.2. *Let H be a finite- or infinite-dimensional Hilbert space over a field K and let*

$$\mathscr{C}(H) \stackrel{\text{def}}{=} \{X \ \subseteq H \mid X^{\perp\perp} = X\} \tag{2.26}$$

be the set of all closed subspaces of H. Then $\mathscr{C}(H)$ is a Hilbert lattice relative to:

$$a \wedge b = X_a \cap X_b \ \text{ and } \ a \vee b = (X_a + X_b)^{\perp\perp}. \tag{2.27}$$

A more difficult problem is to determine, given an HL, how much of Hilbert space can be reconstructed from it. An **isomorphism** is a bijection between two lattices that preserves the lattice ordering (or equivalently the meet and join operations). An **ortho-isomorphism** is an isomorphism that also preserves the orthocomplement operation. One can prove the following representation theorem [56, 57, 120].

Theorem 2.3.3. *For every Hilbert lattice* (HL)*, there exists a field K and a Hilbert space H over K such that the set of closed subspaces of the Hilbert space, $\mathscr{C}(H)$, is orthoisomorphic to* HL. *(Note that multiplication is not necessarily commutative in this field, which is more properly called a "division ring" or "skew field.")*

In order to determine the field over which the Hilbert space in Theorem 2.3.3 is defined, we make use of a theorem proved by Maria Pia Solèr [115, 41]. First, we need a definition.

Definition 2.3.4. *Let p and q be orthogonal atoms in a Hilbert lattice and c be an atom different from p and q such that $c \le p \vee q$. Let x be any atom such that $x \nleq p \vee q$. Let y an atom different* *from x and p such that $y \le x \vee p$. Define $d_1 = (c \vee y) \wedge (q \vee x)$ and $d_2 = (p \vee d_1) \wedge (q \vee y)$. Then $(x \vee d_2) \wedge (p \vee q)$ is the (unique)* **harmonic conjugate** *of c with respect to p and q.*

Now we can state the following application of Solèr's theorem to an HL lattice [40, p. 221, Th. 4.1].

Theorem 2.3.5. *The Hilbert space H from Theorem 2.3.3 is an infinite-dimensional Hilbert space defined over a real, complex, or quaternion (skew) field if the following conditions are met:*

- Infinite orthogonality: *The* HL *contains a countably infinite sequence of orthogonal atoms $p_i, i = 1, 2, \ldots$*
- Harmonic conjugate condition: *The* HL *contains a corresponding sequence of atoms $c_i \le p_i \vee p_{i+1}$ such that the harmonic conjugate of c_i with respect to p_i, p_{i+1} equals $c_i' \wedge (p_i \vee p_{i+1})$.*

In this way we can obtain a full Hilbert space, but as we can see the axioms for the Hilbert lattices that we used for this purpose are rather involved quantified (first-order) statements. In Chapters 4 (p. 37) through 6 (p. 100) below, we will look at some (zeroth-order) equations that may eventually replace some of the quantified conditions or allow weakened versions of them.

One feature of a Hilbert lattice is that the distributive law does not hold when the dimension of the Hilbert space is greater than one.

Theorem 2.3.6. *The distributive law, $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$, fails for some closed subspaces a, b, c of any HL whose underlying Hilbert space has dimension greater than one.*

Proof. We prove the result for $\mathscr{C}(H)$, then use the orthoisomorphism of Th. 2.3.3 (p. 20). Let v_b and v_c be two non-zero, non-co-linear vectors of the Hilbert space. Let $a = \mathrm{span}(v_b + v_c)$, $b = \mathrm{span}(v_b)$, and $c = \mathrm{span}(v_c)$. Since $v_b + v_c$ is not colinear with either v_b or v_c, we have $a \wedge b = 0$ and $a \wedge c = 0$, so $(a \wedge b) \vee (a \wedge c) = 0 \vee 0 = 0$. On the other hand, $b \vee c$ spans a 2-dimensional subspace containing $v_b + v_c$. Therefore, $a \wedge (b \vee c) = a \neq (a \wedge b) \vee (a \wedge c)$. ☐

2.4 States on lattices

Definition 2.4.1. *A* state *(also called* probability measure *or simply* probability *[51, 49, 50, 51, 58]) on a lattice L is a function $m : L \longrightarrow [0,1]$ such that $m(1) = 1$ and $a \perp b \;\Rightarrow\; m(a \cup b) = m(a) + m(b)$, where $a \perp b$ means $a \le b'$.*

Lemma 2.4.2. *The following properties hold for any state m:*

$$m(a)+m(a')=1 \tag{2.28}$$

$$a \le b \;\Rightarrow\; m(a) \le m(b) \tag{2.29}$$

$$0 \le m(a) \le 1 \tag{2.30}$$

$$m(a_1)=\cdots=m(a_n)=1 \;\Leftrightarrow\; m(a_1)+\cdots+m(a_n)=n \tag{2.31}$$

$$m(a_1\cap\cdots\cap a_n)=1 \;\Rightarrow\; m(a_1)=\cdots=m(a_n)=1 \tag{2.32}$$

Definition 2.4.3. *A set S of states on a lattice L is called a* **strong set of quantum states** *(or just a* **strong set of states***) iff*

$$(\forall a,b\in L)(\exists m\in S)((m(a)=1\Rightarrow m(b)=1)\Rightarrow a\le b)\,. \tag{2.33}$$

We assume that L contains more than one element and that an empty set of states is not strong.

2.5 Greechie diagrams

Lattice counterexamples serve as important tools for proving the independence of various equations that hold in Hilbert lattices. There is a compact notation for finite OML lattices, called Greechie diagrams, which we will describe in this section.

Definition 2.5.1. *A* **Hasse diagram** *is a graphical representation of a lattice where an element y is drawn above and connected to an element x if and only if $y \ge x$ and y is the least such element (i.e. y* covers *x).*

The Hasse diagram for any OML consists of connected Hasse diagrams for its maximal Boolean subalgebras, called **blocks**. Such Hasse diagrams have a shorthand notation called Greechie diagrams.

Definition 2.5.2. Greechie diagram *[84, Def. 2.5]. A* **Greechie diagram** *is a notation that represents the atoms within each block of an* OML *as dots connected by a line or smooth curve. The following conditions must be satisfied.*

1. *All blocks share a common* 0 *and* 1*.*

2. *If an atom a belongs to an intersection of blocks and therefore to both of them, then the blocks also share a';*

3. *Blocks contain 3 or more atoms.*

4. *Two blocks may not share more than one atom.*

In terms of graph theory, a Greechie diagram is a type of **hypergraph**, which is a structure consisting of **edges** (the Greechie diagram's lines) containing **vertices** (the Greechie diagram's atoms) and connected at some of the vertices.

This definition is equivalent to Richard Greechie's original definition in 1971 [30]. Recently, the term Greechie diagram has been used to denote other kinds of hypergraphs related to pastings [24, 23, 89], Kochen-Specker sets [117], test spaces [3], etc. For these hypergraphs, condition 4 above does not necessarily hold, but for our elaboration and the generation of our diagrams it is essential. Since this condition is also present in the original definition, it is the one that we use.

Definition 2.5.3. *A* **loop** *of order $n > 2$ is a set of blocks $B_1, \ldots, B_n$ such that B_i shares an atom with B_{i+1} for $i < n$ and B_1 shares an atom with B_n.*

Lemma 2.5.4. *[30] A Greechie diagram represents an orthomodular lattice if and only if the order of every loop of its blocks is at least 5.*

This lemma is known as the **Loop Lemma** [49, p. 38].

Definition 2.5.5. *The unique orthomodular lattice represented by a Greechie diagram satisfying the Loop Lemma is called a* **Greechie lattice**.

The Loop lemma does not hold for lattices represented by the pasting hypergraphs mentioned above but only for the original Greechie diagrams and lattices as defined by Def. 2.5.2.

The Hasse diagrams for the Boolean algebras corresponding to 2-, 3-, and 4-atom blocks are shown in Fig. 2.2. The Greechie diagram for a given lattice may be drawn in several equivalent ways: Fig. 2.3 shows the same Greechie diagram drawn in two different ways, along with the corresponding Hasse diagram. From the definitions we see that the ordering of the atoms on a block does not matter, and we may also draw blocks using arcs as well as straight lines as long as the blocks remain clearly distinguishable.

We use a special ASCII notation to represent Greechie diagrams and other hypergraphs for our computer programs such as `latticeg.c`, which tests whether a given equation holds in a list of Greechie diagrams.

Definition 2.5.6. **MMP encoding** *represents the vertices of a hypergraph (and in particular the atoms of a Greechie diagram) by means of alphanumeric and other printable ASCII characters.*

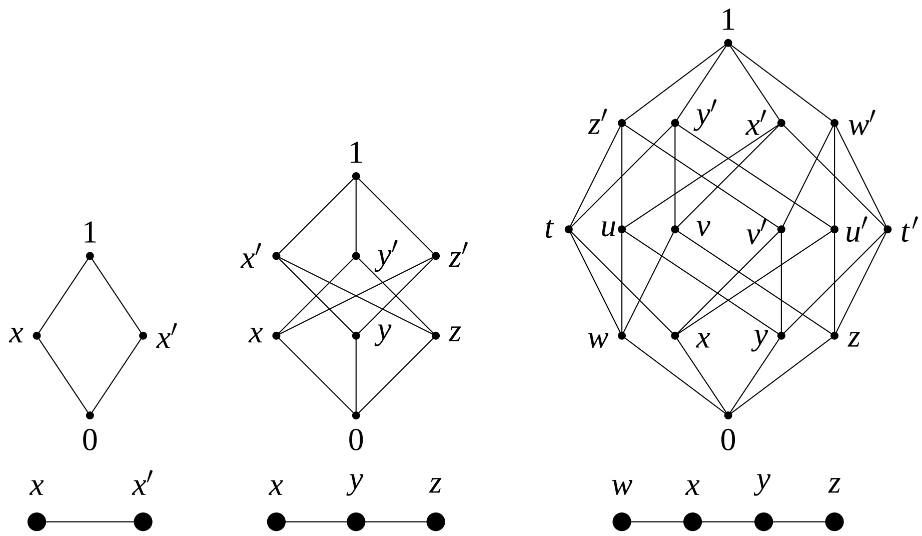


Figure 2.2: Greechie diagrams for Boolean lattices 2^2, 2^3, and 2^4, labeled with the atoms of their corresponding Hasse diagrams shown above them. (2^4 was adapted from [6, Fig. 18, p. 84].)

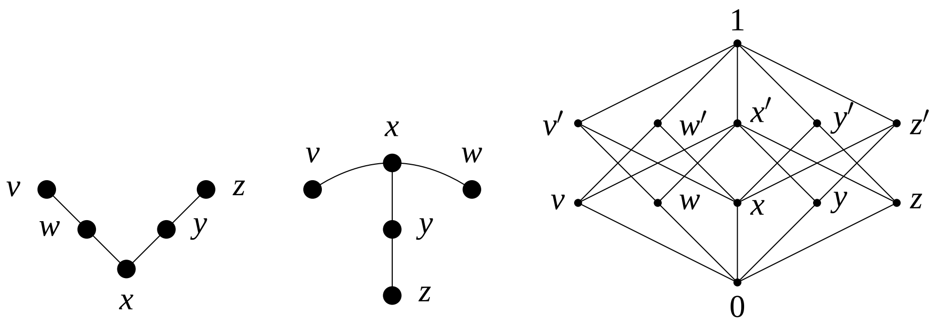


Figure 2.3: Two different ways of drawing the same Greechie diagram, and its corresponding Hasse diagram.

Each hypergraph vertex (lattice atom) is represented by one of the following characters: 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z ! " # $ % & ' ( ) * - / : ; < = > ? @ [\ ] ⊖ ¨ { | } ˜ *, and then again all these characters prefixed by '+', then prefixed by '++', etc. There is no upper limit on the number of atoms that can be represented.*

Each block (hypergraph edge i.e. continuous line connecting dots in a Greechie diagram) is represented by a string of characters that represent atoms. Blocks are separated by commas. The order of the blocks is irrelevant, although sometimes it is useful to present them in a canonical form for comparisons and searches, or to have them start with blocks forming the biggest loop to facilitate their possible drawing. A string ends with a full stop (i.e. a period). Skipping of characters is allowed.

The initialism "MMP" stands for the authors of Ref. [73], where the notation was first introduced.

We will usually provide the MMP encodings for the Greechie diagrams that follow. This way, the reader can, if desired, duplicate the associated results using the programs described in Appendix A (p. 145).

Greechie diagrams are useful for finding finite counterexamples to OML conjectures. However, it is important to note that they are not, in general, subalgebras [Def. 2.1.1 (p. 13)] of any Hilbert lattice; in particular, any Greechie diagram with a chain of three or more blocks cannot be a subalgebra of HL of dimension three or greater. We show this for the 3-block chain of Fig. 7.1 (p. 115). For larger lattices, we prove it in the same way by considering an embedded 3-block chain and taking into account the Loop Lemma [49, p. 43], which states that any loop in a Greechie diagram must contain 5 or more blocks (meaning that the atoms on the extremities of a 3-chain block will not "interfere" with each other).

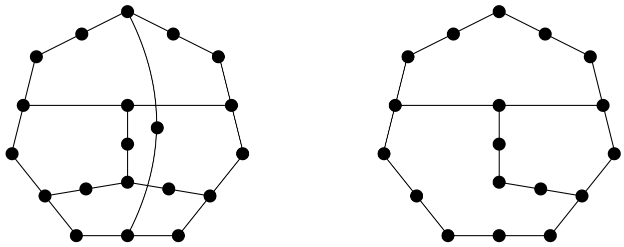
Theorem 2.5.7. *Consider the Greechie diagram whose MMP encoding is* `123,345,567.` *[the Dilworth lattice, Fig. 7.1 below (p. 115)] that pastes a sequence of 3* 2^3 *Boolean algebras* `123.`, `345.`, *and* `567.` *(*`1` *through* `7` *label the atoms). This Greechie diagram is not a subalgebra of a Hilbert lattice of dimension 3 or greater.*

Proof. Consider the join of atoms `1` and `7`. In the Greechie diagram, this is the lattice unit (as can be seen from its Hasse diagram. However, in any Hilbert lattice, the join of any two atoms corresponds to a 2-dimensional subspace, which for a subspace lattice of dimension greater than 2 is not the whole space (lattice unit). Thus the requirement that a subalgebra have the same operation values as its parent algebra is not satisfied. ☐

Perhaps somewhat counterintuitively, the removal of a block from a Greechie diagram does not necessarily result in a subalgebra of the original Greechie diagram.

Theorem 2.5.8. *A subdiagram of a Greechie diagram does not necessarily correspond to a subalgebra of the parent diagram.*

Proof. [73, p. 2403] The Greechie diagram of Fig. 2.4(b)[4] is obviously a subdiagram of the one of Fig. 2.4(a).[5] However, the 4OA law [Eq. (4.30), p. 45 below] passes in Fig. 2.4(a) but fails in its subdiagram Fig. 2.4(b). (This can be verified with, for example, our program `latticeg.c`.) Thus by Lemma 2.1.2 (p. 14), the lattice of Fig. 2.4(b) is not a subalgebra of Fig. 2.4(a). ☐



(a) (b)

Figure 2.4: (a) Lattice L38+; (b) lattice L38, which is a subdiagram but not a subalgebra of L38+.

[4] `123,345,567,789,9AB,BCD,DE1,CF4,FGH,HI6.` is an MMP encoding for Fig. 2.4(b).
[5] `123,345,567,789,9AB,BCD,DE1,CF4,FGH,HI6,AHJ,1K8.` is an MMP encoding for Fig. 2.4(a).

Chapter 3

ORTHOMODULAR LATTICES

3.1 Basic OML properties

The equational theory of OMLs has never been shown to be decidable (except for equations with at most two variables), and proofs can be difficult to find. In this chapter we collect a number of results that will prove useful later or are of interest for their own sake.

Equations with two variables can be proved automatically in several ways. When given a two-variable term (polynomial), the program `beran.c` will return one of the 96 canonical expressions it is equivalent to, and when given a two-variable equation or inequality, it will return "1" iff the equation or inequality is true. The program `lattice.c` will prove both two-variable equations and two-variable conditions (inferences with hypothesis): if a two-variable equation or condition passes all lattices up to (but not necessarily including) the non-OML O6 [Fig. 2.1a (p. 18)], then it holds in all OMLs. If it also fails O6, it is equivalent to the OML law Eq. (2.24) (p. 18). (The programs `beran.c`, `lattice.c`, and all others that we reference are described in Appendix A [p. 145])

We usually omit proofs of two-variable conditions because they can be proved automatically in this way. Whenever conditions with three or more variables are known to have appeared in the literature, we provide their literature references and usually omit their proofs; otherwise, we show their explicit proofs.

First, we give several equivalents to the OML law. Most can be found in the literature, and the others (with two variables) can be easily proved as described above.

Theorem 3.1.1. *Any any* OL*, each of the following conditions is equivalent to the* OML *law,* *Eq. (2.24):*

$$a \equiv b = 1 \quad \Leftrightarrow \quad a = b. \tag{3.1}$$
$$a \le b \quad \Rightarrow \quad a \vee (a' \wedge b) = b \tag{3.2}$$
$$a \le b \quad \Rightarrow \quad b \wedge (b' \vee a) = a \tag{3.3}$$
$$a \vee (a' \wedge (a \vee b)) = a \vee b \tag{3.4}$$
$$a \wedge (a' \vee (a \wedge b)) = a \wedge b \tag{3.5}$$
$$a \to_i b = 1 \quad \Leftrightarrow \quad a \le b, \qquad i = 1, \ldots, 5 \tag{3.6}$$
$$a \to_i b = a \to_j b \;\Rightarrow\; aCb, \qquad i, j = 0, \ldots, 5,\ i \ne j \tag{3.7}$$
$$aCb \quad \& \quad aCc \quad \Rightarrow \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \tag{3.8}$$
$$b \le a \quad \& \quad c \le a' \quad \Rightarrow \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \tag{3.9}$$
$$a \le b \quad \Rightarrow \quad a \vee (b \wedge a') = b \vee (a \wedge b') \tag{3.10}$$
$$a \le b \quad \Rightarrow \quad \exists c (a \le c' \;\;\&\;\; b = a \vee c). \tag{3.11}$$

Proof. For Eqs. (3.2) and (3.4), see Ref. [49, p. 22]. Eq. (3.3) follows from Eq. (3.2) by taking the orthocomplement of both sides of the conclusion then applying De Morgan's laws. For Eq. (3.6), see Ref. [102, Th. 3.2]. For Eq. (3.8), see Ref. [107, Definition 7]. For Eq. (3.9), see Ref. [99, p. 193, Def. 3.8]. For Eq. (3.10), see Ref. [61, p. 250, Th. 3(β_1)].

For Eq. (3.11), see Th. 29.13(ε) of Ref. [59, p. 132]. It is also instructive to see a direct, explicit proof of this condition as an example of how a an existentially quantified condition can be transformed into an equation and vice versa.

First, we show that the OML law follows from Eq. (3.11), which we will write as $a \le d \Rightarrow \exists c (a \le c' \;\&\; d = a \vee c)$. Assume $a \le b$. Since $a \le a \vee b'$, we have $a \le (a \vee b') \wedge b$. Substituting $(a \vee b') \wedge b$ for d, the hypothesis of we Eq. (3.11) is satisfied, and we obtain

$$a \le b \quad \Rightarrow \quad \exists c (a \le c' \;\;\&\;\; (a \vee b') \wedge b = a \vee c). \tag{3.12}$$

Now in any OL,

$$(a \vee b') \wedge b = a \vee c \quad \Rightarrow \quad c \vee a \le b \tag{3.13}$$

Adding a disjunct to the right of the conclusion and removing a disjunct from the left, it follows that

$$(a \vee b') \wedge b = a \vee c \quad \Rightarrow \quad c \leq a \vee b'. \tag{3.14}$$

And of course in any OL we have

$$a \leq c' \quad \Rightarrow \quad c \leq a'. \tag{3.15}$$

From Eqs. (3.13) and (3.15),

$$\begin{aligned}(a \vee b') \wedge b = a \vee c \ \& \ a \leq c' \quad &\Rightarrow \quad (c \vee a) \wedge c \leq a' \wedge b \\ &\Rightarrow \quad c \leq a' \wedge b. \end{aligned} \tag{3.16}$$

From Eqs. (3.16) and (3.14),

$$(a \vee b') \wedge b = a \vee c \ \& \ a \leq c' \quad \Rightarrow \quad c \leq (a \vee b') \wedge (a' \vee b).$$

Since $(a \vee b') \wedge (a' \wedge b) = 0$, we have

$$\begin{aligned}(a \vee b') \wedge b = a \vee c \ \& \ a \leq c' \quad &\Rightarrow \quad c = 0 \\ &\Rightarrow \quad (a \vee b') \wedge b = a \vee 0 \ \& \ a \leq 0' \\ &\Rightarrow \quad (a \vee b') \wedge b = a \end{aligned}$$

Applying the existential quantifier to both sides of this implication,

$$\begin{aligned}\exists c((a \vee b') \wedge b = a \vee c \ \& \ a \leq c') \quad &\Rightarrow \quad \exists c((a \vee b') \wedge b = a) \\ &\Rightarrow \quad (a \vee b') \wedge b = a \end{aligned} \tag{3.17}$$

For the last implication, we can remove the existential quantifier because c does not occur in the quantified expression. Chaining Eqs. (3.12) and (3.17), we conclude

$$a \leq b \quad \Rightarrow \quad (a \vee b') \wedge b = a,$$

which is the OML law Eq. (3.3).

For the converse, assume $a \leq b$. Let $c = a' \wedge b$. Then $a \leq c'$ in any OL, and $b = a \vee (a' \wedge b) = a \vee c$ by Eq. (3.2). Thus there is a c that satisfies Eq. (3.11). ☐

Next, we list some frequently-used properties of the commutativity relation aCb [Def. 2.2.8 (p. 18)].

Theorem 3.1.2. *The following conditions hold in all* OML*s:*

$$aCa \tag{3.18}$$

$$aC0 \tag{3.19}$$

$$aCb \;\Leftrightarrow\; a=(a\wedge b)\vee(a\wedge b') \tag{3.20}$$

$$aCb \;\Leftrightarrow\; a\wedge(a'\vee b)\le b \tag{3.21}$$

$$aCb \;\Leftrightarrow\; b\le a\to_n b,\quad n=1,3,4,5 \tag{3.22}$$

$$aCb \;\Leftrightarrow\; a'\le a\to_n b,\quad n=2,3,4,5 \tag{3.23}$$

$$aCb \;\Leftrightarrow\; bCa \tag{3.24}$$

$$aCb \;\Leftrightarrow\; a'Cb \;\Leftrightarrow\; aCb' \;\Leftrightarrow\; a'Cb' \tag{3.25}$$

$$a\le b \;\Rightarrow\; aCb \tag{3.26}$$

$$aCb \;\&\; aCc \;\Rightarrow\; aCb\wedge c \tag{3.27}$$

$$aCb \;\&\; aCc \;\Rightarrow\; aCb\vee c \tag{3.28}$$

$$aCb \;\Leftrightarrow\; a\to_i b=a\to_j b,\qquad i,j=0,\ldots,5;\ i\ne j \tag{3.29}$$

$$bCc \;\&\; aCb\wedge c \;\Rightarrow\; a\wedge bCc \tag{3.30}$$

$$bCc \;\&\; aCb\vee c \;\Rightarrow\; a\vee bCc \tag{3.31}$$

Eqs. (3.30) and (3.31) are known as the **Gudder-Schelp-Beran theorem (GSB)**.

Proof. For Eq. (3.20), see Theorem 3.7 of Ref. [6, p. 46]. For Eq. (3.21), see Eq. (2.6) of Ref. [76]. Eqs. (3.22) and (3.23) are easily proved with the assistance of a program such as `lattice.c`, as described above. For Eq. (3.29), see Ref. [107, p. 25, footnote 13] for the forward direction; the reverse direction can be proved with e.g. `lattice.c`. For Eqs. (3.30) and (3.31), see Theorem 4.2 of Ref. [6, p. 263]. The proofs for the others can also be found in Ref. [6]. □

Theorem 3.1.3. *If any two terms from the set* $\{a,b,c\}$ *commute, then the following distributive laws hold in all* OML*s:*

$$a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c) \tag{3.32}$$

$$a\vee(b\wedge c)=(a\vee b)\wedge(a\vee c) \tag{3.33}$$

This is known as the **Foulis-Holland theorem (F-H)***.*

Proof. See e.g. Ref. [49, p. 25]. ☐

Theorem 3.1.4. *If aCb, bCc, cCd, and dCa, then the following distributive laws hold in all* OML*s:*

$$(a\vee b)\wedge(c\vee d)=(a\wedge c)\vee(a\wedge d)\vee(b\wedge c)\vee(b\wedge d) \tag{3.34}$$

$$(a\wedge b)\vee(c\wedge d)=(a\vee c)\wedge(a\vee d)\wedge(b\vee c)\wedge(b\vee d) \tag{3.35}$$

This is known as the **Marsden-Herman lemma (M-H)***.*

Proof. See e.g. Lemma 7.2 in Ref. [49, p. 91]. ☐

The next lemma provides some technical results for use in subsequent proofs.

Lemma 3.1.5. *The following conditions hold in all* OML*s:*

$$(a\to b)\wedge a=a\wedge b \tag{3.36}$$

$$(a\to b)\wedge(a'\to b)=(a\to b)\wedge b=(a\wedge b)\vee(a'\wedge b) \tag{3.37}$$

$$(a'\to b)'\le a'\le a\to b \tag{3.38}$$

$$(a\to b)\to b=a'\to b \tag{3.39}$$

$$(a\to b)'\to b=a\to b \tag{3.40}$$

$$(a\to_i b)\vee(a\to_j b)=a\to_0 b,\ i,j=0,\ldots,4,\ i\ne j \tag{3.41}$$

$$(a\to_i b)\wedge(a\to_j b)=a\to_5 b,\ i,j=1,\ldots,5,\ i\ne j \tag{3.42}$$

$$a'\le b\quad\Rightarrow\quad b\le a\to b \tag{3.43}$$

$$a\wedge((a\to c)\vee b)\le c\quad\Leftrightarrow\quad b\le a\to c \tag{3.44}$$

$$a'\wedge(a\vee b)\le c\quad\Leftrightarrow\quad (a\to c)\wedge(a\vee b)\le c \tag{3.45}$$

$$a'\wedge(a\vee b)\le c\quad\Leftrightarrow\quad b\le a'\to c \tag{3.46}$$

Proof. See Lemma 4.6 of [76] for Eqs. (3.36)–(3.41) and (3.43)–(3.44).

For Eq. (3.42), we omit the easy proof.

For Eq. (3.45): If $a'\wedge(a\vee b)\le c$ then $a'\wedge(a\vee b)\le a'\vee c$, so $a\vee b=a\vee(a'\wedge(a\vee b))\le a\vee(a'\wedge c)=(a'\to c)$, so $(a\to c)\wedge(a\vee b)\le(a\to c)\wedge(a'\to c)=(a\wedge c)\vee(a'\wedge c)\le c$ using (3.37); conversely, since $a'\le a\to c$ we have $a'\wedge(a\vee b)\le(a\to c)\wedge(a\vee b)\le c$ by hypothesis.

For Eq. (3.46): If $a' \wedge (a \vee b) \leq c$ then $a' \vee (a \vee b) \leq a' \wedge c$, so $b \leq a \vee b = a \vee (a' \wedge (a \vee b)) \leq a \vee (a' \wedge c) = a' \to c$; conversely, if $b \leq a' \to c$ then $a \vee b \leq a \vee (a' \wedge c)$, so $a' \wedge (a \vee b) \leq a' \wedge (a \vee (a' \wedge c)) = a' \wedge c \leq c$. $\square$

3.2 The Sasaki implication

The most frequent implication that we will use is the Sasaki implication of Def. 2.2.4 (p. 17), which is also the simplest non-classical (quantum) implication. Partly this is convention; any theorem using a Sasaki implication can be restated for the Dishkant implication since the latter just reverses and orthocomplements its arguments. The remaining three quantum implications are used much less frequently. The reason for that isn't clear; it is possible that since they are more complex, they simply haven't been studied as much. However, experience does seem to show that the Sasaki (or Dishkant) implication is the one that shows up more "naturally" in investigations of Hilbert lattice equations.

In this section, we will show some basic properties of the Sasaki implication. The results that haven't been published are accompanied by proofs.

The equality $a \to c = b \to c$ often arises in conjunction with the 3OA identity law described later in Sec. 4.5 (p. 62). The following two lemmas provide equivalences to this equality, and Corollary 3.2.5 below shows a way to infer the equality.

Lemma 3.2.1. *The following condition holds in all* OML*s:*

$$a' \to c = b' \to c \quad \Leftrightarrow \quad a \to c = b \to c \tag{3.47}$$

Proof. If $a \to c = b \to c$, then $(a \to c) \to c = (b \to c) \to c$. Since $(a \to c) \to c = a' \to c$, and similarly for b, it follows that $a' \to c = b' \to c$. The converse is proved similarly. $\square$

Lemma 3.2.2. *The following condition holds in all* OML*s:*

$$\begin{aligned} &((a \to c) \wedge (a' \to c)) \vee ((b \to c) \wedge (b' \to c)) \\ &\qquad = ((a \to c) \vee (b \to c)) \wedge ((a' \to c) \vee (b' \to c)) \\ &\Leftrightarrow \quad a \to c = b \to c \end{aligned} \tag{3.48}$$

Proof. If $a \to c = b \to c$, then by Eq. (3.47) $a' \to c = b' \to c$, and both sides of the left equality reduce to $(a \to c) \wedge (a' \to c)$.

Conversely, starting from the left side, we derive $b \wedge ((b\to c) \vee (a\to c)) \le c$ as follows:

$$\begin{aligned}
&b\wedge((b\to c) \vee (a\to c)) \\
&\quad \le ((a\to c) \vee (b\to c)) \wedge (b'\to c) && \text{since } b \le b'\to c \\
&\quad \le ((a\to c) \vee (b\to c)) \wedge ((a'\to c) \vee (b'\to c)) \\
&\quad = ((a\to c) \wedge (a'\to c)) \vee ((b\to c) \wedge (b'\to c)) && \text{by hypothesis} \\
&\quad = ((a\to c) \wedge c) \vee ((b\to c) \wedge c) && \text{since } (a\to c) \wedge (a'\to c) = (a\to c) \wedge c \\
&\quad \le c \vee c. \\
&\quad = c
\end{aligned}$$

Eq. (3.46) then gives $a\to c \le b\to c$. Swapping a and b, we similarly conclude $b\to c \le a\to c$ and thus $a\to c = b\to c$. ☐

The next lemma and theorem show a commutativity result for the Sasaki implication, followed by a corollary showing an example of its use.

Lemma 3.2.3. *The following conditions hold in all* OML*s:*

$$a\to c\; C\; b'\to c \quad \& \quad b\to c\; C\; a'\to c \quad \Rightarrow \quad a\to c\; C\; b\to c \tag{3.49}$$

$$a\to c\; C\; b'\to c \quad \& \quad b\to c\; C\; a'\to c \quad \Rightarrow \quad a'\to c\; C\; b'\to c \tag{3.50}$$

Proof. For Eq. (3.49): We have $a\to c\; C\; a'\to c$ by Eqs. (3.38), (3.26), and (3.25). From this and $a\to c\; C\; b'\to c$, it follows by Eq. (3.27) that

$$a\to c\; C\; (a'\to c) \wedge (b'\to c). \tag{3.51}$$

Next, observe that

$$\begin{aligned}
(a\to c) \wedge (a'\to c) \wedge (b'\to c) &= (a\to c) \wedge c \wedge (b'\to c) \\
&= (a\to c) \wedge (b\to c) \wedge (b'\to c) \\
&\le b\to c,
\end{aligned}$$

so

$$(b\to c)'\; C\; (a\to c) \wedge (a'\to c) \wedge (b'\to c). \tag{3.52}$$

Applying the Gudder-Schelp-Beran theorem, Eq. (3.30), to Eqs. (3.51) and (3.52), we conclude

$$\begin{aligned} a\to c\ C\ (b\to c)'\wedge(a'\to c)\wedge(b'\to c) & \\ a\to c\ C\ (b\to c)'\wedge(a'\to c) & \qquad \text{since } (b\to c)'\le b'\to c \\ (a\to c)'\ C\ (b\to c)'\wedge(a'\to c). & \end{aligned}$$

By hypothesis, $(b\to c)'\ C\ a'\to c$; applying GSB again, we obtain:

$$\begin{aligned} (b\to c)'\ C\ (a\to c)'\wedge(a'\to c) & \\ (b\to c)'\ C\ (a\to c)' & \qquad \text{since } (a\to c)'\le a'\to c \\ b\to c\ C\ a\to c & \\ a\to c\ C\ b\to c. & \end{aligned}$$

which is the conclusion of Eq. (3.49).

For Eq. (3.50): Replace a and b in Eq. (3.49) with their orthocomplements and apply Eq. (3.24) to the antecedents. ☐

Theorem 3.2.4. *Assume the following two conditions hold in an* OML*:*

$$a\to c\ C\ b'\to c \quad \& \quad b\to c\ C\ a'\to c. \tag{3.53}$$

Then any two terms from the set $\{a\to c,\ b\to c,\ a'\to c,\ b'\to c\}$ *commute.*

Proof. The possible cases are one of the following: one of the two hypothesis, a conclusion of Lemma 3.2.3, the cases obtained from these via Eq. (3.24), or (when a is in both terms or b is in both terms) the cases obtained using Eqs. (3.26) and (3.25). ☐

The following corollary shows a somewhat nonintuitive result where we obtain an equality from two inequalities which, from Eq. (3.38), we might at first think are much weaker than required.

Corollary 3.2.5. *The following condition holds in all* OML*s.*

$$(a'\to c)'\le b\to c \quad \& \quad (b'\to c)'\le a\to c \quad \Rightarrow \quad a\to c = b\to c. \tag{3.54}$$

Proof. From the hypotheses and Eq. (3.26), $a\to c\ C\ b'\to c$ and $b\to c\ C\ a'\to c$, so Theorem 3.2.4 implies that any two terms from the set $\{a\to c,\ b\to c,\ a'\to c,\ b'\to c\}$ commute.

Again from the hypotheses, we have

$$
\begin{aligned}
(a\to c) &= (a\to c)\vee(b'\to c)' \\
(a'\to c)'\vee(b\to c) &= (b\to c)
\end{aligned}
$$

Now, two equations of the form $x = y$ and $z = w$ imply $(x \equiv z')' = (y \equiv w')'$, where $(x \equiv z')' = (x' \vee z) \wedge (x \vee z')$. For the left-hand side we have, using the F-H distributive laws [Th. 3.1.3 (p. 30)] freely,

$$
\begin{aligned}
&((a\to c) \equiv ((a'\to c)'\vee(b\to c))')' \\
&\quad = ((a\to c)'\vee(a'\to c)'\vee(b\to c))\wedge((a\to c)\vee((a'\to c)\wedge(b\to c)')) \\
&\quad = ((a\to c)'\vee c'\vee(b\to c)) \\
&\qquad \wedge(((a\to c)\vee(a'\to c))\wedge((a\to c)\vee(b\to c)')) \\
&\quad = ((a\to c)'\vee 1)\wedge(1\wedge((a\to c)\vee(b\to c)')) \\
&\quad = (a\to c)\vee(b\to c)'.
\end{aligned}
$$

In the second step above, we used $(a\to c)'\vee(a'\to c)' = (a\to c)'\vee c'$ and in the third step, $c'\vee(b\to c) = 1$ and $(a\to c)\vee(a'\to c) = 1$. In general, we may use such two-variable equalities without showing their proofs, since they can be verified automatically, for example with the program `beran.c` or `lattice.c`.

For the right-hand side we have,

$$
\begin{aligned}
&(((a\to c)\vee(b'\to c)') \equiv (b\to c)')' \\
&\quad = (((a\to c)'\wedge(b'\to c))\vee(b\to c))\wedge((a\to c)\vee(b'\to c)'\vee(b\to c)') \\
&\quad = (((a\to c)'\vee(b\to c))\wedge((b'\to c)\vee(b\to c))) \\
&\qquad \wedge((a\to c)\vee c'\vee(b\to c)') \\
&\quad = (((a\to c)'\vee(b\to c))\wedge 1)\wedge(1\vee(b\to c)') \\
&\quad = (a\to c)'\vee(b\to c)).
\end{aligned}
$$

Equating the sides we have,

$$
\begin{aligned}
((a\to c)\vee(b\to c)') &= ((a\to c)'\vee(b\to c)) \\
((a\to c)\vee(b\to c)')\wedge(a\to c) &= ((a\to c)'\vee(b\to c))\wedge(a\to c) \\
a\to c &= ((a\to c)'\wedge(a\to c))\vee((b\to c)\wedge(a\to c))
\end{aligned}
$$

$$\begin{aligned} &= 0 \vee ((b \to c) \wedge (a \to c)) \\ &= (b \to c) \wedge (a \to c) \\ a \to c &\leq b \to c. \end{aligned}$$

Swapping the order of the hypotheses, the same argument gives us $b \to c \leq a \to c$ and hence the conclusion. ☐

The following lemma can assist us in finding commuting terms in expressions involving the Sasaki hook.

Lemma 3.2.6. *1. $(a \to c) \wedge (b \to c) \wedge c \; C \; t$ where t is any term built from $a \to c$, $a' \to c$, $b \to c$, $b' \to c$, and c.*

2. A conjunction of three or more terms from the set $a \to c$, $a' \to c$, $b \to c$, $b' \to c$, and c, that contains both of the variables a and b, is equal to $(a \to c) \wedge (b \to c) \wedge c$.

Proof. For part 1, we have the 5 relationships

$$\begin{aligned} (a \to c) \wedge (b \to c) \wedge c &\leq a \to c \\ (a \to c) \wedge (b \to c) \wedge c &= ((a \to c) \wedge c) \wedge (b \to c) \\ &= ((a \to c) \wedge (a' \to c)) \wedge (b \to c) \leq a' \to c \\ (a \to c) \wedge (b \to c) \wedge c &\leq b \to c \\ (a \to c) \wedge (b \to c) \wedge c &= (a \to c) \wedge ((b \to c) \wedge (b' \to c)) \leq b' \to c \\ (a \to c) \wedge (b \to c) \wedge c &\leq c \end{aligned}$$

Thus $(a \to c) \wedge (b \to c) \wedge c \; C \; a \to c$, etc. by Eq. (3.26). Using these relationships, we build up $(a \to c) \wedge (b \to c) \wedge c \; C \; t$ with Theorem 3.1.2.

For part 2, we exhaust all possible cases using the OML identities

$$(a \to c) \wedge c = (a \to c) \wedge (a' \to c) = (a' \to c) \wedge c.$$

☐

Chapter 4

GENERALIZED ORTHOARGUESIAN LATTICES

As we mentioned in the Introduction, before 1975 the orthomodular lattice (OML) equations were the only ones that were known to hold in a Hilbert lattice. These have been extensively studied in a vast body of research papers and books, particularly in the context of the logic of quantum mechanics, and so "orthomodular lattice" and "quantum logic" have become almost synonymous.

In 1975, Alan Day discovered an equation that holds in any Hilbert lattice but does not in all OMLs [31]. He derived the equation, called the **orthoarguesian law**, by imposing weakening orthogonality hypotheses on the so-called Arguesian law, an equation closely related to the famous law of projective geometry discovered by Desargues in the 1600's as part of an effort to help artists, stonecutters, and engineers.

In 2000, Megill and Pavičić discovered a new infinite class of equations that hold in any Hilbert lattice (and therefore in the $\mathscr{C}(H)$ of any finite- or infinite-dimensional Hilbert space) called **generalized orthoarguesian equations** or nOA laws, $n = 3, 4, \cdots < \infty$, a special case of which is the orthoarguesian law for $n = 4$.

4.1 HS proof of generalized orthoarguesian laws

In this section, we will show how the nOA laws Eq. (4.24) (p. 44) are derived from the elementary properties of a Hilbert space.

We will first derive a condition that holds in all Hilbert spaces (including finite-dimensional ones), from which the nOA laws for infinite-dimensional Hilbert spaces will follow. [We will also use this condition later to derive higher-order Arguesian laws for finite-dimensional Hilbert spaces; see Th. 7.4.1 (p. 140).]

Theorem 4.1.1. (Arguesian property of subspaces) *Let* $a_0,\ldots,a_n$ *and* $b_0,\ \ldots,b_n$*,* $n \ge 1$*, be any subspaces (not necessarily closed) of a Hilbert space, and let* $\cap$ *denote set-theoretical intersection and* $+$ *subspace sum. We define the subspace term* $t_n(i_0,\ldots,i_n)$ *recursively as follows, where* $0 \le i_0,\ldots,i_n \le n$*:*

$$t_1(i_0,i_1) = (a_{i_0} + a_{i_1}) \cap (b_{i_0} + b_{i_1}) \tag{4.1}$$

$$\begin{aligned} t_m(i_0,\ldots,i_m) = t_{m-1}(i_0,i_1,i_3,\ldots,i_m) \\ \cap (t_{m-1}(i_0,i_2,i_3,\ldots,i_m) + t_{m-1}(i_1,i_2,i_3,\ldots,i_m)), \\ 2 \le m \le n \end{aligned} \tag{4.2}$$

For $m = 2$*, this means* $t_2(i_0,i_1,i_2) = t_1(i_0,i_1) \cap (t_1(i_0,i_2) + t_1(i_1,i_2))$*.*[1] *Then the following condition holds in any finite- or infinite-dimensional Hilbert space for* $n \ge 1$*:*

$$\begin{aligned} (a_0 + b_0) \cap \cdots \cap (a_n + b_n) \\ \subseteq b_0 + (a_0 \cap (a_1 + t_n(0,\ldots,n))). \end{aligned} \tag{4.3}$$

Proof. (This theorem was originally proved in sketch form by Megill and Pavičić [76, p. 2368, Th. 5.2]; similar proofs have also been given by R. Mayet [68, p. 529, Th. 1] and us [101, p. 102103-11, Th. II.9]. The proof here includes some minor corrections to theorem statement and proof in the latter reference.) We will use $+$ to denote subspace sum (connecting two subspaces) and $+$ to denote vector sum (connecting two vectors). Let x be a vector belonging to the left-hand side of Eq. (4.3). Then $x \in a_i + b_i$ for $i = 0,\ldots,n$. From the definition of subspace sum, $x \in a_i + b_i$ implies there exist vectors x_i and y_i such that $x_i \in a_i$, $y_i \in b_i$, and $x = x_i + y_i$. From the last property, we have $x_i + y_i = x = x_j + y_j$ or

$$x_i - x_j = -y_i + y_j, \qquad 0 \le i,j \le n. \tag{4.4}$$

For the case $n = 1$ of Eq. (4.3), we need to prove

$$(a_0 + b_0) \cap (a_1 + b_1)$$

[1] Also, for for $m = 3$ we have $t_3(i_0,i_1,i_2,i_3) = t_2(i_0,i_1,i_3) \cap (t_2(i_0,i_2,i_3) + t_2(i_1,i_2,i_3)) = (t_1(i_0,i_1) \cap (t_1(i_0,i_3) + t_1(i_1,i_3))) \cap ((t_1(i_0,i_2) \cap (t_1(i_0,i_3) + t_1(i_2,i_3))) + (t_1(i_1,i_2) \cap (t_1(i_1,i_3) + t_1(i_2,i_3))))$; for $m = 4$ we have $t_4(i_0,i_1,i_2,i_3,i_4) = t_3(i_0,i_1,i_3,i_4) \cap (t_3(i_0,i_2,i_3,i_4) + t_3(i_1,i_2,i_3,i_4))$; and so on.

$$\subseteq b_0 + (a_0 \cap (a_1 + ((a_0 + a_1) \cap (b_0 + b_1)))) \tag{4.5}$$

Any linear combination of vectors from two subspaces belongs to their subspace sum. Since $y_0 \in b_0$ and $y_1 \in b_1$, we have $-y_0 + y_1 \in b_0 + b_1$. Therefore by Eq. (4.4), $x_0 - x_1 \in b_0 + b_1$. Also, $x_0 - x_1 \in a_0 + a_1$. Therefore

$$x_0 - x_1 \in (a_0 + a_1) \cap (b_0 + b_1). \tag{4.6}$$

Since $x_1 \in a_1$, we have $x_0 = x_1 + (x_0 - x_1) \in a_1 + ((a_0 + a_1) \cap (b_0 + b_1))$. Also, $x_0 \in a_0$, so $x_0 \in a_0 \cap (a_1 + ((a_0 + a_1) \cap (b_0 + b_1)))$. Finally, since $y_0 \in b_0$, we have $x = y_0 + x_0 \in b_0 + (a_0 \cap (a_1 + ((a_0 + a_1) \cap (b_0 + b_1))))$, proving that x belongs to the right-hand side of Eq. (4.5) and thus establishing the subset relation. This argument is illustrated by the following diagram:

$$\cdots \subseteq \underbrace{\underbrace{b_0}_{y_0} + (\underbrace{a_0}_{x_0} \cap \underbrace{(\underbrace{a_1}_{x_1} + \underbrace{(\underbrace{(a_0+a_1)}_{x_0-x_1} \cap \underbrace{(b_0+b_1)}_{-y_0+y_1=x_0-x_1})}_{x_0-x_1})}_{x_1+(x_0-x_1)=x_0})}_{y_0+x_0=x}.$$

For $n > 1$, notice that on the right-hand side of the above diagram, the term $(a_0 + a_1) \cap (b_0 + b_1) = t_1(0,1)$ from Eq. (4.5) gets replaced by the larger term $t_n(0,\ldots,n)$, with the rest of the right-hand side the same. Looking at the vector component $x_0 - x_1$ in this generalization of above diagram above, it is apparent that if we can prove

$$x_0 - x_1 \in t_n(0,\ldots,n), \tag{4.7}$$

then Eq. (4.3) is established. We will actually prove a more general result,

$$x_{i_0} - x_{i_1} \in t_m(i_0,\ldots,i_m), \qquad 0 \le i_0,\ldots,i_m \le n, 1 \le m \le n \tag{4.8}$$

from which Eq. (4.7) follows as a special case by setting $m = n$ and $i_0 = 0,\ldots,i_m = n$.

We will prove Eq. (4.8) by induction on m. For the basis step $m = 1$, the same argument that led to Eq. (4.6) above shows that

$$x_{i_0} - x_{i_1} \in t_1(i_0,i_1) = (a_{i_0} + a_{i_1}) \cap (b_{i_0} + b_{i_1}).$$

for $0 \leq i_0, i_1 \leq n$. For $m > 1$, assume we have proved $x_{i_0} - x_{i_1} \in t_{m-1}(i_0, i_1, \ldots, i_{m-1})$ for all $0 \leq i_0, \ldots, i_{m-1} \leq n$. Then, in particular, we have the substitution instances

$$x_{i_0} - x_{i_1} \in t_{m-1}(i_0, i_1, i_3, \ldots, i_m) \tag{4.9}$$
$$x_{i_0} - x_{i_2} \in t_{m-1}(i_0, i_2, i_3, \ldots, i_m) \tag{4.10}$$
$$x_{i_1} - x_{i_2} \in t_{m-1}(i_1, i_2, i_3, \ldots, i_m). \tag{4.11}$$

Combining Eqs. (4.10) and (4.11),

$$\begin{aligned} x_{i_0} - x_{i_1} &= (x_{i_0} - x_{i_2}) - (x_{i_1} - x_{i_2}) \\ &\in t_{m-1}(i_0, i_2, i_3, \ldots, i_m) + t_{m-1}(i_1, i_2, i_3, \ldots, i_m). \end{aligned}$$

Combining this with Eq. (4.9) and using Eq. (4.2),

$$\begin{aligned} x_{i_0} - x_{i_1} &\in t_{m-1}(i_0, i_1, i_3, \ldots, i_m) \\ &\qquad \cap (t_{m-1}(i_0, i_2, i_3, \ldots, i_n) + t_{m-1}(i_1, i_2, i_3, \ldots, i_m)) \\ &= t_m(i_0, \ldots, i_m) \end{aligned}$$

as required. ☐

We will use the above theorem to derive a condition that holds in the lattice of closed subspaces of a Hilbert space. We recall the following definitions. Two vectors are orthogonal when their inner product is zero, and the orthocomplement of a subspace a, denoted $a^\perp$, is the set of all vectors orthogonal to all vectors in a. We will use $a \perp b$ to denote $a \subseteq b^\perp$, meaning that subspaces a and b are orthogonal. The join of two subspaces $a \vee b$ is defined as $(a+b)^{\perp\perp}$, their meet $a \cap b$ is defined as set intersection $a \cap b$, and their ordering $a \leq b$ is defined as $a \subseteq b$. The following lemma states two well-known facts we will use; see, for example, Ref. [7] or [35, p. 28].

Lemma 4.1.2. *Let a and b be two closed subspaces of a Hilbert space. Then*

$$a + b \subseteq a \vee b \tag{4.12}$$
$$a \perp b \quad \Rightarrow \quad a + b = a \vee b \tag{4.13}$$

We can actually prove a stronger version of Eq. (4.13). Since it apparently does not occur in the literature, we give a detailed proof.

Lemma 4.1.3. *Let a and b be two closed subspaces of a Hilbert space H. Then*

$$aCb \quad \Rightarrow \quad a+b = a\vee b \tag{4.14}$$

where aCb denotes "a commutes with b" (Def. 2.2.8).

Proof. We will use aCb in the form $a\wedge(a'\vee b)\le b$ and make use of the fact that $\mathscr{C}(H)$ is an OML. We will also use the following property that can be shown to hold in $\mathscr{C}(H)$ by direct appeal to the definition of subspace sum (recall that $\le$ is the same as $\subseteq$):

$$r\le s \quad \Rightarrow \quad r+t\le s+t \tag{4.15}$$

where r, s, t are any (not necessarily closed) subspaces. In any OML, it is easily verified that $(a\wedge(a'\vee b'))\vee b = a\vee b$. Equating both sides to $(a\wedge(a'\vee b'))+b$,

$$(a\wedge(a'\vee b'))+b=(a\wedge(a'\vee b'))\vee b \;\Leftrightarrow\; (a\wedge(a'\vee b'))+b=a\vee b. \tag{4.16}$$

As a special case of Eq. (4.15), $a\wedge(a'\vee b')\le a\Rightarrow(a\wedge(a'\vee b'))+b\le a+b$. The antecedent is true in any OL, so by modus ponens $(a\wedge(a'\vee b'))+b\le a+b$. Applying an equality law gives $(a\wedge(a'\vee b'))+b=a\vee b\Rightarrow a\vee b\le a+b$. Chaining this and Eq. (4.16),

$$(a\wedge(a'\vee b'))+b=(a\wedge(a'\vee b'))\vee b\Rightarrow a\vee b\le a+b. \tag{4.17}$$

Assuming the hypothesis aCb, we have aCb' by Eq. (3.25). By the definition of commutes, $a\wedge(a'\vee b')\le b'$ i.e. $a\wedge(a'\vee b')\perp b$. Using Eq. (4.13), this gives $(a\wedge(a'\vee b'))+b=(a\wedge(a'\vee b'))\vee b$. By modus ponens and Eq. (4.17), we obtain $a\vee b\le a+b$. The other direction $a+b\le a\vee b$ holds by Eq. (4.12). Combining the two directions, we conclude $a+b=a\vee b$. $\square$

Note that Eq. (4.13) now becomes a special case of Eq. (4.14), since in any OML, $a\le b$ implies aCb by Eq. (3.26).

We are now ready to state our main theorem.

Theorem 4.1.4. (Generalized Orthoarguesian Laws) *Let $a_0,\ldots,a_n$ and $b_0,\ldots,b_n$, $n\ge 1$, be closed subspaces of a Hilbert space. We define the term $t_n^{\vee}(i_0,\ldots,i_n)$ by substituting $\vee$ for $+$ in the term $t_n(i_0,\ldots,i_n)$ from Theorem 4.1.1. Then following condition holds in any finite- or* *infinite-dimensional Hilbert space for* $n \geq 1$*:*

$$\begin{aligned} a_0 \perp b_0 \;\&\; \cdots \;\&\; a_n \perp b_n \quad &\Rightarrow \\ (a_0 \vee b_0) \cap \cdots \cap (a_n \vee b_n) & \\ \leq b_0 \vee (a_0 \cap (a_1 \vee t_n^{\vee}(0,\ldots,n))). & \end{aligned} \tag{4.18}$$

Proof. By the orthogonality hypotheses and Eq. (4.13), the left-hand side of Eq. (4.18) equals the left-hand side of Eq. (4.3). By Eq. (4.12), the right-hand side of Eq. (4.3) is a subset of the right-hand side of Eq. (4.18). Eq. (4.18) follows by Theorem 4.1.1 and the transitivity of the subset relation. □

We can also put the above theorem in a more general form.

Theorem 4.1.5. *Th. 4.1.4 also holds when the hypotheses*

$$a_0 \perp b_0 \;\&\; \cdots \;\&\; a_n \perp b_n$$

are replaced with the weaker hypotheses

$$a_0 C b_0 \;\&\; \cdots \;\&\; a_n C b_n$$

where C is the commutes relation.

Proof. The proof is the same as the one for Th. 4.1.4, except that we use Eq. (4.14) in place of Eq. (4.13). □

Th. 4.1.4 now becomes a special case of Th. 4.1.5, since in any OML, $a \leq b$ implies aCb by Eq. (3.26). We mention that Th. 4.1.5 and Th. 4.1.4 can actually be shown to be equivalent to each other in an OML, so in that sense Th. 4.1.5 does not provide any new information. However, Th. 4.1.5 may be more convenient in some cases because of its weaker hypotheses.

Theorem 4.1.6. *An* OL *in which Eq. (4.18) holds is an* OML.

Proof. [84, Th. 2.16] It suffices to show this for the lowest-order equation, which follows from the higher order ones. For $n = 1$, we can express Eq. (4.18) as

$$x \perp y \;\&\; z \perp w \;\Rightarrow\; (x \cup y) \cap (z \cup w) \leq y \cup (x \cap (z \cup ((x \cup z) \cap (y \cup w)))). \tag{4.19}$$

Putting $b, 0, a, a'$ for x, y, z, w respectively, the hypotheses are satisfied and the conclusion becomes $(b \cup 0) \cap (a \cup a') \leq 0 \cup (b \cap (a \cup ((b \cup a) \cap (0 \cup a'))))$. Simplifying, we get $b \leq b \cap (a \cup$

$(a' \cap (a \cup b)$. Dropping the conjunct b from the right-hand side, adding the disjunct a to the left-hand side, and noticing that the other direction of the resulting inequality holds in any OL, we arrive at $a \cup b = a \cup (a' \cap (a \cup b))$, which is the orthomodular law (Def. 2.2.7). ☐

Note that the orthomodular law also follows (in any OL) from the nOA laws in the form of Eq. (4.24) below (p. 44). However, those equations make use of the orthomodular law for their derivation from Eq. (4.18). The above theorem gives us an alternate way to derive the orthomodular law directly from Hilbert space that is, in some ways, more elementary than the traditional proof by contradiction (e.g. Ref. [49, p. 65]).

4.2 Definitions

Ref. [76] shows that in any OML (which includes the lattice of closed subspaces of a Hilbert space), Eq. (4.18) is equivalent to the mOA law (that we will introduce below) Eq. (4.24) for $m = n+2$, thus establishing Theorem 4.2.3 below. First, we will introduce the definitions needed to state those laws.

For the following definition, we recall that $a \to b \stackrel{\text{def}}{=} a' \vee (a \wedge b)$ [Def. 2.2.5 (p. 18)].

Definition 4.2.1. *We define an operation* $\stackrel{(n)}{\equiv}$ *on n variables* $a_1, \ldots, a_n$ *(*$n \geq 3$*) as follows:*[2]

$$a_1 \stackrel{(3)}{\equiv} a_2 \stackrel{\text{def}}{=} ((a_1 \to a_3) \wedge (a_2 \to a_3)) \vee ((a_1' \to a_3) \wedge (a_2' \to a_3)) \tag{4.20}$$

$$a_1 \stackrel{(n)}{\equiv} a_2 \stackrel{\text{def}}{=} (a_1 \stackrel{(n-1)}{\equiv} a_2) \vee ((a_1 \stackrel{(n-1)}{\equiv} a_n) \wedge (a_2 \stackrel{(n-1)}{\equiv} a_n)), \quad n \geq 4. \tag{4.21}$$

For the cases $n = 4$ and 5, the above definition reads:

$$a_1 \stackrel{(4)}{\equiv} a_2 \stackrel{\text{def}}{=} (a_1 \stackrel{(3)}{\equiv} a_2) \vee ((a_1 \stackrel{(3)}{\equiv} a_4) \wedge (a_2 \stackrel{(3)}{\equiv} a_4)) \tag{4.22}$$

$$a_1 \stackrel{(5)}{\equiv} a_2 \stackrel{\text{def}}{=} (a_1 \stackrel{(4)}{\equiv} a_2) \vee ((a_1 \stackrel{(4)}{\equiv} a_5) \wedge (a_2 \stackrel{(4)}{\equiv} a_5)) \tag{4.23}$$

[2] To obtain $\stackrel{(n)}{\equiv}$ we substitute in each $\stackrel{(n-1)}{\equiv}$ subexpression only the two explicit variables, leaving the other variables the same. For example, $(a_2 \stackrel{(4)}{\equiv} a_5)$ on the right side of (4.21) for $n = 5$ means $(a_2 \stackrel{(3)}{\equiv} a_5) \vee ((a_2 \stackrel{(3)}{\equiv} a_4) \wedge (a_5 \stackrel{(3)}{\equiv} a_4))$ which means $(((a_2 \to a_3) \wedge (a_5 \to a_3)) \vee ((a_2' \to a_3) \wedge (a_5' \to a_3))) \vee ((((a_2 \to a_3) \wedge (a_4 \to a_3)) \vee ((a_2' \to a_3) \wedge (a_4' \to a_3))) \wedge (((a_5 \to a_3) \wedge (a_4 \to a_3)) \vee ((a_5' \to a_3) \wedge (a_4' \to a_3))))$. The explicit expansion can also be obtained from the output of the program `oagen.c` described in Sec. A.8 (p. 156).

Definition 4.2.2. *For each $n \geq 3$, the equation*

$$(a_1 \to a_3) \wedge (a_1 \overset{(n)}{\equiv} a_2) \leq a_2 \to a_3\,. \tag{4.24}$$

is called OA-*n. The equational variety consisting of the* OML*s in which* OA-*n holds is called* nOA*, and thus we also call equation* OA-*n the* nOA **law**.

The important property of these equations is the following:

Theorem 4.2.3. [76] *The* nOA **laws** *($n \geq 3$) hold in all* HL*s.*

The notation $a_1 \overset{(n)}{\equiv} a_2$ is useful when we do not need to specify assignments to implicit variables $a_3, \ldots, a_n$. When constructing the expression $a_1 \overset{(n)}{\equiv} a_2$ using Def. 4.2.1 above, the variables with the names $a_3, \ldots, a_n$ must be assigned strictly according to the footnote for that definition. In particular, if an expression contains two or more occurrences of the operation $\overset{(n)}{\equiv}$ (for example, $a \overset{(n)}{\equiv} b$ and $c \overset{(n)}{\equiv} d$), the implicit variables are assumed to be the same in each one unless otherwise specified.

When $\overset{(n)}{\equiv}$ occurs more than once in a condition, we sometimes need new variable names that are different from the implicit ones $a_3, \ldots, a_n$, and this notation becomes inadequate. The most frequent case is when we need to assign a different variable to a_3, and for that purpose we introduce the following alternate notation.

Definition 4.2.4.

$$a_1 \overset{a_3}{\equiv}_n a_2 \overset{\text{def}}{=} a_1 \overset{(n)}{\equiv} a_2 \tag{4.25}$$

Again, the implicit variables $a_4, \ldots, a_n$ are assumed to be the same in each occurrence of expressions of the form $a \overset{c}{\equiv}_n b$ unless we specify otherwise.

Finally, and in particular for the frequent special cases $n = 3$ and $n = 4$, it is convenient to have a notation that specifies all variables explicitly.

Definition 4.2.5. *We define* $a_1\stackrel{a_3}{\equiv}a_2 \stackrel{\text{def}}{=} a_1\stackrel{(3)}{\equiv}a_2$ *and* $a_1\stackrel{a_4,a_3}{\equiv}a_2 \stackrel{\text{def}}{=} a_1\stackrel{(4)}{\equiv}a_2$. *Explicitly, we have*

$$a\stackrel{c}{\equiv}b = ((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c)) \tag{4.26}$$

$$\begin{aligned} a\stackrel{c,d}{\equiv}b &= (a\stackrel{d}{\equiv}b)\vee((a\stackrel{d}{\equiv}c)\wedge(b\stackrel{d}{\equiv}c)) &(4.27)\\ &= ((a\to d)\wedge(b\to d))\vee((a'\to d)\wedge(b'\to d))\\ &\quad\vee((((a\to d)\wedge(c\to d))\vee((a'\to d)\wedge(c'\to d)))\\ &\qquad\wedge(((b\to d)\wedge(c\to d))\vee((b'\to d)\wedge(c'\to d)))) &(4.28)\end{aligned}$$

Thus, a **3OA** *is an* OML *in which the following additional condition is satisfied:*

$$(a\to c)\wedge(a\stackrel{c}{\equiv}b)\le b\to c. \tag{4.29}$$

A **4OA** *is an* OML *in which the following additional condition is satisfied:*

$$(a\to d)\wedge(a\stackrel{c,d}{\equiv}b)\le b\to d. \tag{4.30}$$

In general, we define $a_1\stackrel{a_n,\ldots,a_3}{\equiv}a_2 \stackrel{\text{def}}{=} a_1\stackrel{(n)}{\equiv}a_2$, *where* $a_n,\ldots,a_3$ *may be an explicit list of variables (no* $\ldots$*) if necessary.*

Thus we have three notations for the nOA operation that are increasingly explicit, depending on the needs of their application. To summarize these, we have

$$\begin{aligned} a_1\stackrel{(3)}{\equiv}a_2 &= a_1\stackrel{a_3}{\equiv}_3a_2 = a_1\stackrel{a_3}{\equiv}a_2 &(4.31)\\ a_1\stackrel{(4)}{\equiv}a_2 &= a_1\stackrel{a_3}{\equiv}_4a_2 = a_1\stackrel{a_4,a_3}{\equiv}a_2 &(4.32)\\ a_1\stackrel{(5)}{\equiv}a_2 &= a_1\stackrel{a_3}{\equiv}_5a_2 = a_1\stackrel{a_5,a_4,a_3}{\equiv}a_2 &(4.33)\\ &\vdots\\ a_1\stackrel{(n)}{\equiv}a_2 &= a_1\stackrel{a_3}{\equiv}_na_2 = a_1\stackrel{a_n,\ldots,a_3}{\equiv}a_2. &(4.34)\end{aligned}$$

The following lemma shows some general properties of the nOA operation that hold in all OMLs and will be of use to us later.

Lemma 4.2.6. *The following conditions, for* $n\ge 3$, *hold in all* OMLs. *Note that whenever the operation* $\stackrel{(n)}{\equiv}$ *or* $\stackrel{a_3}{\equiv}_n$ *appears more than once in a condition, the implicit variables are assumed* *to be the same. The expression* $a_n,\ldots,a_4,a_3$ *means* a_3 *for* $n=3$.

$$a_1 \stackrel{(n)}{\equiv} a_2 = a_2 \stackrel{(n)}{\equiv} a_1 \tag{4.35}$$

$$a_1 \stackrel{(n)}{\equiv} a_2 = a_1' \stackrel{(n)}{\equiv} a_2' \tag{4.36}$$

$$a_1 \stackrel{(n)}{\equiv} a_1 = 1 \tag{4.37}$$

$$a \equiv b \le a_1 \stackrel{(n)}{\equiv} a_2 \tag{4.38}$$

$$a_1 \stackrel{(n)}{\equiv} a_2 \le a_1 \stackrel{(n+1)}{\equiv} a_2 \tag{4.39}$$

$$a \stackrel{c}{\equiv} b = ((a \to c) \equiv (b \to c)) \vee ((a' \to c) \equiv (b' \to c)) \tag{4.40}$$

$$a \to c = b \to c \quad \Rightarrow \quad a \stackrel{c}{\equiv}_n b = 1 \tag{4.41}$$

$$(a \to c) \stackrel{c}{\equiv}_n b = a' \stackrel{c}{\equiv}_n b \tag{4.42}$$

$$(a \to c) \stackrel{c}{\equiv}_n (b \to c) = a \stackrel{c}{\equiv}_n b \tag{4.43}$$

$$(a_1 \to a_3) \stackrel{a_n \to a_3,\ldots,a_4 \to a_3,a_3}{\equiv} (a_2 \to a_3) = a_1 \stackrel{a_n,\ldots,a_4,a_3}{\equiv} a_2. \tag{4.44}$$

Proof. The proofs for most of these are obvious from Defs. 4.2.1 and 2.2.6.

For Eq. (4.40), $((a \to c) \equiv (b \to c)) \vee ((a' \to c) \equiv (b' \to c)) = ((a \to c) \wedge (b \to c)) \vee ((a \to c)' \wedge (b \to c)') \vee ((a' \to c) \wedge (b' \to c)) \vee ((a' \to c)' \wedge (b' \to c)')$, and the second and fourth disjuncts are absorbed by the third and first respectively.

For Eq. (4.41): For $n=3$, by Eq. (3.1), $a \to c = b \to c$ implies $1 = (a \to c) \equiv (b \to c) = ((a \to c) \wedge (b \to c)) \vee ((a \to c)' \wedge (b \to c)')$. Since $(a \to c)' \le a' \to c$ and $(b \to c)' \le b' \to c$ by Eq. (3.38), $1 \le ((a \to c) \wedge (b \to c)) \vee ((a' \to c) \wedge (b' \to c)) = a \stackrel{c}{\equiv} b$. The $n > 3$ case follows from the $n=3$ case by Eq. (4.39).

For Eqs. (4.42), (4.43), (4.44): Use $(a \to b) \to b = a' \to b$, $(a' \to b) \to b = a \to b$, and induction on n. ☐

To make certain equations slightly shorter when fully expanded (which can be faster to run with computer programs such as `lattice.c`), we also define the following modified version of the nOA operation. The remark in the footnote to Def. 4.2.1 concerning implicit variable names also applies to this definition.

Definition 4.2.7.

$$a_1 \overset{(3)}{\eqcirc} a_2 \overset{\text{def}}{=} (a_1 \wedge a_2) \vee ((a_1 \to a_3) \wedge (a_2 \to a_3)) \tag{4.45}$$

$$a_1 \overset{(4)}{\eqcirc} a_2 \overset{\text{def}}{=} (a_1 \overset{(3)}{\eqcirc} a_2) \vee ((a_1 \overset{(3)}{\eqcirc} a_4) \wedge (a_2 \overset{(3)}{\eqcirc} a_4)) \tag{4.46}$$

$$a_1 \overset{(5)}{\eqcirc} a_2 \overset{\text{def}}{=} (a_1 \overset{(4)}{\eqcirc} a_2) \vee ((a_1 \overset{(4)}{\eqcirc} a_5) \wedge (a_2 \overset{(4)}{\eqcirc} a_5)) \tag{4.47}$$

$$a_1 \overset{(n)}{\eqcirc} a_2 \overset{\text{def}}{=} (a_1 \overset{(n-1)}{\eqcirc} a_2) \vee ((a_1 \overset{(n-1)}{\eqcirc} a_n) \wedge (a_2 \overset{(n-1)}{\eqcirc} a_n)), \quad n \geq 4\,. \tag{4.48}$$

We also define analogous notations for making variable names explicit:

Definition 4.2.8.

$$a_1 \overset{a_3}{\eqcirc} a_2 \overset{\text{def}}{=} a_1 \overset{a_3}{\eqcirc}_3 a_2 \overset{\text{def}}{=} a_1 \overset{(3)}{\eqcirc} a_2 \tag{4.49}$$

$$a_1 \overset{a_4,a_3}{\eqcirc} a_2 \overset{\text{def}}{=} a_1 \overset{a_3}{\eqcirc}_4 a_2 \overset{\text{def}}{=} a_1 \overset{(4)}{\eqcirc} a_2 \tag{4.50}$$

$$a_1 \overset{a_5,a_4,a_3}{\eqcirc} a_2 \overset{\text{def}}{=} a_1 \overset{a_3}{\eqcirc}_5 a_2 \overset{\text{def}}{=} a_1 \overset{(5)}{\eqcirc} a_2 \tag{4.51}$$

$$\vdots$$

$$a_1 \overset{a_n,\ldots,a_3}{\eqcirc} a_2 \overset{\text{def}}{=} a_1 \overset{a_3}{\eqcirc}_n a_2 \overset{\text{def}}{=} a_1 \overset{(n)}{\eqcirc} a_2. \tag{4.52}$$

For the frequent cases of $n = 3, 4$, we have explicitly

$$a \overset{c}{\eqcirc} b = (a \wedge b) \vee ((a \to c) \wedge (b \to c)) \tag{4.53}$$

$$a \overset{c,d}{\eqcirc} b = (a \overset{d}{\eqcirc} b) \vee ((a \overset{d}{\eqcirc} c) \wedge (b \overset{d}{\eqcirc} c)) \tag{4.54}$$

$$\begin{aligned} &= (a \wedge b) \vee ((a \to d) \wedge (b \to d)) \\ &\quad \vee (((a \wedge c) \vee ((a \to d) \wedge (c \to d))) \\ &\qquad \wedge ((b \wedge c) \vee ((b \to d) \wedge (c \to d)))) \end{aligned} \tag{4.55}$$

The modified nOA operation does not satisfy all of the properties of the standard nOA operation listed in Lemma 4.2.6. Some of its properties are as follows.

Lemma 4.2.9. *The following conditions hold in all* OML*s. Note that whenever the operations* $\overset{(n)}{\equiv}$, $\overset{a_3}{\equiv}_n$, $\overset{(n)}{\eqcirc}$, *and* $\overset{a_3}{\eqcirc}_n$ *appear more than once in a condition, the implicit variables are assumed to* *be the same. The expression* $a_n, \ldots, a_4, a_3$ *means* a_3 *for* $n = 3$.

$$a_1 \stackrel{(n)}{=\!\!\!\!\circ} a_2 = a_2 \stackrel{(n)}{=\!\!\!\!\circ} a_1 \tag{4.56}$$

$$a_1 \equiv a_2 \le a_1 \stackrel{(n)}{=\!\!\!\!\circ} a_2 \le a_1 \stackrel{(n)}{\equiv} a_2 \tag{4.57}$$

$$a_1 \to a_3 \stackrel{a_n\to c,\ldots,a_4\to c,a_3}{=\!\!\!\!\circ} a_2 \to a_3 = a_1 \stackrel{a_n,\ldots,a_4,a_3}{\equiv} a_2 \tag{4.58}$$

$$(a \to c) \stackrel{c}{=\!\!\!\!\circ} (b \to c) = a \stackrel{c}{\equiv} b \tag{4.59}$$

Proof. The proofs follow directly from Defs. 4.2.7 and 4.2.8. In particular, we use the relationships $a \le a' \to b$, $a' \le a \to b$, $(a \to b) \to b = a' \to b$, and $(a' \to b) \to b = a \to b$, applying induction on n as needed. ☐

4.3 Independence results

It is conjectured that the equational variety $(n+1)$OA is strictly smaller than nOA for all n. In this section, we review what is known about this conjecture.

Corollary 4.3.1. *In any* OML*, Day's orthoarguesian law* [31] *is equivalent to the* 4OA *law and the equations found by Godowski and Greechie in 1984* [27] *are equivalent to each other and to* 3OA.

Proof. As given in Ref. [76]. ☐

Theorem 4.3.2. *Any ortholattice* (OL) [107, Def. 1] *to which an* nOA *law is added is orthomodular. No* nOA *law holds in all* OML*s.*

Proof. All nOA laws fail in ortholattice O6 (**benzene ring, hexagon**) [107, Sec. 2].

We prove the second statement of the theorem by finding an orthomodular lattice in which the 3OA law fails. One such OML is shown in Figs. 4.1(a) below. Since the $(n+1)$OA law implies the nOA law (see Theorem 4.3.3 below), the result follows. ☐

We conjecture that the second statement of the following theorem holds for any n. To prove it for $n \ge 7$ is an open problem.

Theorem 4.3.3. *In an* OL*, the* nOA *law implies the* $(n-1$OA *law for any* $n > 3$. *In an* OL*, the* nOA *law does not imply the* $(n+1)$OA *law for* $3 \le n \le 6$.

Proof. The first statement easily follows from the definition of the *n*OA laws.

The proof for each *n* of the second statement consists of exhibiting an OML that satisfies nOA and violates $(n+1)$OA. For $n = 3, 4$, see Ref. [76]. For $n = 5$, see Ref. [105, p. 766, Th. 11]. For $n = 6$, see Ref. [84]. We also show these counterexamples in Figs. 4.1 and 4.2 below. ☐

These counterexamples were found by the following method. We started with the program `nauty` written by Brendan McKay [73], which exhaustively generates finite OML lattices. These in turn were fed into the program `latticeg.c` (or its faster variant, `lattice2g.c`), which tests the *n*OA laws against those lattices [see Sec. A.1 below (p. 146) and also Ref. [84]]. The *n*OA laws are very long equations whose lengths grow exponentially with *n* (with $4 \cdot 3^{n-2} + 3$ variable occurrences when expanded to elementary operations). As *n* increases, the difficulty of finding these counterexamples increases exponentially. Finding the counterexamples for 4OA vs. 5OA and 5OA vs. 6OA required over 10 years of CPU time on the Cluster Isabella (224 CPUs) and Civil Engineering Cluster (60 CPUs) of the University of Zagreb. Some additional lattices in which 5OA holds and 6OA can be found in Ref. [105, p. 767]. The search that resulted in the 6OA vs. 7OA counterexample is described in Ref. [84]. To pursue the search for higher *n*'s is currently too costly with the available algorithms and computers.

In Figs. 4.1 and 4.2, we show the Greechie diagrams of the counterexample lattices used for the above proof.[3] For Figs. 4.2(a)[4] and (b),[5] we drew[6] them with outer loops of orders 9 and 10, continuing a possible pattern in the outer loops of orders 6, 7, and 8 of Figs. 4.1(a),[7] (b),[8] and (c).[9] In Refs. [105, p. 767, Fig. 6] and 33-21-oa6p7f [84], the lattices of Figs. 4.2(a) and (b) are shown using maximal outer loops of orders 11 and 14. As we show below in Fig. 5.3 (p. 81), a Greechie diagram drawn with a maximal outer loop may disguise a pattern to be sought. While our redrawn diagrams in Figs. 4.2(a) and (b) also do not reveal any apparent pattern, they show an example of the different approaches that may be needed to reveal a pattern, if there is one.

[3] The notation "17-10-oa3p4f" means "17 atoms, 10 edges, in which the 3OA law passes and the 4OA law fails."

[4] `HIO,FHM,FGN,EGJ,CIL,ADQ,9BP,8IK,7BF,678,5CD,34A,26E,23H,159,14G,JRS,IPS.` is an MMP encoding for Fig. 4.2(a).

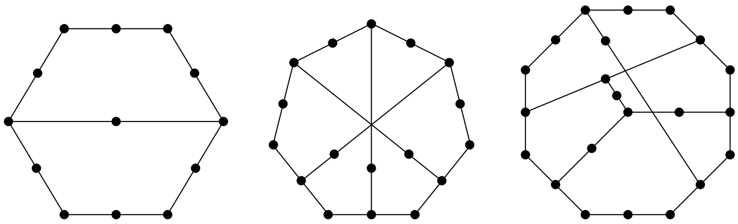
[5] `123,345,567,789,9AB,BCD,DEF,FGH,HIJ,JKL,LMN,NOP,PQR,RS1,4EK,4AP,AVH,BXL, DUQ,FWN,JTQ.` is an MMP encoding for Fig. 4.2(b).

[6] Assisted by the program `loop.c` [Sec. A.7 (p. 154)]. For Fig. 4.2(a), from the possibilities with an outer loop of 9, we chose the unique one that had no completely internal edges i.e. in which every internal edge connects to the outer loop. Figs. 4.1(c) and 4.2(c) do have such completely internal edges.

[7] `123,345,567,789,9AB,BC1,BD5.` is an MMP encoding for Fig. 4.1(a).

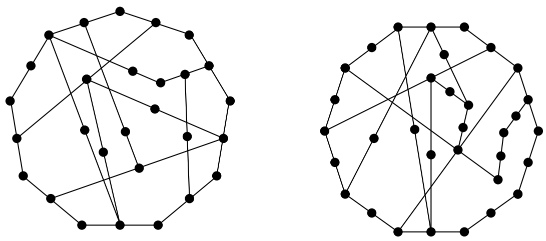
[8] `123,345,567,789,9AB,BCD,DE1,3FA,1G8,6HD.` is an MMP encoding for Fig. 4.1(b).

[9] `123,345,567,789,9AB,BCD,DEF,FG1,1I8,4HE,6LK,CJK,HMK.` is an MMP encoding for Fig. 4.1(c).



(a) (b) (c)

Figure 4.1: Lattices (a) 13-7-OMLp3f, which is an OML but not a 3OA, (b) 17-10-oa3p4f, which is a 3OA but not a 4OA, (c) 22-13-oa4p5f-a, which is a 4OA but not a 5OA [105, p. 766, Fig. 5].



(a) (b)

Figure 4.2: Lattices (a) 28-18-oa5p6f-b [105, p. 767, Fig. 6], which is a 5OA but not a 6OA, and (b) 33-21-oa6p7f [84], which is a 6OA but not a 7OA.

4.4 Equivalents for the 3OA law

We will focus on 3OA in this section. In many cases the results also hold for nOA with straightforward generalizations. In particular, the term "$a\stackrel{c}{\equiv}b$" can often be replaced with "$a\stackrel{c}{\equiv}_n b$" without further modification.

For easier reference, we collect below the 3OA equivalents proved in this section.

$$(a\to c)\wedge(a\stackrel{c}{\equiv}b) = (b\to c)\wedge(a\stackrel{c}{\equiv}b) \qquad \text{[see Eq. (4.60), p. 51]}$$

$$(a\to c)\wedge(a\stackrel{c}{\equiv}b) = (a\to c)\wedge(b\to c) \qquad \text{[see Eq. (4.61), p. 51]}$$

$$(a\to c)\wedge(a\stackrel{c}{=}b) \le b\to c \qquad \text{[see Eq. (4.62), p. 51]}$$

$$(a\to c)\wedge(a\stackrel{c}{=}b) = (a\to c)\wedge(b\to c) \qquad \text{[see Eq. (4.63), p. 51]}$$

$$(a\to c)\wedge(a\stackrel{c}{=}b) = (b\to c)\wedge(a\stackrel{c}{=}b) \qquad \text{[see Eq. (4.64), p. 51]}$$

$$(a\to c)\wedge(a\stackrel{c}{\equiv}b) = (b\to c)\wedge(a\stackrel{c}{=}b) \qquad \text{[see Eq. (4.65), p. 51]}$$

$$(a\to c)\wedge(a\stackrel{c}{\equiv}b) \le a\stackrel{c}{=}b \qquad \text{[see Eq. (4.66), p. 51]}$$

$$a\wedge(a\stackrel{c}{=}b) \le b'\to c \qquad \text{[see Eq. (4.67), p. 52]}$$

$$b \wedge (a \overset{c}{=} b) \le a' \to c \qquad \text{[see Eq. (4.68), p. 52]}$$

$$a \wedge (a \overset{c}{\equiv} b) \le b' \to c \qquad \text{[see Eq. (4.69), p. 52]}$$

$$a' \wedge (a \overset{c}{\equiv} b) \le b \to c \qquad \text{[see Eq. (4.70), p. 52]}$$

$$(b' \to c) \wedge ((b \to c) \vee ((a \to c) \wedge (a \overset{c}{\equiv} b))) \le c \qquad \text{[see Eq. (4.72), p. 53]}$$

$$(a \to c) \wedge (a \vee (b \wedge (a \overset{c}{\equiv} b))) \le c \qquad \text{[see Eq. (4.73), p. 54]}$$

$$a' \wedge (a \vee (b \wedge (a \overset{c}{\equiv} b) \le c \qquad \text{[see Eq. (4.74), p. 55]}$$

$$a' \wedge (a \vee (b \wedge (a \overset{c}{=} b) \le c \qquad \text{[see Eq. (4.75), p. 55]}$$

$$\begin{aligned} & a \perp b \quad \& \quad c \perp d \Rightarrow \\ & \quad (a \vee b) \wedge (c \vee d) \le b \vee (a \wedge (c \\ & \qquad\qquad \vee ((a \vee c) \wedge (b \vee d)))) \qquad \text{[see Eq. (4.76), p. 56]} \end{aligned}$$

$$\begin{aligned} & d \wedge (e \vee (d \wedge f)) = (d \wedge e) \vee (d \wedge f) \\ & \qquad \text{where } d = a \to c, \\ & \qquad e = (a' \to c) \wedge (b' \to c), \\ & \qquad \text{and } f = (a \to c) \wedge (b \to c) \qquad \text{[see Eq. (4.77), p. 56]} \end{aligned}$$

$$(a \to c) \wedge (a \overset{c}{\equiv} b)\ C\ b \to c \qquad \text{[see Eq. (4.78), p. 57]}$$

$$((a \to c) \wedge (a \overset{c}{\equiv} b)) \to c = ((b \to c) \wedge (a \overset{c}{\equiv} b)) \to c \qquad \text{[see Eq. (4.82), p. 59]}$$

Theorem 4.4.1. *An* OML *in which any of the following equations*

$$(a \to c) \wedge (a \overset{c}{\equiv} b) = (b \to c) \wedge (a \overset{c}{\equiv} b) \tag{4.60}$$

$$(a \to c) \wedge (a \overset{c}{\equiv} b) = (a \to c) \wedge (b \to c) \tag{4.61}$$

$$(a \to c) \wedge (a \overset{c}{=} b) \le b \to c \tag{4.62}$$

$$(a \to c) \wedge (a \overset{c}{=} b) = (a \to c) \wedge (b \to c) \tag{4.63}$$

$$(a \to c) \wedge (a \overset{c}{=} b) = (b \to c) \wedge (a \overset{c}{=} b) \tag{4.64}$$

$$(a \to c) \wedge (a \overset{c}{\equiv} b) = (b \to c) \wedge (a \overset{c}{=} b) \tag{4.65}$$

$$(a \to c) \wedge (a \overset{c}{\equiv} b) \le a \overset{c}{=} b \tag{4.66}$$

holds is a 3OA *and vice versa.*

Proof. To obtain Eq. (4.60), apply Eq. (4.29) twice, once with a and b swapped. The converse is trivial.

For Eq. (4.61), we note that $(a\to c)\wedge(b\to c)\le a\overset{c}{\equiv}b$.

Eq. (4.62) follows from Eq. (4.29) since $a\le a'\to c$ and $b\le b'\to c$; conversely, substituting $a'\to c$ for a and $b'\to c$ for b into Eq. (4.62), we obtain Eq. (4.29).

Eq. (4.63) follows from Eq. (4.62) since $(a\to c)\wedge(b\to c)\le a\overset{c}{=}b$.

To obtain Eq. (4.64), apply Eq. (4.62) twice.

Eq. (4.65) follows from Eqs. (4.61) and Eq. (4.63). To obtain the 3OA law in the form of Eq. (4.61), substitute $a'\to c$ for a and $b'\to c$ for b into Eq. (4.65)

Eq. (4.66) follows immediately from Eq. (4.65). For the converse, substitute $(a\to c)'$ for a and $(b\to c)'$ for b into Eq. (4.66). Using $(a\to c)'\to c=a\to c$ and similarly for a', b, and b', we have:

$$((a\to c)'\to c)\wedge((a\to c)'\overset{c}{\equiv}(b\to c)')\le(a\to c)'\overset{c}{=}(b\to c)'$$
$$(a\to c)\wedge(a\overset{c}{\equiv}b)\le((a\to c)\wedge(b\to c))\vee((a\to c)'\wedge(b\to c)')$$
$$(a\to c)\wedge(a\overset{c}{\equiv}b)\le(((a\to c)\wedge(b\to c))\vee((a\to c)'\wedge(b\to c)'))\wedge(a\to c)$$
$$(a\to c)\wedge(a\overset{c}{\equiv}b)\le((a\to c)\wedge(b\to c))\vee 0$$
$$\text{using F-H (Th. 3.1.3)}$$
$$(a\to c)\wedge(a\overset{c}{\equiv}b)\le b\to c$$

which is the 3OA law, Eq. (4.29). □

Theorem 4.4.2. *An* OML *in which any of the equations*

$$a\wedge(a\overset{c}{=}b)\le b'\to c \tag{4.67}$$
$$b\wedge(a\overset{c}{=}b)\le a'\to c \tag{4.68}$$
$$a\wedge(a\overset{c}{\equiv}b)\le b'\to c \tag{4.69}$$
$$a'\wedge(a\overset{c}{\equiv}b)\le b\to c \tag{4.70}$$

holds is a 3OA *and vice versa.*

Proof. **For Eq. (4.67):** To obtain the 3OA law, Eq. (4.29), from Eq. (4.67), we substitute $a\to c$ for a and $b\to c$ for b, then we use the OML identities $(a\to c)\to c=a'\to c$, $(b\to c)\to c=b'\to c$, and $(b'\to c)\to c=b\to c$.

For the converse, since $x \le x'{\rightarrow}y$,

$$\begin{aligned}
&a \wedge ((a \wedge b) \vee ((a{\rightarrow}c) \wedge (b{\rightarrow}c)))\\
&\qquad \le (a'{\rightarrow}c) \wedge (((a'{\rightarrow}c) \wedge (b'{\rightarrow}c)) \vee ((a{\rightarrow}c) \wedge (b{\rightarrow}c)))\\
&\qquad = (a'{\rightarrow}c) \wedge (a' \stackrel{c}{\equiv} b')\\
&\qquad \le b'{\rightarrow}c,
\end{aligned}$$

where the last step is an instance of Eq. (4.29).

A proof of Eq. (4.67) can also be found in Ref. [84, Th. 5.1].

For Eq. (4.68): This is a trivial variant of Eq. (4.67) obtained by swapping a and b and applying Eq. (4.56). We mention it because it is used for the `-sh` output of the program `oagen.c` [Sec. A.8 (p. 156)].

For Eq. (4.69): Since $a \stackrel{c}{=} b \le a \stackrel{c}{\equiv} b$ by Eq. (4.57), Eq. (4.69) implies the 3OA law in the form of Eq. (4.67). Conversely, since $a \le a'{\rightarrow}c$, then putting a' for a and b' for b in the 3OA law Eq. (4.29), we have $a \wedge (a \stackrel{c}{\equiv} b) \le (a'{\rightarrow}c) \wedge (a' \stackrel{c}{\equiv} b') \le b'{\rightarrow}c$. Eq. (4.69) follows since $a' \stackrel{c}{\equiv} b' = a \stackrel{c}{\equiv} b$ by Eq. (4.36).

For Eq. (4.70): This is shown equivalent to Eq. (4.69) using $a' \stackrel{c}{\equiv} b' = a \stackrel{c}{\equiv} b$. □

An open problem is whether the following analogue of Eq. (4.70),

$$a' \wedge (a \stackrel{c}{=} b) \le b{\rightarrow}c, \tag{4.71}$$

is equivalent to the 3OA law. It follows from Eq. (4.70) using Eq. (4.57). By substituting $a'{\rightarrow}c$ for a and $b'{\rightarrow}c$ for b, it implies Eq. (4.94) below, meaning that it implies the 3OA identity law of Sec. 4.5.

Theorem 4.4.3. *An* OML *in which*

$$(b'{\rightarrow}c) \wedge ((b{\rightarrow}c) \vee ((a{\rightarrow}c) \wedge (a \stackrel{c}{\equiv} b))) \le c \tag{4.72}$$

holds is a 3OA *and vice versa.*

Proof. To obtain the 3OA law Eq. (4.29),

$$\begin{aligned}
(b'{\rightarrow}c) \wedge ((b{\rightarrow}c) \vee ((a{\rightarrow}c) \wedge (a \stackrel{c}{\equiv} b))) &\le c\\
b \wedge (b'{\rightarrow}c) \wedge ((b{\rightarrow}c) \vee ((a{\rightarrow}c) \wedge (a \stackrel{c}{\equiv} b))) &\le b \wedge c
\end{aligned}$$

$$b \wedge ((b \to c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b))) \le b \wedge c \qquad \text{since } b \wedge (b' \to c) = b$$
$$b' \vee (b \wedge (b' \vee (b \wedge c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b)))) \le b' \vee (b \wedge c)$$
$$b' \vee (b \wedge c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b)) \le b \to c \qquad \text{using Eq. (3.4)}$$
$$(a \to c) \wedge (a \stackrel{c}{\equiv} b) \le b \to c$$

For the converse, starting with the 3OA law,

$$(a \to c) \wedge (a \stackrel{c}{\equiv} b) \le b \to c$$
$$(b \to c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b)) \le b \to c$$
$$(b' \to c) \wedge ((b \to c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b))) \le (b' \to c) \wedge (b \to c) = (b \to c) \wedge c$$
$$(b' \to c) \wedge ((b \to c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b))) \le c$$

□

Theorem 4.4.4. *An* OML *in which*

$$(a \to c) \wedge (a \vee (b \wedge (a \stackrel{c}{\equiv} b))) \le c \tag{4.73}$$

holds is a 3OA *and vice versa.*

Proof. This equation can be derived from the 3OA law as follows:

$$\begin{aligned}
&(a \to c) \wedge (a \vee (b \wedge (a \stackrel{c}{\equiv} b))) \\
&\qquad \le (a \to c) \wedge ((a' \to c) \vee ((b' \to c) \wedge (a \stackrel{c}{\equiv} b))) \\
&\qquad = (a'' \to c) \wedge ((a' \to c) \vee ((b' \to c) \wedge (b' \stackrel{c}{\equiv} a'))) \\
&\qquad\qquad\qquad \text{using Eqs. (4.35), (4.36)} \\
&\qquad \le c \qquad \text{using Eq. (4.72)}
\end{aligned}$$

To obtain the 3OA law, we substitute $b \to c$ for a and $a \to c$ for b in Eq. (4.73):

$$((b \to c) \to c) \wedge ((b \to c) \vee ((a \to c) \wedge ((b \to c) \stackrel{c}{\equiv} (a \to c)))) \le c.$$

Using Eqs. (3.39), (4.43), and (4.35) we obtain

$$(b' \to c) \wedge ((b \to c) \vee ((a \to c) \wedge (a \stackrel{c}{\equiv} b))) \le c,$$

which is the 3OA law in the form of Eq. (4.72). $\square$

Theorem 4.4.5. *An* OML *in which*

$$a' \wedge (a \vee (b \wedge (a \overset{c}{\equiv} b) \leq c \tag{4.74}$$

holds is a 3OA *and vice versa.*

Proof. Using $a' \leq a \to c$, Eq. (4.74) follows immediately from the OA3 law in the form of Eq. (4.73):

$$\begin{aligned} a' \wedge (a \vee (b \wedge (a \overset{c}{\equiv} b) &\leq (a \to c) \wedge (a \vee (b \wedge (a \overset{c}{\equiv} b))) \\ &\leq c \end{aligned}$$

To obtain the OA3 law, we substitute b' for a and $(a \to c)$ for b in Eq. (4.74), obtaining

$$b'' \wedge (b' \vee ((a \to c) \wedge (b' \overset{c}{\equiv} (a \to c)))) \leq c$$

We have $b' \overset{c}{\equiv} (a \to c) = a \overset{c}{\equiv} b$ by Lemma 4.2.6, so

$$\begin{aligned} b \wedge (b' \vee ((a \to c) \wedge (a \overset{c}{\equiv} b))) &\leq c \\ b \wedge (b' \vee ((a \to c) \wedge (a \overset{c}{\equiv} b))) &\leq b \wedge c \\ b' \vee (b \wedge (b' \vee ((a \to c) \wedge (a \overset{c}{\equiv} b)))) &\leq b' \vee (b \wedge c) \\ b' \vee ((a \to c) \wedge (a \overset{c}{\equiv} b)) &\leq b \to c \qquad \text{using Eq. (3.4)} \\ (a \to c) \wedge (a \overset{c}{\equiv} b) &\leq b \to c \end{aligned}$$

which is Eq. (4.29). $\square$

Theorem 4.4.6. *An* OML *in which*

$$a' \wedge (a \vee (b \wedge (a \overset{c}{=} b) \leq c \tag{4.75}$$

holds is a 3OA *and vice versa.*

Proof. Since $a \overset{c}{=} b \leq a \overset{c}{\equiv} b$ (Lemma 4.2.9), Eq. (4.75) follows immediately from the OA3 law in the form of Eq. (4.74).

To obtain the OA3 law from Eq. (4.75), we substitute $b\to c$ for a and $a\to c$ for b. From Lemma 4.2.9, $(b\to c)\overset{c}{=}(a\to c) = a\overset{c}{\equiv}b$. Thus

$$\begin{aligned}
(b\to c)'\wedge((b\to c)\vee((a\to c)\wedge((b\to c)\overset{c}{=}(a\to c)))) &\le c\\
(b\to c)'\wedge((b\to c)\vee((a\to c)\wedge(a\overset{c}{\equiv}b))) &\le c\\
(b\to c)'\wedge((b\to c)\vee((a\to c)\wedge(a\overset{c}{\equiv}b))) &\le (b\to c)'\wedge c\\
(b\to c)\vee((b\to c)'\wedge((b\to c)\vee((a\to c)\wedge(a\overset{c}{\equiv}b)))) &\le (b\to c)\vee((b\to c)'\wedge c)\\
&= b\to c\\
(b\to c)\vee((a\to c)\wedge(a\overset{c}{\equiv}b)) &\le b\to c \qquad \text{using Eq. (3.4)}\\
(a\to c)\wedge(a\overset{c}{\equiv}b) &\le b\to c
\end{aligned}$$

which is Eq. (4.29). ☐

The following theorem shows a version of the 3OA law with perpendicularity hypotheses and four variables instead of three.

Theorem 4.4.7. *An* OML *in which*

$$\begin{aligned}
&a\perp b \quad \& \quad c\perp d\\
&\qquad\Rightarrow\quad (a\vee b)\wedge(c\vee d)\le b\vee(a\wedge(c\vee((a\vee c)\wedge(b\vee d))))
\end{aligned}\tag{4.76}$$

holds is a 3OA *and vice versa.*

Proof. See Theorem 4.9 of Ref. [76]. ☐

The 3OA law is a consequence of the modular law $a\wedge(b\vee(a\wedge c)) = (a\wedge b)\vee(a\wedge c)$ (Th. 7.2.2).

Theorem 4.4.8. *Let* $d = a\to c$, $e = (a'\to c)\wedge(b'\to c)$, *and* $f = (a\to c)\wedge(b\to c)$. *Then an* OML *in which*

$$d\wedge(e\vee(d\wedge f)) = (d\wedge e)\vee(d\wedge f) \tag{4.77}$$

holds is a 3OA *and vice versa. In other words, the* 3OA *law holds in any modular ortholattice.*

Proof.

$$
\begin{aligned}
&d\wedge(e\vee(d\wedge f))\\
&\quad =(a\to c)\wedge(((a'\to c)\wedge(b'\to c))\vee((a\to c)\wedge((a\to c)\wedge(b\to c))))\\
&\quad =(a\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c)))\\
&\quad =(a\to c)\wedge(b\to c) \qquad\qquad \text{by the 3OA law Eq. (4.61)}\\
&\quad =((a\to c)\wedge(b\to c)\wedge c)\vee(a\to c)\wedge((a\to c)\wedge(b\to c)))\\
&\quad =((a\to c)\wedge((a'\to c)\wedge(b'\to c)))\vee((a\to c)\wedge((a\to c)\wedge(b\to c)))\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{by Lemma 3.2.6}\\
&\quad =(d\wedge e)\vee(d\wedge f).
\end{aligned}
$$

□

Theorem 4.4.9 is interesting because it appears to "weaken" the 3OA law's inequality to a commutes relationship where ordering can't be inferred directly, but in fact the result is equivalent.

Theorem 4.4.9. *An* OML *in which*

$$(a\to c)\wedge(a\overset{c}{\equiv}b)\ C\ b\to c \tag{4.78}$$

holds is a 3OA *and vice versa.*

Proof. The law $(a\to c)\wedge(a\overset{c}{\equiv}b)\ C\ b\to c$ follows trivially from 3OA in the form $(a\to c)\wedge(a\overset{c}{\equiv}b)$ $\le b\to c$, using Eq. (3.26).

For the converse, we assume the law $(a\to c)\wedge(a\overset{c}{\equiv}b)\ C\ b\to c$ as well as its consequence $(b'\to c)\wedge(a\overset{c}{\equiv}b)\ C\ a'\to c$ that follows from Lemma 4.2.6. Applying the commutativity expansion

$$xCy \quad\Leftrightarrow\quad x\le y\vee(y'\wedge x)$$

to $(a\to c)\wedge(a\overset{c}{\equiv}b)\ C\ b\to c$, we have

$$(a\to c)\wedge(a\overset{c}{\equiv}b)\le(b\to c)\vee((b\to c)'\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)) \tag{4.79}$$

Similarly, $(b'\to c)\wedge(a\overset{c}{\equiv}b)\ C\ a'\to c$, so

$$(b'\to c)\wedge(a\overset{c}{\equiv}b)\le(a'\to c)\vee((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)) \tag{4.80}$$

We need to show that the rightmost disjunct in Eq. (4.79) is 0 in order to obtain the 3OA law.

$$\begin{aligned}
&(b\to c)'\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\\
&\quad=(a\to c)\wedge(b\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\\
&\qquad\qquad\text{since } (b\to c)'\le b'\to c\\
&\quad\le(a\to c)\wedge(b\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\\
&\qquad\quad\wedge((a'\to c)\vee((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\qquad\qquad\text{using Eq. (4.80)}\\
&\quad=(b\to c)'\wedge(a\to c)\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\\
&\qquad\wedge((a'\to c)\vee((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\qquad\qquad\text{by rearranging terms.}
\end{aligned}$$

Using F-H with $(a\to c)\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\ C\ a'\to c$ [from $a\to c\,C\,a'\to c, (b'\to c)\wedge(a\overset{c}{\equiv}b)\,C\,a'\to c$] and $a'\to c\,C\,(a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)$, we get

$$\begin{aligned}
&(b\to c)'\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\\
&\quad=(b\to c)'\wedge(((a\to c)\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\wedge(a'\to c))\\
&\qquad\vee((a\to c)\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\wedge((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\quad\le(b\to c)'\wedge(((a\to c)\wedge(b'\to c)\wedge(a'\to c))\\
&\qquad\vee((a\to c)\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)\wedge((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\quad=(b\to c)'\wedge(((a\to c)\wedge(b'\to c)\wedge(a'\to c))\\
&\qquad\vee((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\qquad\qquad\text{since } (a'\to c)'\le a\to c\\
&\quad=(b\to c)'\wedge(((a\to c)\wedge(b\to c)\wedge c)\vee(((a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\qquad\qquad\text{since } (a\to c)\wedge(b'\to c)\wedge(a'\to c)=((a\to c)\wedge(b\to c)\wedge c\\
&\qquad\qquad\text{by Part 2 of Lemma 3.2.6.}
\end{aligned}$$

Using F-H and Part 1 of Lemma 3.2.6, which implies $(a\to c)\wedge(b\to c)\wedge c\ C\ (b\to c)'$ and

$(a\to c)\wedge(b\to c)\wedge c\ C\ (a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b)$,

$$\begin{aligned}(b\to c)'\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)&\\ &=((b\to c)'\wedge(a\to c)\wedge(b\to c)\wedge c)\\ &\quad\vee((b\to c)'\wedge(a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b))\\ &=0\vee((b\to c)'\wedge(a'\to c)'\wedge(b'\to c)\wedge(a\overset{c}{\equiv}b))\\ &=0\vee((b\to c)'\wedge(a'\to c)'\wedge(a\overset{c}{\equiv}b))\\ &\qquad\qquad\text{since } (b\to c)'\le(b'\to c)\\ &=0\vee 0 \qquad\qquad (4.81)\\ &\qquad\qquad\text{since } (b\to c)'\wedge(a'\to c)' \text{ commutes with both}\\ &\qquad\qquad\text{terms of } (a\overset{c}{\equiv}b) \text{ and zeroes them out.}\end{aligned}$$

From Eqs. (4.79) and (4.81), we conclude the 3OA law,

$$(a\to c)\wedge(a\overset{c}{\equiv}b)\le(b\to c)$$

□

The following theorem expresses the 3OA law in the form $s\to c=t\to c$, which has the same structure as the conclusion of the 3OA identity law [Eq. (4.104) below]. It may be useful for studying the 3OA identity law and in particular the conjecture that the 3OA identity law implies the 3OA law.

Theorem 4.4.10. *An* OML *in which*

$$((a\to c)\wedge(a\overset{c}{\equiv}b))\to c=((b\to c)\wedge(a\overset{c}{\equiv}b))\to c \qquad (4.82)$$

holds is a 3OA *and vice versa.*

Proof. That Eq. (4.82) follows from the 3OA law in the form $(a\to c)\wedge(a\overset{c}{\equiv}b)=(b\to c)\wedge(a\overset{c}{\equiv}b)$ is a trivial consequence of equality.

Conversely, expanding Eq. (4.82) we have

$$\begin{aligned}((a\to c)\wedge(a\overset{c}{\equiv}b))'\vee(((a\to c)\wedge(a\overset{c}{\equiv}b))\wedge c)&\\ &=((b\to c)\wedge(a\overset{c}{\equiv}b))'\vee(((b\to c)\wedge(a\overset{c}{\equiv}b))\wedge c)\end{aligned}$$

$$c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)$$

$$\leq ((b\to c)\wedge(a\overset{c}{\equiv}b))'\vee(c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)) \tag{4.83}$$

A substitution instance of Eq. (4.82) is

$$((a'\to c)\wedge(a'\overset{c}{\equiv}b'))\to c=((b'\to c)\wedge(a'\overset{c}{\equiv}b'))\to c,$$

from which we obtain similarly

$$c\wedge(a'\to c)\wedge(a'\overset{c}{\equiv}b')\leq((b'\to c)\wedge(a'\overset{c}{\equiv}b'))'\vee(c\wedge(b'\to c)\wedge(a'\overset{c}{\equiv}b')) \tag{4.84}$$

Using $c\wedge(a'\to c)=c\wedge(a\to c)$, $c\wedge(b'\to c)=c\wedge(b\to c)$, and $(a'\overset{c}{\equiv}b')=(a\overset{c}{\equiv}b)$, we can express Eq. (4.84) as

$$c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\leq((b'\to c)\wedge(a\overset{c}{\equiv}b))'\vee(c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)) \tag{4.85}$$

Combining Eq. (4.83) and Eq. (4.85),

$$\begin{aligned}
&c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\\
&\quad\leq(((b\to c)\wedge(a\overset{c}{\equiv}b))'\vee(c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)))\\
&\qquad\wedge(((b'\to c)\wedge(a\overset{c}{\equiv}b))'\vee(c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)))
\end{aligned} \tag{4.86}$$

Note the four commutativity relations

$$\begin{aligned}
&((b\to c)\wedge(a\overset{c}{\equiv}b))'\ C\ c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b),\\
&c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)\ C\ ((b'\to c)\wedge(a\overset{c}{\equiv}b))'\\
&\qquad\qquad\qquad\qquad[\text{using } c\wedge(b'\to c)=c\wedge(b\to c)],\\
&((b'\to c)\wedge(a\overset{c}{\equiv}b))'\ C\ c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b),\ \text{and}\\
&c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)\ C\ ((b\to c)\wedge(a\overset{c}{\equiv}b))',
\end{aligned}$$

allowing us to apply M-H (Theorem 3.1.4) to Eq. (4.86), yielding

$$\begin{aligned}
&c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\\
&\quad\leq(((b\to c)\wedge(a\overset{c}{\equiv}b))'\wedge((b'\to c)\wedge(a\overset{c}{\equiv}b))')\\
&\qquad\vee(((b\to c)\wedge(a\overset{c}{\equiv}b))'\wedge(c\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)))
\end{aligned}$$

$$\begin{aligned}
&\vee ((c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b)) \wedge ((b' \to c) \wedge (a \overset{c}{\equiv} b))') \\
&\vee ((c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b)) \wedge (c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b))) \\
= {}& (((b \to c) \wedge (a \overset{c}{\equiv} b))' \wedge ((b' \to c) \wedge (a \overset{c}{\equiv} b))') \\
&\vee 0 \vee 0 \vee (c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b)) \qquad (4.87)
\end{aligned}$$

where we used $c \wedge (b \to c) = c \wedge (b' \to c)$ to achieve the second cancellation. Since $(a \to c) \wedge (b \to c) \le (b \to c) \wedge (a \overset{c}{\equiv} b)$ and $(a' \to c) \wedge (b' \to c) \le (b' \to c) \wedge (a \overset{c}{\equiv} b)$,

$$\begin{aligned}
&((b \to c) \wedge (a \overset{c}{\equiv} b))' \wedge ((b' \to c) \wedge (a \overset{c}{\equiv} b))' \\
&\quad \le ((a \to c) \wedge (b \to c))' \wedge ((a' \to c) \wedge (b' \to c))' \\
&\quad = (a \overset{c}{\equiv} b)',
\end{aligned}$$

so Eq. (4.87) gives

$$c \wedge (a \to c) \wedge (a \overset{c}{\equiv} b) \le (a \overset{c}{\equiv} b)' \vee (c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b))$$

Multiplying both sides by $a \overset{c}{\equiv} b$,

$$\begin{aligned}
c \wedge (a \to c) \wedge (a \overset{c}{\equiv} b) &\le ((a \overset{c}{\equiv} b)' \vee (c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b))) \wedge (a \overset{c}{\equiv} b) \\
&= 0 \vee (c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b))
\end{aligned}$$

using F-H. By symmetry the other direction also holds, so

$$c \wedge (a \to c) \wedge (a \overset{c}{\equiv} b) = c \wedge (b \to c) \wedge (a \overset{c}{\equiv} b). \qquad (4.88)$$

Combining Eqs. (4.82) and (4.88),

$$\begin{aligned}
&(c \wedge ((a \to c) \wedge (a \overset{c}{\equiv} b)) \vee (((a \to c) \wedge (a \overset{c}{\equiv} b)) \to c)' \\
&\quad = (c \wedge ((b \to c) \wedge (a \overset{c}{\equiv} b)) \\
&\qquad \vee (((b \to c) \wedge (a \overset{c}{\equiv} b)) \to c)' \qquad (4.89)
\end{aligned}$$

Using the OML identity $(c \wedge x) \vee (x \to c)' = x$, Eq. (4.89) becomes

$$(a \to c) \wedge (a \overset{c}{\equiv} b) = (b \to c) \wedge (a \overset{c}{\equiv} b) \le b \to c$$

which is the 3OA law Eq. (4.29). □

4.5 The orthoarguesian identity laws

An interesting law that holds in an nOA lattice is the nOA **identity law** given by the following Theorem.

Theorem 4.5.1. *In any nOA we have:*

$$a_1 \stackrel{a_3}{\equiv}_n a_2 = 1 \qquad \Leftrightarrow \qquad a_1 \to a_3 = a_2 \to a_3 \tag{4.90}$$

This also means that $a_1 \stackrel{a_3}{\equiv}_n a_2$ being equal to one is a relation of equivalence.

Proof. See Ref. [76, Th. 4.10] for $n = 3,4$. The extension to all n by induction is straightforward. (*Erratum:* This theorem also appears as Theorem 12 of Ref. [105, p. 767], where a_3 is incorrectly called a_n.) Note that the reverse direction, which we will sometimes omit when stating this law, holds in all OMLs by Eq. (4.41), p. 46. □

An immediate consequence of Eq. (4.90) is the transitive law

$$a \stackrel{d}{\equiv}_i b = 1 \qquad \& \qquad b \stackrel{d}{\equiv}_j c = 1 \qquad \Rightarrow \qquad a \stackrel{d}{\equiv}_k c = 1. \tag{4.91}$$

where above we have used the notation of Def. 4.2.4. (Erratum: Note that the variable d must be the same in the hypotheses and conclusion. This requirement was omitted in Eq. (10) of [105, p. 768].) While weaker than the nOA law where $n = \max(i, j, k)$ (verified to be strictly weaker for $i = j = k = 3,4$), Eq. (4.91) cannot be derived from the OML axioms [76]. Note that except for the variable corresponding to a_3, the implicit or "internal" variables may be different in each $\stackrel{d}{\equiv}_i$ operation and are therefore irrelevant to the conclusion. The only effect they have is to make the strength of the condition stronger or weaker depending on their assignments, although never stronger than the nOA law.

The nOA identity law bears a resemblance to the OML law in the form $a \equiv b = 1 \Leftrightarrow a = b$ (and in fact reduces to it when $c = 0$ in $a \stackrel{c}{\equiv}_i b$). Thus is it natural to think that they might be equivalent to the nOA laws. This is known as the **orthoarguesian identity conjecture** [76], which asks whether the nOA laws can be derived, in an OML, from Eq. (4.90). Tests run against several million finite lattices (for $n = 3$) have not found a counterexample, but the conjecture has so far defied attempts to find a proof.

Conjecture 4.5.2. *Any OML in which the* nOA *identity law Eq. (4.90) holds is an* nOA *and vice versa.*

A **quasi-identity** is an inference of the form $s_1 = t_1, \ldots, s_n = t_n \Rightarrow s = t$, where s_i, t_i, s, t are terms (polynomials in lattice variables) and $n \geq 0$. When $n = 0$, a quasi-identity is also an identity. A **quasi-variety** is the class of all algebras that satisfy a given set of quasi-identities. The nOA identity law is a quasi-identity, and it generates a quasi-variety when added to the equational axioms for an OML. Conjecture 4.5.2 can be subdivided into two conjectures, the first weaker than the second: (1) Is the quasi-variety generated by the nOA identity law a variety? (2) Is the quasi-variety generated by the nOA identity law the same as the variety nOA?

An affirmative answer to the second question (i.e. Conjecture 4.5.2 itself) would provide us with a powerful tool to prove new equivalents to the nOA laws. It turns out that it is often much easier to derive the nOA identity law from a conjectured nOA law equivalent than it is to derive the nOA law itself. For example, under the assumption that the 3OA identity law implies the 3OA law, all of the following conditions would be established as equivalents to the 3OA law (where $a\,C\,b$ means $a = (a \vee b) \wedge (a \vee b')$ i.e. a commutes with b):

$$(a\to c) \wedge (a\overset{c}{\equiv}b)\ C\ b\to c \tag{4.92}$$

$$(a\to c) \wedge (a\overset{c}{\equiv}b)\ C\ (b\to c) \wedge (a\overset{c}{\equiv}b) \tag{4.93}$$

$$(a'\to c)' \wedge (a\overset{c}{\equiv}b) \leq b\to c \tag{4.94}$$

$$(a'\to c)' \wedge (a\overset{c}{\equiv}b)\ C\ b\to c \tag{4.95}$$

$$(a'\to c)' \wedge (a\overset{c}{\equiv}b)\ C\ (b\to c) \wedge (a\overset{c}{\equiv}b) \tag{4.96}$$

$$c \wedge (a\to c) \wedge (a\overset{c}{\equiv}b) \leq (b\to c) \tag{4.97}$$

$$c \wedge (a\to c) \wedge (a\overset{c}{\equiv}b)\ C\ (b\to c) \tag{4.98}$$

$$c \wedge (a\to c) \wedge (a\overset{c}{\equiv}b)\ C\ (b\to c) \wedge (a\overset{c}{\equiv}b) \tag{4.99}$$

$$((a\to c) \wedge (a\overset{c}{\equiv}b))\to c = ((b\to c) \wedge (a\overset{c}{\equiv}b))\to c \tag{4.100}$$

$$((a\to c) \wedge (a\overset{c}{\equiv}b))\to c\ C\ ((b\to c) \wedge (a\overset{c}{\equiv}b))\to c \tag{4.101}$$

At the present time, only Eqs. (4.92) and (4.100) from the above set of conditions are known to be equivalent to the 3OA law. Denoting the 3OA law [Eq. (4.24) for $n = 3$] and the 3OA identity law [Eq. (4.90)] by OA-3 and OI-3 respectively, the currently known relationships among the above conditions are shown by the following theorem. (Note that "$\Rightarrow$" below means "the right-hand equation can be proved from the axiom system of OML $+$ the left-hand equation added as an axiom.")

Theorem 4.5.3. *The following relationships hold in all* OML*s.*

$$\begin{array}{lllllllll}
\text{OA-3} & \Leftrightarrow & \text{Eq. (4.92)} & \Rightarrow & \text{Eq. (4.93)} & \Rightarrow & \text{OI-3} & & \\
\text{OA-3} & \Rightarrow & \text{Eq. (4.94)} & \Rightarrow & \text{Eq. (4.95)} & \Rightarrow & \text{Eq. (4.96)} & \Rightarrow & \text{OI-3} \\
\text{OA-3} & \Rightarrow & \text{Eq. (4.97)} & \Leftrightarrow & \text{Eq. (4.98)} & \Leftrightarrow & \text{Eq. (4.99)} & \Rightarrow & \text{OI-3} \\
\text{OA-3} & \Leftrightarrow & \text{Eq. (4.100)} & \Rightarrow & \text{Eq. (4.101)} & \Rightarrow & \text{OI-3.} & &
\end{array}$$

Proof. **(1) For OA-3 $\Leftrightarrow$ Eq. (4.92)**: See Theorem 4.4.9.

(2) For Eq. (4.92) $\Rightarrow$ Eq. (4.93): Since $(a\to c)\wedge(a\overset{c}{\equiv}b)C(b\to c)$ by Eq. (4.92) and $(a\to c)\wedge(a\overset{c}{\equiv}b)C(a\overset{c}{\equiv}b)$ by Eq. (3.26), we conclude $(a\to c)\wedge(a\overset{c}{\equiv}b)C(b\to c)\wedge(a\overset{c}{\equiv}b)$ by Eq. (3.27).

(3) For Eq. (4.93) $\Rightarrow$ OI-3 [and Eq. (4.96) $\Rightarrow$ OI-3]: Using the OI-3 hypothesis, $a\overset{c}{\equiv}b=1$, we substitute 1 for $a\overset{c}{\equiv}b$ into Eq. (4.93) [Eq. (4.96)], as well as the version of that equation with a and b negated, using $a'\overset{c}{\equiv}b' = a\overset{c}{\equiv}b$ by Eq. (4.36). This results in the pair of commutation relationships $a\to c\ C\ b\to c$ and $a'\to c\ C\ b'\to c$ [$a\to c\ C\ b'\to c$ and $b\to c\ C\ a'\to c$]. From Theorem 3.2.4, this also implies $a\to c\ C\ b'\to c$ and $a'\to c\ C\ b\to c$ [$a\to c\ C\ b\to c$ and $b'\to c\ C\ a'\to c$]. So together we have the four commutativity relations $a\to c\ C\ b\to c$, $a'\to c\ C\ b'\to c$, $a\to c\ C\ b'\to c$, and $b\to c\ C\ a'\to c$. Combined with $a\to c\ C\ a'\to c$ and $b\to c\ C\ b'\to c$ by Theorem 3.1.2 (and the fact that any term commutes with itself), we have that any two terms from the set $a\to c$, $a'\to c$, $b\to c$, and $b'\to c$ commute. Thus all terms in the hypothesis $a\overset{c}{\equiv}b=1$ are distributive, so

$$\begin{aligned}
1 &= ((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))\\
&= ((a\to c)\vee((a'\to c)\wedge(b'\to c)))\wedge((b\to c)\vee((a'\to c)\wedge(b'\to c)))\\
&= ((a\to c)\vee(a'\to c))\wedge((a\to c)\vee(b'\to c))\\
&\qquad\wedge((b\to c)\vee(a'\to c))\wedge((b\to c)\vee(b'\to c))\\
&\le (a\to c)\vee(a'\to c)
\end{aligned}$$

Therefore,

$$\begin{aligned}
(a\to c)\vee(b'\to c) &= 1\\
((a\to c)\vee(b'\to c))\wedge(a\to c)' &= (a\to c)'\\
((a\to c)\wedge(a\to c)')\vee((b'\to c)\wedge(a\to c)' &= (a\to c)'\\
((b'\to c)\wedge(a\to c)' &= (a\to c)'
\end{aligned}$$

so $(a\to c)'\le b'\to c$. Similarly, $(b\to c)'\le a'\to c$. This satisfies the hypotheses of Corollary 3.2.5, so by that corollary and Eq. (3.47), $a\to c = b\to c$, which is the conclusion of OI-3.

(4) For OA-3 $\Rightarrow$ Eq. (4.94): $(a'\to c)' \le a\to c$, so from OA-3, $(a'\to c)'\wedge(a\overset{c}{\equiv}b) \le (a\to c)\wedge(a\overset{c}{\equiv}b) \le (b\to c)$.

(5) For Eq. (4.94) $\Rightarrow$ Eq. (4.95): Comparable terms commute by Eq. (3.26).

(6) For Eq. (4.95) $\Rightarrow$ Eq. (4.96): Same reasoning as for part (2).

(7) For Eq. (4.96) $\Rightarrow$ OI-3: See part (3).

(8) For OA-3 $\Rightarrow$ Eq. (4.97): Same reasoning as for part (4), since $c\wedge(a\to c) \le a\to c$.

(9) For Eq. (4.97) $\Rightarrow$ Eq. (4.98): Same reasoning as for part (5).

(10) For Eq. (4.97) $\Leftarrow$ Eq. (4.98): Follows from parts (11) and (12) below.

(11) For Eq. (4.98) $\Rightarrow$ Eq. (4.99): Same reasoning as for part (2).

(12) For Eq. (4.97) $\Leftarrow$ Eq. (4.99): Denote $(a\to c)\wedge(b\to c)\wedge c$ by U ("universally commutes"). Let $S = \{a\to c, a'\to c, b\to c, b'\to c, c\}$. Recall from Lemma 3.2.6 that:

1. U commutes with any polynomial built from the terms in S,
2. U is less than or equal to any product of terms from S, and
3. U is equal to the product of any subset of three or more terms from S that contains both variables a and b.

From two instances of Eq. (4.99) and using $c\wedge(a\to c) = c\wedge(a'\to c)$,

$$c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\ C\ (b\to c)\wedge(a\overset{c}{\equiv}b)$$
$$c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\ C\ (b'\to c)\wedge(a\overset{c}{\equiv}b)$$

Thus

$$\begin{aligned} c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b) &\le (c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge(b\to c)\wedge(a\overset{c}{\equiv}b)) \\ &\qquad \vee (c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))') \\ &= (U\wedge(a\overset{c}{\equiv}b)) \\ &\qquad \vee (c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))') \\ &= U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))') \end{aligned}$$

and similarly,

$$c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b) \le U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b'\to c)\wedge(a\overset{c}{\equiv}b))').$$

Combining,

$$\begin{aligned} c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b) \le (U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))'))\\ \wedge(U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b'\to c)\wedge(a\overset{c}{\equiv}b))')). \end{aligned}$$

Since U commutes with all terms, from M-H (Th 3.1.4) we obtain

$$\begin{aligned} c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b) \le{}& (U\wedge U)\\ &\vee(U\wedge(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b'\to c)\wedge(a\overset{c}{\equiv}b))'))\\ &\vee((c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))')\wedge U)\\ &\vee((c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))')\\ &\quad\wedge(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge((b'\to c)\wedge(a\overset{c}{\equiv}b))'))\\ \le{}& U\vee U\vee U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\\ &\wedge((b\to c)\wedge(a\overset{c}{\equiv}b))'\wedge((b'\to c)\wedge(a\overset{c}{\equiv}b))')\\ ={}& U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge(((b\to c)\wedge(a\overset{c}{\equiv}b))\\ &\vee((b'\to c)\wedge(a\overset{c}{\equiv}b)))'). \end{aligned}$$

Note that $(a\to c)\wedge(b\to c)\le(b\to c)\wedge(a\overset{c}{\equiv}b)$ and $(a'\to c)\wedge(b'\to c)\le(b'\to c)\wedge(a\overset{c}{\equiv}b)$, so

$$\begin{aligned} (a\overset{c}{\equiv}b) &= ((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))\\ &\le((b\to c)\wedge(a\overset{c}{\equiv}b))\vee((b'\to c)\wedge(a\overset{c}{\equiv}b)) \end{aligned} \tag{4.102}$$

i.e.

$$(((b\to c)\wedge(a\overset{c}{\equiv}b))\vee((b'\to c)\wedge(a\overset{c}{\equiv}b)))'\le(a\overset{c}{\equiv}b)'. \tag{4.103}$$

Hence

$$\begin{aligned} c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b) &\le U\vee(c\wedge(a\to c)\wedge(a\overset{c}{\equiv}b)\wedge(a\overset{c}{\equiv}b)')\\ &= U\vee 0 = U\\ &\le b\to c. \end{aligned}$$

(13) For Eq. (4.99) $\Rightarrow$ OI-3: We use the OI-3 hypothesis, $a\overset{c}{\equiv}b=1$, to substitute 1 for $a\overset{c}{\equiv}b$

in Eq. (4.99), as well as into its equivalent version with a and b negated. Since $c\wedge(a\to c) = c\wedge(a'\to c) = (a'\to c)\wedge(a\to c)$, one of these substitutions gives us $(a'\to c)\wedge(a\to c)\ C\ b'\to c$. Since $a'\to c\ C\ a\to c$, the GSB theorem, Eq. (3.30), yields $(a'\to c)\wedge(b'\to c)\ C\ a\to c$. Thus

$$\begin{aligned}
(a\to c)\wedge 1 &= (a\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c)))\\
&\qquad \text{using the OI-3 hypothesis}\\
&= ((a\to c)\wedge((a\to c)\wedge(b\to c)))\vee((a\to c)\wedge((a'\to c)\wedge(b'\to c)))\\
&\qquad \text{since } (a\to c)\ C\ ((a\to c)\wedge(b\to c))\\
&\qquad\ \text{and } (a\to c)\ C\ (a'\to c)\wedge(b'\to c)\\
&= ((a\to c)\wedge(b\to c))\vee(((a\to c)\wedge(b\to c))\wedge c)\\
&\qquad \text{using Lemma 3.2.6 for the second conjunct}\\
&\le b\to c.
\end{aligned}$$

Similarly, $b\to c \le a\to c$.

(14) For OA-3 $\Leftrightarrow$ Eq. (4.100): See Theorem 4.4.10.

(15) For Eq. (4.100) $\Rightarrow$ Eq. (4.101): Same reasoning as for part (5).

(16) For Eq. (4.101) $\Rightarrow$ OI-3: Using the OI-3 hypothesis, we substitute 1 for $a\stackrel{c}{\equiv}b$ into Eq. (4.101) to obtain $(a\to c)\to c\ C\ (b\to c)\to c$. Since $(a\to c)\to c = a'\to c$ and similarly for b, we have $a'\to c\ C\ b'\to c$. Doing the same with a and b negated, we also have $a\to c\ C\ b\to c$. The rest of the proof is the same as for part (3) above. $\square$

4.5.1 Equivalent forms of the 3OA identity law

The following theorem shows that the 3OA identity law can be viewed as taking an OR (join) condition to a stronger AND (meet) condition.

Theorem 4.5.4. *In any* OML*, the* 3OA *identity law,*

$$a\stackrel{c}{\equiv}b = 1 \qquad\Rightarrow\qquad a\to c = b\to c \tag{4.104}$$

is equivalent to the following condition:

$$\begin{aligned}
&((a\to c)\equiv(b\to c))\vee((a'\to c)\equiv(b'\to c)) = 1\\
&\qquad\Rightarrow\quad ((a\to c)\equiv(b\to c))\wedge((a'\to c)\equiv(b'\to c)) = 1
\end{aligned}\tag{4.105}$$

Proof. Use Eq. (4.40) for the hypothesis. Apply Eq. (4.104) twice, the second time also apply Eq. (3.47), apply Eq. (3.1) to each conclusion, and conjoin them. The recovery of Eq. (4.104) should be obvious. ☐

The next theorem expresses the 3OA identity law in forms that have separate variables on the left- and right-hand sides of the conclusion.

Theorem 4.5.5. *In any* OML, *the* 3OA *identity law Eq. (4.104) is equivalent to either of the following conditions:*

$$a \overset{c}{\equiv} b = 1 \qquad \Rightarrow \qquad (a \to b)' \le c \tag{4.106}$$

$$a \overset{c}{\equiv} b = 1 \qquad \Rightarrow \qquad a' \le b \to c \tag{4.107}$$

Proof. **For Eq. (4.106):** Starting with the conclusion of Eq. (4.104), we obtain Eq. (4.106) as follows:

$$\begin{aligned} b \to c &\le a \to c \\ (a \to c) \vee (b \to c) &\le a \to c \\ (a' \to c) \wedge ((a \to c) \vee (b \to c)) &\le (a' \to c) \wedge (a \to c) \\ &= c \wedge (a \to c) \\ (a' \to c) \wedge ((a \to c) \vee (b \to c)) &\le c \\ a \wedge (a' \vee b') = (a \to b)' &\le c \end{aligned}$$

where for the last line we used $a \le a' \to c$, $a' \le a \to c$, $b' \le b \to c$.

For the converse, we substitute $(b \to c)'$ for b into Eq. (4.106). Its hypothesis remains the same by Lemma 4.2.6, and we transform its conclusion as follows:

$$\begin{aligned} a \wedge (a' \vee (b \to c)'') &\le c \\ &\le a \wedge c \\ a' \vee (a \wedge (a' \vee (b \to c))) &\le a' \vee (a \wedge c) = a \to c \\ a' \vee (b \to c) &\le a \to c \qquad \text{using Eq. (3.4)} \\ b \to c &\le a \to c \end{aligned}$$

Combining this with a similar derivation with a and b swapped, we arrive at the conclusion of the 3OA identity law Eq. (4.104).

For Eq. (4.107): The conclusion of Eq. (4.107) follows immediately from the conclusion of Eq. (4.104) using $a' \le a \to c$. For the converse, we substitute $(a\to c)'$ for a into Eq. (4.107). The hypothesis stays the same by Lemma 4.2.6, and the conclusion will be one direction of the conclusion of Eq. (4.104). ☐

The 3OA identity can also be expressed with a slightly stronger hypothesis.

Theorem 4.5.6. *In any* OML*, the* 3OA *identity law Eq. (4.104) is equivalent to any one of the following conditions:*

$$a \overset{c}{\equiv} b = 1 \qquad \Rightarrow \qquad a\to c = b\to c \tag{4.108}$$

$$a \overset{c}{\equiv} b = 1 \qquad \Rightarrow \qquad (a\to b)' \le c \tag{4.109}$$

$$a \overset{c}{\equiv} b = 1 \qquad \Rightarrow \qquad a' \le b\to c \tag{4.110}$$

Proof. **For Eq. (4.108):** The hypothesis of Eq. (4.108) follows immediately from the hypothesis of Eq. (4.104) using $a \le a' \to c$ and $b \le b' \to c$. Conversely, to obtain Eq. (4.104), we substitute $a'\to c$ for a and $b'\to c$ for b into Eq. (4.108) and use the OML identities $(a'\to c)\to c = a\to c$ and $(b'\to c)\to c = b\to c$.

For Eq. (4.109): The hypothesis of Eq. (4.109) follows immediately from the hypothesis of Eq. (4.106) using $a \le a'\to c$ and $b \le b'\to c$. Conversely, by substituting $a'\to c$ for a and $b'\to c$ for b into Eq. (4.109), we obtain the hypothesis of Eq. (4.104), and we transform the conclusion as follows:

$$\begin{aligned}
((a'\to c)\to(b'\to c))' &\le c\\
(a'\to c)\wedge((a'\to c)'\vee(b'\to c)') &\le c\\
(a'\to c)\wedge((a'\to c)'\vee(b'\to c)') &\le (a'\to c)\wedge c\\
(a'\to c)'\vee((a'\to c)\wedge((a'\to c)'\vee(b'\to c)')) &\le (a'\to c)'\vee((a'\to c)\wedge c)\\
&= (a'\to c)\to c\\
&= a\to c\\
(a'\to c)'\vee(b'\to c)') &\le a\to c \qquad \text{using Eq. (3.4)}\\
(b'\to c)' &\le a\to c
\end{aligned}$$

By symmetry, swapping a and b also yields hypothesis of Eq. (4.104) hypothesis but the conclusion $(a'\to c)' \le b\to c$. Combining the two conclusions, Corollary 3.2.5 gives us $a\to c$ = $b\to c$, which is the conclusion of Eq. (4.104).

For Eq. (4.110): The hypothesis of Eq. (4.110) follows immediately from the hypothesis of Eq. (4.107) using $a \le a' \to c$ and $b \le b' \to c$. Conversely, by substituting $a' \to c$ for a and $b' \to c$ for b into Eq. (4.110), we obtain the hypothesis of Eq. (4.104) and the conclusion $(a' \to c)' \le (b' \to c) \to c = b \to c$. Swapping a and b yields the same hypothesis with the conclusion $(b' \to c)' \le a \to c$. Combining the two conclusions, Corollary 3.2.5 gives us $a \to c = b \to c$, which is the conclusion of Eq. (4.104). $\square$

It is sometimes useful to work with a dual form of the nOA identity law having $2n-2$ variables. The following theorem shows several equivalent 4-variable dual forms for the 3OA identity law. Analogous versions for $n > 3$ can also be stated but involve more complicated expressions. Note that Eqs. (4.112) through (4.115) make successively “stronger” assertions (i.e. have successively weaker hypotheses). The proof shows that the weakest implies the 3OA identity law, which in turn is used to recover the strongest. Eq. (4.116) is the dual form of Eq. (4.113).

Theorem 4.5.7. *In any* OML*, the* 3OA *identity law*

$$a \stackrel{c}{\equiv} b = 1 \qquad \Rightarrow \qquad a \to c = b \to c \tag{4.111}$$

is equivalent to any of the following conditions:

$$\begin{aligned} a \perp b \quad \& \quad c \perp d \quad \& \quad (a \vee c) \wedge (b \vee d) = 0 & \\ \& \quad a' \vee ((a \vee b) \wedge (c \vee d)) = 1 \qquad \Rightarrow \qquad a \le c & \qquad (4.112) \\ aCb \quad \& \quad cCd \quad \& \quad (a \vee c) \wedge (b \vee d) = 0 & \\ \& \quad a' \vee ((a \vee b) \wedge (c \vee d)) = 1 \qquad \Rightarrow \qquad a \le c & \qquad (4.113) \\ a \perp b \quad \& \quad c \perp d \quad \& \quad (a \vee c) \wedge (b \vee d) = 0 & \\ \Rightarrow \qquad a \wedge (a' \vee ((a \vee b) \wedge (c \vee d))) \le c & \qquad (4.114) \\ aCb \quad \& \quad cCd \quad \& \quad (a \vee c) \wedge (b \vee d) = 0 & \\ \Rightarrow \qquad a \wedge (a' \vee ((a \vee b) \wedge (c \vee d))) \le c & \qquad (4.115) \\ aCb \quad \& \quad cCd \quad \& \quad (a \wedge c) \vee (b \wedge d) = 1 & \\ \& \quad a' \wedge ((a \wedge b) \vee (c \wedge d)) = 0 \qquad \Rightarrow \qquad c \le a. & \qquad (4.116) \end{aligned}$$

Proof. We obtain Eq. (4.116) from Eq. (4.113) and vice versa by first replacing each variable with its orthocomplement then using De Morgan's laws and $a'Cb' \Leftrightarrow aCb$.

We will prove the others, except Eq. (4.114), by showing Eq. (4.115) $\Rightarrow$ Eq. (4.113) $\Rightarrow$

Eq. (4.112) $\Rightarrow$ Eq. (4.111) $\Rightarrow$ Eq. (4.115). Finally, Eq. (4.115) obviously implies Eq. (4.114), and we can show Eq. (4.114) $\Rightarrow$ Eq. (4.112) with essentially the same proof as for Eq. (4.115) $\Rightarrow$ Eq. (4.113).

For Eq. (4.115) $\Rightarrow$ Eq. (4.113): The hypothesis $a' \vee ((a\vee b)\wedge(c\vee d)) = 1$ of Eq. (4.113), applied to the conclusion of Eq. (4.115), results in the conclusion of Eq. (4.113). The other hypotheses are identical.

For Eq. (4.113) $\Rightarrow$ Eq. (4.112): The hypotheses $a\perp b$ and $c\perp d$ of Eq. (4.112) imply the hypotheses $a\mathcal{C}b$ and $c\mathcal{C}d$ of Eq. (4.113) by Eqs. (3.26) and (3.25).

For Eq. (4.112) $\Rightarrow$ Eq. (4.111): The right-to-left direction of Eq. (4.111) holds in all OMLs. For the left-to-right direction, assume that the hypothesis $a\overset{c}{\equiv}b = 1$ of Eq. (4.111) holds. Let $p = (a\to c)'$, $q = (a'\to c)'$, $r = (b\to c)'$, and $s = (b'\to c)'$. It follows that

$$p\perp q \tag{4.117}$$

$$r\perp s. \tag{4.118}$$

We also have

$$\begin{aligned}(p\vee r)\wedge(q\vee s) &= ((a\to c)'\vee(b\to c)')\wedge((a'\to c)'\vee(b'\to c)')\\ &= (((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c)))'\\ &= (a\overset{c}{\equiv}b)' = 1'\\ (p\vee r)\wedge(q\vee s) &= 0. \end{aligned}\tag{4.119}$$

In any OML, $(a\to c)'\vee(a'\to c)' = (a'\vee c')\wedge(a\vee c')$, so

$$\begin{aligned}p'\vee((p\vee q)\wedge(r\vee s)) &= (a\to c)\vee(((a\to c)'\vee(a'\to c)')\\ &\qquad\wedge((b\to c)'\vee(b'\to c)'))\\ &= (a\to c)\vee(((a'\vee c')\wedge(a\vee c'))\\ &\qquad\wedge((b'\vee c')\wedge(b\vee c')))\\ &\geq (a\to c)\vee c'\\ &= a'\vee(a\wedge c)\vee c' = 1\\ p'\vee((p\vee q)\wedge(r\vee s)) &= 1 \end{aligned}\tag{4.120}$$

The hypotheses of Eq. (4.112) are satisfied by Eqs. (4.117), (4.118), (4.119), and (4.120), from which we conclude $p\le r$ i.e. $(a\to c)'\le(b\to c)'$ i.e. $b\to c\le a\to c$. Since $a\overset{c}{\equiv}b = b\overset{c}{\equiv}a$

by Eq. (4.35), the same argument proves $a \to c \le b \to c$; combining, we have the conclusion of Eq. (4.111), $a \to c = b \to c$.

For Eq. (4.111) $\Rightarrow$ Eq. (4.115): Assume that the hypotheses of Eq. (4.115) hold. Let $k = ((a \vee b) \wedge (c \vee d))' = (a' \wedge b') \vee (c' \wedge d')$. If aCb then $a'Cb'$, so $a' \to b' = a'' \vee b'$ by Eq. (3.29). Therefore,

$$\begin{aligned} b' &\le a \vee b' \\ &= a' \to b' \\ &= a \vee (a' \wedge (a' \wedge b')) \\ &\le a \vee (a' \wedge ((a' \wedge b') \vee (c' \wedge d')) \\ &= a \vee (a' \wedge k) \\ &= a' \to k \end{aligned}$$

Similarly, if cCd, then $d' \le c' \to k$. The hypothesis of Eq. (4.111) holds as follows:

$$\begin{aligned} a \overset{k}{\equiv} c &= ((a \to k) \wedge (c \to k)) \vee ((a' \to k) \vee (c' \to k)) \\ &\ge (a' \wedge c') \vee (b' \wedge d') \\ &= ((a \vee c) \wedge (b \vee d))' = 0' = 1 \end{aligned}$$

Therefore, by the conclusion of Eq. (4.111), $a \to k = c \to k$, so $c' \le c \to k = a \to k = a' \vee (a \wedge k) = a' \vee (a \wedge ((a \vee b) \wedge (c \vee d))')$, which is equivalent to $a \wedge (a' \vee ((a \vee b) \wedge (c \vee d))) \le c$ as required. $\square$

The 3OA law is equivalent to a substitution instance of von Neumann's lemma for modular lattices, Eq. (7.12), which reads: $(a \vee b) \wedge (c \vee d) = 0 \Rightarrow (a \vee c) \wedge (b \vee d) = (a \wedge b) \vee (c \wedge d)$.

Theorem 4.5.8. *Let $e = (a \to c)'$, $f = (b \to c)'$, $g = (a' \to c)'$, and $h = (b' \to c)'$. Then in any* OML, *the* 3OA *identity law is equivalent to the following condition:*

$$(e \vee f) \wedge (g \vee h) = 0 \quad \Rightarrow \quad (e \vee g) \wedge (f \vee h) = (e \wedge f) \vee (g \wedge h) \tag{4.121}$$

Proof. The hypothesis $(e \vee f) \wedge (g \vee h) = 0$ is equivalent to $((a \to c) \wedge (b \to c)) \vee ((a' \to c) \wedge (b' \to c) = a \overset{c}{\equiv} b = 1$. The conclusion $((a \to c) \wedge (a' \to c)) \vee ((b \to c) \wedge (b' \to c)) = ((a \to c) \vee (b \to c)) \wedge ((a' \to c) \vee (b' \to c))$ is equivalent to $a \to c = b \to c$ by Eq. (3.48). Thus Eq. (4.121 is equivalent to the 3OA identity law in the form of Eq. (4.104). $\square$

The 3OA identity conjecture can also be viewed as a weakening of von Neumann's lemma for modular lattices, Eq. (7.12).

Theorem 4.5.9. *In any* OML*, the* 3OA *identity law is equivalent tothe following condition:*

$$\begin{aligned} a\perp c \quad \& \quad b\perp d \quad \& \quad (a\vee b)\wedge(c\vee d)=0 \\ \Rightarrow \qquad (a\vee c)\wedge(b\vee d)=(a\wedge b)\vee(c\wedge d) \end{aligned} \tag{4.122}$$

Proof. We first show that the 3OA identity law implies Eq. (4.122). We start with two instances of the 3OA identity law in the form of Eq. (4.114).

$$\begin{aligned} a\perp c \quad \& \quad b\perp d \quad \& \quad (a\vee b)\wedge(c\vee d)=0 \\ \Rightarrow \qquad a\wedge(a'\vee((a\vee c)\wedge(b\vee d)))\le b \\ c\perp a \quad \& \quad d\perp b \quad \& \quad (c\vee d)\wedge(a\vee b)=0 \\ \Rightarrow \qquad c\wedge(c'\vee((c\vee a)\wedge(d\vee b)))\le d. \end{aligned}$$

After conjoining a and c to their respective conclusions and commuting some terms in the second instance, we have

$$\begin{aligned} a\perp c \quad \& \quad b\perp d \quad \& \quad (a\vee b)\wedge(c\vee d)=0 \\ \Rightarrow \qquad a\wedge(a'\vee((a\vee c)\wedge(b\vee d)))\le a\wedge b \\ a\perp c \quad \& \quad b\perp d \quad \& \quad (a\vee b)\wedge(c\vee d)=0 \\ \Rightarrow \qquad c\wedge(c'\vee((a\vee c)\wedge(b\vee d)))\le c\wedge d. \end{aligned}$$

Combining, we have

$$\begin{aligned} a\perp c \quad \& \quad b\perp d \quad \& \quad (a\vee b)\wedge(c\vee d)=0 \\ \Rightarrow \quad (a\wedge(a'\vee((a\vee c)\wedge(b\vee d))))\wedge(c\wedge(c'\vee((a\vee c)\wedge(b\vee d)))) \\ \le(a\wedge b)\vee(c\wedge d). \end{aligned}$$

The left-hand side of the conclusion can be transformed as follows, using M-H (Th 3.1.4) in the first step.

$$\begin{aligned} (a\wedge(a'\vee((a\vee c)\wedge(b\vee d))))\vee(c\wedge(c'\vee((a\vee c)\wedge(b\vee d)))) \\ =(a\vee c)\wedge(a\vee(c'\vee((a\vee c)\wedge(b\vee d)))) \end{aligned}$$

$$\begin{aligned}
&\wedge ((a' \vee ((a \vee c) \wedge (b \vee d))) \vee c) \\
&\wedge ((a' \vee ((a \vee c) \wedge (b \vee d))) \vee (c' \vee ((a \vee c) \wedge (b \vee d)))) \\
\geq\ & ((a \vee c) \wedge ((a \vee c) \wedge (b \vee d))) \\
=\ & (a \vee c) \wedge (b \vee d)
\end{aligned}$$

which establishes

$$\begin{aligned}
& a \perp c \quad \& \quad b \perp d \quad \& \quad (a \vee b) \wedge (c \vee d) = 0 \\
& \quad \Rightarrow \quad (a \vee c) \wedge (b \vee d) \leq (a \wedge b) \vee (c \wedge d).
\end{aligned}$$

For the other direction of the conclusion, $(a \vee c) \wedge (b \vee d) \geq (a \wedge b) \vee (c \wedge d)$ holds in any OL.

For the converse, we substitute $(a \to c)'$ for a, $(b \to c)'$ for b, $(a' \to c)'$ for c, and $(b' \to c)'$ for d in Eq. (4.122). The two orthogonality hypotheses are satisfied, resulting in the 3OA law in the form of Eq. (4.121). ☐

4.5.2 Conjectures that imply the 3OA identity conjecture

In this section, we will describe several conjectures which, if true, would imply the 3OA identity conjecture [Conjecture 4.5.2 (p. 63) for $n = 3$].

Consider the following substitution instance of the 3OA identity law expressed in the form of Eq. (4.104):

$$x \overset{c}{\equiv} y = 1 \qquad \Rightarrow \qquad x \to c = y \to c \tag{4.123}$$

where

$$\begin{aligned}
x &= d \wedge f \\
y &= e \wedge f \\
d &= a \to c \\
e &= b \to c \\
f &= a \overset{c}{\equiv} b.
\end{aligned}$$

Theorem 4.5.10. *The conclusion of Eq. (4.123) is the* 3OA *law.*

Proof. After applying Eq. (3.47) (p. 32), the conclusion of Eq. (4.123) becomes:

$$((a\to c)\wedge(a\overset{c}{\equiv}b))\to c=((b\to c)\wedge(a\overset{c}{\equiv}b))\to c$$

which is the 3OA law by Th. 4.4.10 (p. 59). ☐

It is currently unknown whether the hypothesis of Eq. (4.123) holds in all OMLs, although we could not find a finite OML in which it failed.

Conjecture 4.5.11. *The hypothesis of Eq. (4.123) holds in all* OML*s.*

If it holds, this conjecture would provide a positive answer to the 3OA identity conjecture. A generalization would answer it for all nOA.

Part of the difficulty in searching for an OML proof of the hypothesis of Eq. (4.123) is the shear size of the equation when fully expanded. The successive steps in its expansion read as follows:

$$\begin{aligned}
&x\overset{c}{\equiv}y=1\\
&(d\wedge f)\overset{c}{\equiv}(e\wedge f)=1\\
&((((d\wedge f)\to c)\wedge((e\wedge f)\to c))\vee(((d\wedge f)'\to c)\wedge((e\wedge f)'\to c)))=1\\
&(((((a\to c)\wedge(a\overset{c}{\equiv}b))\to c)\wedge(((b\to c)\wedge(a\overset{c}{\equiv}b))\to c))\\
&\quad\vee((((a\to c)\wedge(a\overset{c}{\equiv}b))'\to c)\wedge(((b\to c)\wedge(a\overset{c}{\equiv}b))'\to c)))=1\\
&(((((a\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))))\to c)\\
&\qquad\wedge(((b\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))))\to c))\\
&\quad\vee((((a\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))))'\to c)\\
&\qquad\wedge(((b\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))))'\to c)))=1.
\end{aligned}$$

Note that the length of the penultimate equation above will approximately double when we expand the four terms of the form $z\to c$ into $z'\vee(z\wedge c)$, where z is a large expression. In particular, it will have 8 instances of the expression $a\overset{c}{\equiv}b$. It will grow an additional 50% or so when all of the remaining $\to$ terms are expanded into $\vee$ and $\wedge$.

If we use Eq. (4.108) with the same substitution instances as above, its conclusion will be the same but its hypothesis will be about half as large, making it somewhat more manageable to study. A drawback is that it is stronger in the sense that it immediately implies Conjecture 4.5.11 and thus possibly more difficult to prove. On the other hand, it still passes in all of the finite lattices we tested.

Conjecture 4.5.12. *The hypothesis of*

$$x\overset{c}{=}y = 1 \qquad \Rightarrow \qquad x\to c = y\to c, \tag{4.124}$$

where the substitutions for x and y are the same as in Eq. (4.123), holds in all OML*s.*

If it holds, this conjecture would provide a positive answer to the 3OA identity conjecture.

The smaller size of the hypothesis of Eq. (4.124) is seen by expanding it as follows, using the simplification $(d\wedge f)\wedge(e\wedge f) = d\wedge e$:

$$\begin{aligned}
&x\overset{c}{=}y = 1\\
&(d\wedge f)\overset{c}{=}(e\wedge f) = 1\\
&((((d\wedge f)\to c)\wedge((e\wedge f)\to c))\vee((d\wedge f)\wedge(e\wedge f))) = 1\\
&((((d\wedge f)\to c)\wedge((e\wedge f)\to c))\vee(d\wedge e)) = 1\\
&(((((a\to c)\wedge(a\overset{c}{\equiv}b))\to c)\wedge(((b\to c)\wedge(a\overset{c}{\equiv}b))\to c))\\
&\quad\vee((a\to c)\wedge(b\to c)) = 1\\
&(((((a\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))))\to c)\\
&\qquad\wedge(((b\to c)\wedge(((a\to c)\wedge(b\to c))\vee((a'\to c)\wedge(b'\to c))))\to c))\\
&\quad\vee((a\to c)\wedge(b\to c)) = 1.
\end{aligned}$$

In particular, the penultimate equation will have 4 instances of the expression $a\overset{c}{\equiv}b$ rather than 8 when $\to$ is expanded.

If we replace $a\overset{c}{\equiv}b$ by $a\overset{c}{=}b$ in 4.124, we can achieve a further simplification of the hypothesis, but it also requires an additional conjecture for the conclusion. Specifically, we have:

Conjecture 4.5.13. *Consider the following substitution instance of the* 3OA *identity law, in the form of Eq. (4.108):*

$$x\overset{c}{=}y = 1 \qquad \Rightarrow \qquad x\to c = y\to c \tag{4.125}$$

where

$$\begin{aligned}
x &= d\wedge f\\
y &= e\wedge f\\
d &= a\to c
\end{aligned}$$

$$e = b \to c$$
$$f = a \overset{c}{\equiv} b.$$

Two statements are conjectured: (a) The hypothesis of Eq. (4.125) holds in all OML*s. (b) The conclusion of Eq. (4.125) is equivalent to the* 3OA *law.*

If both of these conjectures hold, they will prove the 3OA identity conjecture. Empirically, both of them hold for the finite lattices that we tested them against.

For comparison, the expansion of the hypothesis of Eq. (4.125) is as follows, using the simplification $(d \wedge f) \wedge (e \wedge f) = d \wedge e$:

$$\begin{aligned}
& x \overset{c}{\equiv} y = 1 \\
& (d \wedge f) \overset{c}{\equiv} (e \wedge f) = 1 \\
& ((((d \wedge f) \to c) \wedge ((e \wedge f) \to c)) \vee ((d \wedge f) \wedge (e \wedge f))) = 1 \\
& ((((d \wedge f) \to c) \wedge ((e \wedge f) \to c)) \vee (d \wedge e)) = 1 \\
& (((((a \to c) \wedge (a \overset{c}{\equiv} b)) \to c) \wedge (((b \to c) \wedge (a \overset{c}{\equiv} b)) \to c)) \\
& \quad \vee ((a \to c) \wedge (b \to c)) = 1 \\
& (((((a \to c) \wedge (((a \to c) \wedge (b \to c)) \vee (a \wedge b))) \to c) \\
& \qquad \wedge (((b \to c) \wedge (((a \to c) \wedge (b \to c)) \vee (a \wedge b))) \to c)) \\
& \quad \vee ((a \to c) \wedge (b \to c)) = 1.
\end{aligned}$$

The principle difference is the reduction of the 4 terms of the form $a \overset{c}{\equiv} b$ to the shorter form $a \overset{c}{\equiv} b$.

Chapter 5

OTHER $\mathscr{C}(H)$ EQUATIONS

5.1 Godowski's equations

In 1981, Radoslaw Godowski [26] found an infinite series of equations partly corresponding to the strong set of states [Def. 2.4.3 (p. 22)], forming a series of algebras contained in the class of all orthomodular lattices and containing the class of all Hilbert lattices (as shown by the next theorem). Importantly, there are OMLs that do not admit a strong set of states, so Godowski's equations provide us with new equational laws that extend the OML laws that hold in Hilbert lattices.

Theorem 5.1.1. *Any Hilbert lattice admits a strong set of states.*

Proof. See Ref. [105, p. 770, Th. 17]. □

We will now define the family of equations found by Godowski, introducing a special notation for them. Then we will prove that they hold in any lattice admitting a strong set of states and thus, in particular, any Hilbert lattice.

Definition 5.1.2. *Let us call the following expression the* **Godowski identity***:*

$$a_1 \overset{\gamma}{\equiv} a_n \overset{\text{def}}{=} (a_1 \to a_2) \wedge (a_2 \to a_3) \wedge \cdots \wedge (a_{n-1} \to a_n) \wedge (a_n \to a_1), \quad n = 3, 4, \ldots \tag{5.1}$$

We define $a_n \overset{\gamma}{\equiv} a_1$ in the same way with variables a_i and a_{n-i+1} swapped; in general $a_i \overset{\gamma}{\equiv} a_j$ will be an expression with $|j-i|+1 \geq 3$ variables $a_i, \ldots, a_j$ first appearing in that order. For completeness and later use (Theorem 5.1.8) we define $a_i \overset{\gamma}{\equiv} a_i \overset{\text{def}}{=} (a_i \to a_i) = 1$ and $a_i \overset{\gamma}{\equiv} a_{i+1} \overset{\text{def}}{=} (a_i \to a_{i+1}) \wedge (a_{i+1} \to a_i) = a_i \equiv a_{i+1}$, the last equality holding in any OML.

Theorem 5.1.3. *Godowski's equations* [26]

$$a_1 \overset{\gamma}{\equiv} a_3 = a_3 \overset{\gamma}{\equiv} a_1 \tag{5.2}$$
$$a_1 \overset{\gamma}{\equiv} a_4 = a_4 \overset{\gamma}{\equiv} a_1 \tag{5.3}$$
$$a_1 \overset{\gamma}{\equiv} a_5 = a_5 \overset{\gamma}{\equiv} a_1 \tag{5.4}$$
$$\dots$$

hold in all ortholattices (OLs) *with strong sets of states. An* OL *to which these equations are added is a variety smaller than* OML.

We shall call these equations n-Go (3-Go, 4-Go, *etc.*). *We also denote by n*GO (3GO, 4GO, *etc.*) *the* OL *variety determined by n*-Go *and call it the class of n*GO **lattices**.

Proof. See Ref. [105, p. 771, Th. 19]. ☐

Lemma 5.1.4. *[105, p. 771, Lemma 20] Any n*GO *is an* $(n-1)$GO, $n = 4, 5, 6, \dots$

Proof. Substitute a_1 for a_2 in equation n-Go. ☐

The converse of Lemma (5.1.4) does not hold. Indeed, the *wagon wheel* OMLs Gn, $n = 3, 4, 5, \dots$, are related to the n-Go equations in the sense that Gn violates n-Go but (for $n \geq 4$) not $(n-1)$-Go. In Fig. 5.1 we show examples G3[1] and G4;[2] the obvious way (according to the general scheme described in [26]).

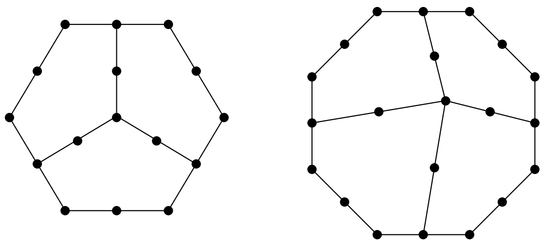


Figure 5.1: (a) Greechie diagram for OML G3; (b) Greechie diagram for OML G4.

Megill and Pavičić [76] explored many properties and consequences of the n-Go equations. The theorems below, whose proofs we omit and can be found in the cited reference, summarize some of the results that work.

[1] `123,147,258,369,7CE,8AC,8BD,9DG,EFG.` is an MMP encoding for G3 (Fig. 5.1(a)).
[2] `123,345,567,789,9AB,BCD,DEF,FG1,GHL,4IL,8JL,CKL.` is an MMP encoding for G4 (Fig. 5.1(b)).

Theorem 5.1.5. *An* OL *in which any of the following equations holds is an n*GO *and vice versa.*

$$a_1 \overset{\gamma}{\equiv} a_n = (a_1 \equiv a_2) \wedge (a_2 \equiv a_3) \wedge \cdots \wedge (a_{n-1} \equiv a_n) \tag{5.5}$$

$$a_1 \overset{\gamma}{\equiv} a_n \le a_1 \to a_n, \tag{5.6}$$

$$(a_1 \overset{\gamma}{\equiv} a_n) \wedge (a_1 \vee a_2 \vee \cdots \vee a_n) = a_1 \wedge a_2 \wedge \cdots \wedge a_n \tag{5.7}$$

Theorem 5.1.6. *In any n*GO*, $n = 3, 4, 5, \ldots$, the following relations hold.*

$$a_1 \overset{\gamma}{\equiv} a_n \le a_j \to a_k, \qquad 1 \le j \le n,\ 1 \le k \le n \tag{5.8}$$

The n-Go equations can be equivalently expressed as inferences involving $2n$ variables, as the following theorem shows. In this form they can be useful for certain kinds of proofs.

Theorem 5.1.7. *Any* OML *in which*

$$a_1 \perp b_1 \perp a_2 \perp b_2 \perp \ldots \perp a_n \perp b_n \perp a_1 \qquad \Rightarrow$$
$$(a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \cdots \wedge (a_n \vee b_n) \le b_1 \vee a_2 \tag{5.9}$$

*holds is an n*GO *and vice versa.*

Finally, the following theorem shows a transitive-like property that can be derived from the Godowski equations.

Theorem 5.1.8. *The following equation holds in n*GO*, where $i, j \ge 1$ and $n = \max(i, j, 3)$.*

$$(a_1 \overset{\gamma}{\equiv} a_i) \wedge (a_i \overset{\gamma}{\equiv} a_j) \le a_1 \overset{\gamma}{\equiv} a_j \tag{5.10}$$

While the wagon wheel OMLs characterize nGO laws in an elegant way, they are not the smallest OMLs that are not nGOs. Smaller OMLs exist that can be used to distinguish $(n+1)$-Go from n-Go, which can improve computational efficiency [105, p. 772]. For example, the Peterson OML, G4s,[3] Fig. 5.2(a), is the smallest that violates 4-Go but not 3-Go; it has 32 nodes vs. 44 nodes in the wagon wheel G4 in Fig. 5.1(b). Lattice G5s,[4] Fig. 5.2(b), with 42 nodes (vs. 54 nodes in G5), is the smallest that violates 5-Go but not 4-Go. OML G6s2,[5]

[3] `123,345,567,789,9AB,BC1,2E8,4FA,6DC,DEF.` is an MMP encoding for OML G4s (Fig. 5.2(a)).

[4] `FGL,EHL,BCK,ADJ,9AF,8BE,79K,68J,67I,5DH,4CG,35K,24J,1IL,123.` is an MMP encoding for OML G5s (Fig. 5.2(b)).

[5] `FGI,EHJ,9AF,8BE,7CH,6DG,3BD,357,2AC,246,189,145,1KL,GHL.` is an MMP encoding for OML G6s2 (Fig. 5.2(c))

Fig. 5.2(c) is one of three smallest that violates 6-Go but not 5-Go, with 44 nodes (vs. 64 nodes) in G6. Lattice G7s1,[6] Fig. 5.2(d), is one of several smallest we obtained to violate 7-Go but not 6-Go. They both have 50 nodes, respectively (vs. 74 nodes in G7).

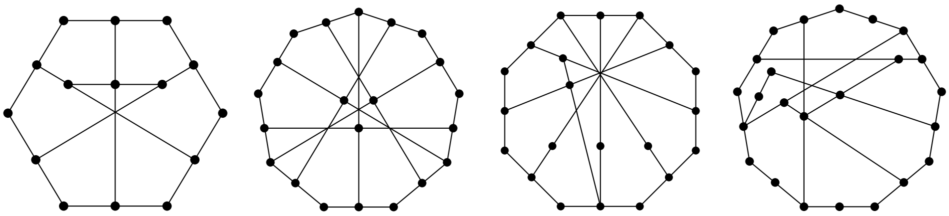


Figure 5.2: (a) OML G4s; (b) OML G5s; (c) OML G6s; (d) OML G7s. [105, p. 773, Fig. 8]

Whether there is a pattern in the OMLs G4s through G7s is unknown. While their Greechie diagrams reveal no obvious pattern, the appearance of a Greechie diagram is highly dependent on how it is drawn. For example, the "wagon wheel" pattern in OML G3 [Fig. 5.1 (p. 79)] is apparent only when it is drawn with a loop of order 6. Fig. 5.3 compares the three ways of drawing it, from a loop of order 5 to the maximal loop of order 7.[7]

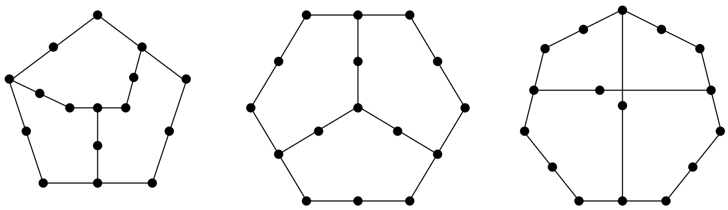


Figure 5.3: Three ways of drawing OML G3, only one of which reveals the "wagon wheel" pattern.

5.2 Mayet-Godowski equations

In 1985, René Mayet [64] described an equational variety of lattices, which he called OM_S^*, that included all Hilbert lattices and were included in the nGO varieties (found by Godowski) that we described in the previous section. In 1986, Mayet [65] displayed several examples

[6] `IKO,GHN,FJL,EJM,BDF,9AE,8CI,5CD,56H,4BK,47G,2AK,236,189,137,HJO.` is the MMP encoding for OML G7s1 (Fig. 5.2(d)).

[7] These drawings were assisted with the `loop.c` program [Sec. A.7 (p. 154)] applied to the Greechie diagram with MMP encoding `123,345,567,789,9AB,BC1,2DG,6EG,AFG.`.

of equations that held in this new variety. However, Megill and Pavičić [76] showed that all of Mayet's equational examples can be derived in nGO for some n. Thus for some years it remained unclear whether Mayet's variety was strictly contained in the nGOs. Later, though, Megill and Pavičić [81] exhibited an equation that holds in his variety (and thus in all Hilbert lattices) but cannot be derived in any nGO, thus showing that Mayet's variety, which we will call MGO, is indeed strictly contained in all nGOs (Theorem 5.2.7).

In this section, we will first review this work, then we will present some additional equations that have not yet been published.

We will describe a general family of equations that hold in all Hilbert lattices and contains the new equation, and we will define a simplified notation for representing these equations. We call the equations in this family Mayet-Godowski equations and, in Theorem 5.2.4, prove that they hold in all Hilbert lattices.[8]

Definition 5.2.1. *A* Mayet-Godowski equation *(*MGE*) is an equality with $n \geq 2$ conjuncts on each side:*

$$t_1 \wedge \cdots \wedge t_n = u_1 \wedge \cdots \wedge u_n \tag{5.11}$$

where each conjunct t_i (or u_1) is a term consisting of either a variable or a disjunction of two or more distinct variables:

$$t_i = a_{i,1} \vee \cdots \vee a_{i,p_i} \qquad \text{i.e. } p_i \text{ disjuncts} \tag{5.12}$$

$$u_i = b_{i,1} \vee \cdots \vee b_{i,q_i} \qquad \text{i.e. } q_i \text{ disjuncts} \tag{5.13}$$

and where the following conditions are imposed on the set of variables in the equation:

1. *All variables in a given term t_i or u_i are mutually orthogonal.*

2. *Each variable occurs the same number of times on each side of the equality.*

We will call a lattice in which all MGEs hold an MGO; i.e., MGO is the class (equational variety) of all lattices in which all MGEs hold.

Lemma 5.2.2. *In any* OL*, the following orthogonality condition holds:*

$$a \perp b \quad \& \quad a \perp c \quad \Rightarrow \quad a \perp (b \vee c). \tag{5.14}$$

[8] A family of equations equivalent to the family MGE, with a different presentation, was given by Mayet as $E(Y_2)$ on p. 183 of [65].

Proof. See Ref. [105, p. 773, Lemma 26]. ☐

Lemma 5.2.3. *If $a_1,\ldots a_n$ are mutually orthogonal, then for any state m,*

$$m(a_1)+\cdots+m(a_n)=m(a_1\vee\cdots\vee a_n). \tag{5.15}$$

Proof. See Ref. [105, p. 773, Lemma 27]. ☐

Theorem 5.2.4. *A Mayet-Godowski equation holds in any ortholattice L admitting a strong set of states and thus, in particular, in any Hilbert lattice.*

Proof. See Ref. [105, p. 773, Th. 28]. ☐

In order to represent MGEs efficiently, we introduce a special notation for them. Consider the following MGE (which will be of interest to us later):

$$\begin{aligned}a\perp b\ \&\ a\perp c\ \&\ b\perp c\ \&\ d\perp e\ \&\ f\perp g\ \&\ h\perp j\ \&\ g\perp b\ \&\\ e\perp c\ \&\ j\perp a\ \&\ h\perp f\ \&\ h\perp d\ \&\ f\perp d\ \Rightarrow\\ (a\vee b\vee c)\wedge(d\vee e)\wedge(f\vee g)\wedge(h\vee j)=\\ (g\vee b)\wedge(e\vee c)\wedge(j\vee a)\wedge(h\vee f\vee d).\end{aligned} \tag{5.16}$$

Following the proof of Theorem 5.2.4, this equation arises from the following equality involving states:

$$\begin{aligned}m(a\vee b\vee c)+m(d\vee e)+m(f\vee g)+m(h\vee j)=\\ m(g\vee b)+m(e\vee c)+m(j\vee a)+m(h\vee f\vee d).\end{aligned} \tag{5.17}$$

A **condensed state equation** is an abbreviated representation of this equality, wherein we represent join by juxtaposition and remove all mentions of the state function, leaving only its arguments. Thus the condensed state equation representing Eq. (5.17), and thus Eq. (5.16), is:

$$abc+de+fg+hj\quad=\quad gb+ec+ja+hfd. \tag{5.18}$$

Another example of an MGE shows that repeated or **degenerate** terms may be needed in the condensed state equation in order to balance the number of variable occurrences on each side, in order to satisfy Condition 2 of Def. 5.2.1:

$$\begin{aligned} ab+cde+fg+fg+hjk+lk+mn+pe\ = \\ gk+gk+db+fe+fe+nlc+pja+mh \end{aligned} \qquad (5.19)$$

Theorem 5.2.5. *The family of all Mayet-Godowski equations includes, in particular, the Godowski equations* [Eqs. (5.2), (5.3),...]*; in other words, the class* MGO *is included in n*GO *for all n.*

Proof. See Ref. [105, p. 776, Th. 29] ☐

While every MGE holds in a Hilbert lattice, many of them are derivable from the equations *n*-Go and others trivially hold in all OMLs. We will call an MGE "interesting" if it does not hold in all *n*GOs. To find such MGEs, we seek OMLs that are *n*GOs for all *n* but have no strong set of states. Once we find such an OML, it is possible to deduce an MGE that it will violate.

The search for such OMLs was done with the assistance of several computer programs written by Brendan McKay and Norman Megill. These programs are described in Appendix A (p. 145). An isomorph-free, exhaustive list of finite OMLs with certain characteristics was generated. The ones admitting no strong set of states were identified (by using the simplex linear programming algorithm, implemented in our program `states.c`, to show that the constraints imposed by a strong set of states resulted in an infeasible solution). Among these, the ones violating some *n*-Go were discarded, leaving only the OMLs of interest. (To identify an OML of interest, a special dynamic programming algorithm, described in [81], was used in our program `latticego.c`. This algorithm was crucial for the results in this section, providing a proof that the OML "definitely" violated no *n*-Go for all *n* less than infinity, rather than just "probably" as would be obtained by testing up to some large *n* with a standard lattice-checking program.) Finally, an MGE was "read off" of the OML, using a variation of a technique described by Mayet [65] for producing an equation that is violated by a lattice admitting no strong set of states.

In Fig. 5.4 (p. 87), we show examples of such OMLs found by these programs. Eq. (5.16) was deduced from OML MG1[9] in the figure, and it provides the answer (Theorem 5.2.7 below) to the problem posed at the beginning of this section. In order to show how we constructed Eq. (5.16), we will show the details of the proof that OML MG1 admits no strong set of states. That proof will provide us with an algorithm for stating an equation that fails in OML MG1 but holds in all OMLs admitting a strong set of states.

[9] `ABC,9BI,8CJ,7AH,6DE,5DF,4DG,358,269,147,123.` is an MMP encoding for OML MG1 [Fig. 5.4(a)].

Theorem 5.2.6. *The* OML MG1 *does not admit a strong set of states.*

Proof. [105, p. 773, Lemma 28] Referring to Fig. 5.4 (p. 87), suppose that m is a state such that $m(v) = 1$. Since the state values of the atoms in a block sum to 1, $m(a_1) = m(a_2) = m(a_3) = 0$. Thus $m(b_1)+m(c_1) = m(b_2)+m(c_2) = m(b_3)+m(c_3) = 1$. Since $m(b_1)+m(b_2)+m(b_3) \le 1$, it follows that $m(c_1)+m(c_2)+m(c_3) \ge 2$. Since $m(d_1)+m(d_2)+m(d_3) = 1$, we have $[m(c_1)+m(d_1)]+[m(c_2)+m(d_2)]+[m(c_3)+m(d_3)] \ge 3$. Since $m(c_1)+m(d_1) \le 1$, $m(c_2)+m(d_2) \le 1$, and $m(c_3)+m(d_3) \le 1$, we must have $m(c_3)+m(d_3) = 1$. Hence $m(u)$=0, since u is on the same block as c_3 and d_3. So, $m(u') = 1$. To summarize, we have shown that for any m, $m(v) = 1$ implies $m(u') = 1$. If MG1 admitted a strong set of states, we would conclude that $v \le u'$, which is a contradiction since v and u' are incomparable. □

In the above proof, we made use of several specific conditions that hold for the atoms and blocks in that OML. That proof was actually carefully constructed so as to minimize the need for these conditions. For example, we used $m(b_1)+m(b_2)+m(b_3) \le 1$ even though the stronger $m(b_1)+m(b_2)+m(b_3) = 1$ holds, because the strength of the latter was not required. The complete set of such conditions that the proof used are the following facts:

- $v \perp a_i$, $i = 1,2,3$;
- $d_i \perp c_i$, $i = 1,2$;
- The atoms in each of the triples $\{a_i,b_i,c_i\}$ ($i = 1,2,3$), and $\{d_1,d_2,d_3\}$ are mutually orthogonal and their disjunction is 1 (i.e. the sum of their state values is 1).
- The atoms in each of the triples $\{b_1,b_2,b_3\}$ and $\{c_3,u,d_3\}$ are mutually orthogonal and the sum of their state values is ≤ 1 (the sum is actually equal to 1, but we used only ≤ 1 for the proof).

If the elements of any OML L satisfy these facts, then we can prove (with a proof essentially identical to that of Theorem 5.2.6, using the above facts as hypotheses in place of the atom and block conditions in OML MG1) that for any state m on L, $m(v) = 1$ implies $m(u') = 1$. Then, if L admits a strong set of states, we also have $v \le u'$.

We can construct an equation that expresses this result as follows. We use the orthogonality conditions from the above list of fact as hypotheses, and we incorporate each "disjunction is 1" condition as a conjunct on the left-hand side. We will denote the set of all orthogonality conditions in the above list of facts by Ω. We can ignore the conditions "the sum of their state values is ≤ 1" from the above list of facts, because that happens automatically due to the mutual orthogonality of those elements. This procedure then leads to the equation,

$$\Omega \Rightarrow \quad v \wedge (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge (a_3 \vee b_3 \vee c_3) \wedge (d_1 \vee d_2 \vee d_3) \le u' \qquad (5.20)$$

This equation holds in all OMLs with a strong set of states but fails in lattice MG1.

The condensed state equation Eq. (5.18) was obtained using the following mechanical procedure. We consider only variables corresponding to the atoms used by the proof (i.e. the labeled atoms in Fig. 5.4) and only the blocks whose orthogonality conditions were used as hypotheses for the proof. We ignore all variables whose state value is shown to be equal to 1 or 0 by the proof, and we ignore all blocks in which only one variable remains as a result. For the left-hand side, we consider all the remaining blocks that have "disjunction is 1" in the assumptions listed above. We juxtapose the (non-ignored) variables in each block to become a term, and we connect the terms with $+$. For the right-hand side, we do the same for the remaining blocks that do not have "disjunction is 1" in the assumptions listed above. Thus we obtain:

$$b_1c_1 + b_2c_2 + b_3c_3 + d_1d_2d_3 \quad = \quad c_1d_1 + c_2d_2 + c_3d_3 + b_1b_2b_3 \qquad (5.21)$$

After renaming variables and rearranging terms, this is Eq. (5.18), which corresponds to the MGE Eq. (5.16) and which can be verified to fail in lattice MG1.

This mechanical procedure is simple and practical to automate—the simplex algorithm used to find states lets us determine which blocks must have a disjunction equal to 1—but it is not guaranteed to be successful in all cases: in particular, it will not work when the condensed state equation has degenerate terms, as in Eq. (5.19) above. However, such cases are easily identified by counting the variable occurrences on each side, and we can add duplicate terms to make the counts balance in the case of a degeneracy. This balancing ensures that the corresponding equation is an MGE and therefore holds in all Hilbert lattices.

Having constructed Eq. (5.16), which holds in all Hilbert lattices but fails in lattice MG1 (in which all n-Go equations hold), we now state the main result of this section.

Theorem 5.2.7. *The class* MGO *is properly included in all n*GO*s, i.e., not all* MGE *equations can be deduced from the equations n*-Go*.*

Proof. See Ref. [105, p. 773, Th. 31] ☐

In particular, Eq. (5.16) therefore provides an an example of a new Hilbert lattice equation that is independent from all Godowski equations.

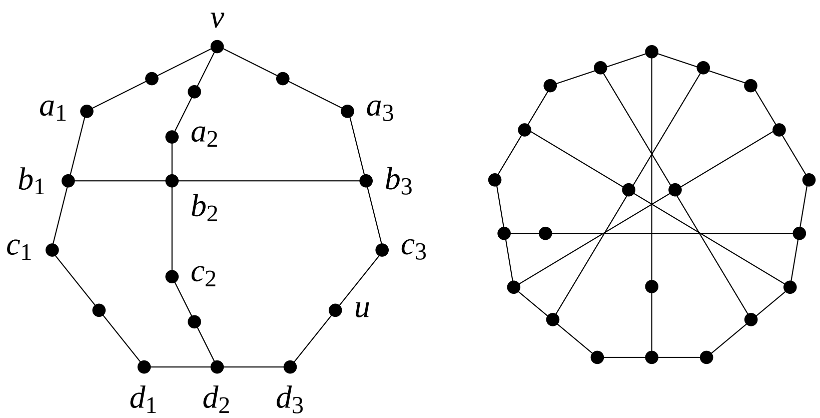


Figure 5.4: OMLs that admit no strong sets of states but which are *n*GOs for all *n*. (a) OML MG1; (b) OML MG5s.

Having 9 variables and 12 hypotheses, Eq. (5.16) can be somewhat awkward to work with directly. It is possible to derive from it a simpler equation through the use of substitutions that Mayet calls generators [65, p. 189]. If, in Eq. (5.16), we substitute (simultaneously) c' for a, $c\wedge b$ for b, $(c\to b)'$ for c, $(a\to b)'$ for d, $(c\to b)\wedge(a\to b)$ for e, $b\wedge a$ for f, b' for g, a' for h, and $a\wedge c$ for j, all of the hypotheses are satisfied (in any OML) and the conclusion evaluates to:

$$((a\to b)\to(c\to b))\wedge(a\to c)\wedge(b\to a) \quad\leq\quad c\to a \tag{5.22}$$

where we also dropped all but one conjunct on the right-hand-side. While such a procedure can sometimes weaken an MGE, it can be verified that Eq. (5.22) still fails in OML MG1[10] of Fig. 5.4 as desired, thus providing us with a Hilbert lattice equation that is convenient to work with but is still independent from all Godowski equations. For example, Eq. (5.22) can be used in place of Eq. (5.16) to provide a simpler proof of Theorem 5.2.7.

Eq. (5.19) (p. 84) was deduced from the OML MG5s[11] in Fig. 5.4, and it provides us with another new Hilbert lattice equation that is independent from all *n*-Gos. A comparison to OML G5s in Fig. 5.2 (p. 81) illustrates how the addition of an atom can affect the behaviour of a lattice.

The OMLs of Ref. [105, p. 780, Fig. 10], which we will not repeat here, provide further examples that admit no strong sets of states but are *n*GOs for all *n*. The following MGEs

[10] `ABC,9BI,8CJ,7AH,6DE,5DF,4DG,358,269,147,123.` is an MMP encoding for MG1 [Fig. 5.4(a)].

[11] `HKM,FGL,EGJ,DFI,BCH,ABI,9CJ,67D,58E,48K,37K,26J,24A,15I,139.` is an MMP encoding for MG5s [Fig. 5.4(b)].

(represented with condensed state equations) can be deduced from them, respectively:

$$abc + de + fg + hj + kl = eb + dh + faj + lc + kg \tag{5.23}$$

$$ab + cd + ef + ghj + kl + kl = kd + bl + jl + fk + ha + gec \tag{5.24}$$

$$abc + def + gh + jk + lmn + pqr = fn + rc + dkb + gma + qeh + plj. \tag{5.25}$$

Using generators, the following examples of simpler Hilbert lattice equations can be derived from these MGEs, again respectively:

$$(d\to(a\to b)) \wedge ((a\to c)\to d) \wedge (b\to c) \wedge (c\to a) \le b\to a \tag{5.26}$$

$$(d\to(c \wedge (a\to b)) \wedge ((b\to a)\to d) \wedge (c\to a) \wedge (b\to d) \le a\to c \tag{5.27}$$

$$\begin{aligned}((d\to a)\to(b\to c)') \wedge ((c\to d)\to(a\to b)') \wedge ((b\to a)'\to(d\to c))\\ \wedge ((a\to d)'\to(c\to b)) \le (d\to c)\to(b\to a)'.\end{aligned} \tag{5.28}$$

Each of these simpler equations, while possibly weaker than the MGEs they were derived from, still fail in their corresponding OMLs, thus providing us with additional new Hilbert lattice equations that are independent from all *n*GOs.

While the complete picture of interdependence of the three lattice families we have presented (*n*OA, *n*GO, and MGO) is not fully understood, some results can be established. We have already shown that every MGO is an *n*GO for all *n*, and moreover that the inclusion is proper (Theorem 5.2.7). We can also prove the following:

Theorem 5.2.8. *There are* MGO*s* (*and therefore n*GO*s*) *that are not* 3OA*s and thus not n*OA*s for any n.*

Proof. See Ref. [105, p. 780, Th. 32]. ☐

Theorem 5.2.9. *There are n*OA*s for* $n = 3, 4, 5, 6$ *that are not* 3GO*s and thus not n*GO*s for any n nor* MGO*s.*

Proof. See Ref. [105, p. 780, Th. 33]; specifically, Ref. [105, p. 781, Fig. 11] shows an OML which is a 6OA but not a 3GO. ☐

Whether Theorem 5.2.9 holds for all *n*OAs remains an open problem. However, our observation is that the smallest OMLs in which the *n*OA law passes but the $(n+1)$OA law fails grow in size with increasing *n*, as indicated by the OMLs used to prove Theorem 4.3.3. Compared to them, the OML of Ref. [105, p. 781, Fig. 11] is "small," leading us to conjecture that it is an

nOA for all n. If this conjecture is true, it would show that no n-Go equation can be derived (in an OML) from the nOA laws.

5.2.1 Additional MGE equations

In this section, we summarize additional MGE equations found during this project. Except for #1 and #18 in the tables below, whose Greechie diagrams are shown in Fig. 5.4 above (p. 87) and which we include for completeness, these do not appear in the literature.

We scanned all lattices with 3 atoms per block, up to 15 blocks, and found 883 that satisfied all n-Go equations [Th. 5.1.3 (p. 79)], using the program `latticego.c`, while also not admitting a strong set of states, using the program `states.c`. Using technique in the proof of Th. 5.2.6 (p. 85), each of these was used to derive an equation that holds in all lattices admitting a strong set of states (and thus in all $\mathscr{C}(H)$s) but fails in the given lattice.

We performed this detailed analysis on a sample of 19 lattices for which we could derive a new $\mathscr{C}(H)$ equation. We summarize these results in the following 4 tables.

Table 5.1 shows the lattice as a Greechie diagram encoded in MMP format [Def. 2.5.6 (p. 23)].

Table 5.2 shows the condensed state equation derived from the lattice. As described above, the state equation is a shorthand to express a $\mathscr{C}(H)$ equation, although typically such an equation has many variables and orthogonality hypotheses and is unwieldy to work with. Degenerate condensed state equations are marked with $*$ [see definition of degenerate above Th. 5.2.5 (p. 84)].

Table 5.3 gives an equation derived from the condensed state equation, using the "generator" method described above. While it is not necessarily as strong as the equation corresponding to the condensed state equation, it is still strong enough to fail in the corresponding lattice (and thus serves as an "interesting" new $\mathscr{C}(H)$ equation).

Finally, Table 5.5 provides a simplified inference from the equation of Table 5.3, obtained by changing the equality to an inequality ($<$) and empirically discarding conjuncts on the right-hand side so that the equation still failed in the corresponding lattice. This final equation, even though it is not necessarily as strong as the one corresponding to the condensed state equation or even the equation of Table 5.3 that it was derived from, is the most convenient to work with when exploring new $\mathscr{C}(H)$ equational properties.

Table 5.1: MMP encodings for the Greechie diagrams used to derive the MGE equations of Tables 5.2, 5.3, and 5.5

Eq. #	MMP encoding for Greechie diagram
1	ABC,9BI,8CJ,7AH,6DE,5DF,4DG,358,269,147,123.
2	DEF,ACJ,9BI,8EM,7DL,6BH,5CK,468,34G,257,13A,129,CFI.
3	9AI,8AH,7CE,78F,6BD,69G,5EG,4DF,3CI,2BH,145,123.
4	DFG,BCK,AEI,9DH,8EJ,6AF,5CF,37B,346,279,148,125,DLM,BEM.
5	DGJ,CFH,BEI,89A,7AD,68B,59C,347,2FG,246,1EG,135.
6	FGL,EHM,DIK,AHJ,9IJ,7CJ,67F,56B,48C,3AB,34D,28G,15E,129.
7	BCD,9EF,9AD,8BJ,7AI,78L,6GH,6CK,5EK,4FL,3HI,2GJ,134,125.
8	BEF,BCD,ADL,9CK,8AI,79J,68G,67H,5JL,4IK,3FH,2EG,135,124.
9	JKL,HIJ,EGK,DFK,ABG,8CF,79B,6AC,45E,37D,34I,256,189,12H.
10	JKL,HIJ,EFK,DGK,9CG,78B,6AC,5AB,46F,35E,289,23H,17D,14I.
11	JKL,HIJ,EFK,DGK,89B,7AC,6CF,5BG,469,35A,34H,28E,17D,12I.
12	9FH,9AB,8EG,8CD,7CL,7AI,6DJ,6BK,5FJ,4GK,3HL,2EI,134,125.
13	FHL,EGM,DIJ,CIK,ABH,8GH,7FI,458,36B,35D,29A,24C,179,16E.
14	FHI,CGL,BDJ,AEK,67C,59E,48D,3BH,379,2AH,268,1FG,145.
15	JKL,HIJ,EGK,DFK,9CG,8BF,7AC,6AB,45E,37D,34I,256,189,12H.
16	JKL,HIJ,EGK,DFK,9AG,8BE,7CF,6BC,45D,358,269,24I,17A,13H.
17	GHJ,FIK,EGI,CEM,BDL,7AC,69B,58F,4AH,489,36I,237,1DH,125.
18	HKM,FGL,EGJ,DFI,BCH,ABI,9CJ,67D,58E,48K,37K,26J,24A,15I,139.
19	FHJ,EGK,DIL,9AI,6BE,68A,5CF,579,4BC,38H,27G,14D,123.

Table 5.2: Condensed state equations derived from Greechie diagrams of Table 5.1 ($*$ = degenerate).

Eq. #	Condensed state equation
1	$abc+de+fg+hj=gb+ec+ja+hfd$
2	$ab+cd+efg+hj+kl=gb+ja+fd+le+khc$
3	$abc+de+fg+hj+kl=eb+dh+faj+lc+kg$
4	$ab+cd+ef+ghj+kl=kd+bl+jl+fk+ha+gec\,*$
5	$abc+def+gh+jk+lm=fk+cm+be+jha+lgd$
6	$ab+cd+ef+gh+jk+lmn=fk+nb+hd+me+ga+ljc$
7	$ab+cd+efg+hj+kl+mnp=cfb+aj+el+pg+nd+mkh$
8	$ab+cde+fgh+jk+lm+np=ad+gm+kcl+jf+phe+nb$
9	$ab+cd+efg+hjk+lm+np=jgb+kd+ma+lec+ph+nf$
10	$ab+cd+efg+hjk+lm+np=kd+jg+phb+ma+lf+nec$
11	$ab+cd+efg+hjk+lm+np=kb+gd+mf+lj+pea+nhc$
12	$abc+def+gh+jk+lm+np=agl+dnj+em+bp+fh+ck$
13	$ab+cdb+ef+ghj+kl+mn=eb+nd+jf+lf+khc+gma\,*$
14	$abc+def+ghj+kl+mn+pq=cq+lj+nf+kbe+mah+pgd$
15	$ab+cd+efg+hj+klm+npq=qgb+pd+mf+jla+hec+nk$
16	$ab+cd+efg+hjk+lm+npq=qb+kfa+pgd+mjc+le+nh$
17	$ab+cde+fgh+jk+lm+npk=ak+hb+ep+jgd+mcb+nlf\,*$
18	$ab+cde+fg+hjk+lk+mn+pe=gk+db+fe+nlc+pja+mh\,*$
19	$abc+def+gh+jk+lmn+pqr=fn+rc+dkb+gma+qeh+plj$

Table 5.3: MGE equations derived from condensed state equations of Table 5.2. (Continued in Table 5.4.)

Eq. #	MGE equation
1	$((a\to b)\to(c\to b))\wedge((a\to c)\wedge(b\to a))$
	$=((c\to b)\to(a\to b))\wedge((b\to c)\wedge(c\to a))$
2	$((d\to(c\to b))\wedge((a\to b)\to d))\wedge((b\to a)\wedge(a\to c))$
	$=((d\to(a\to b))\wedge((c\to b)\to d))\wedge((b\to c)\wedge(c\to a))$
3	$((d\to(a\to b))\wedge((a\to c)\to d))\wedge((b\to c)\wedge(c\to a))$
	$=((d\to(a\to c))\wedge((a\to b)\to d))\wedge((c\to b)\wedge(b\to a))$
4	$((d\to(c\wedge(a\to b)))\wedge((b\to a)\to d))\wedge((c\to a)\wedge(b\to d))$
	$=(((c\to(d\wedge(a\to b)))\wedge((a\to b)\to(d\wedge c)))$
	$\wedge((d\to(b\to a))\wedge(d\to b)))\wedge(a\to c)$
5	$(b\to((a\to c)\to((b\to a)\to c)))\wedge((a\to b)\wedge(c\to a))$
	$=((((a\to c)\to((b\to a)\to c))\to b)$
	$\wedge(((b\to a)\to c)\to(a\to c)))\wedge(c\to(b\to a))$
6	$((d\to e)\wedge((e\to(a\to b))\wedge((a\to c)\to d)))\wedge((b\to c)\wedge(c\to a))$
	$=((e\to d)\wedge((d\to(a\to c))\wedge((a\to b)\to e)))\wedge((c\to b)\wedge(b\to a))$
7	$((b\to((a\to c)\to((b\to d)\to c)))\wedge((a\to d)\wedge(c\to a)))\wedge(d\to b)$
	$=((((a\to c)\to((b\to d)\to c))\to b)$
	$\wedge((d\to a)\wedge(((b\to d)\to c)\to(a\to c))))\wedge(c\to(b\to d))$
8	$((((a\to b)\to d)\to((a\to c)\to d))\wedge(d\to(a\to b)))\wedge((b\to c)\wedge(c\to a))$
	$=((((a\to c)\to d)\to((a\to b)\to d))$
	$\wedge(d\to(a\to c)))\wedge((b\to a)\wedge(c\to b))$
9	$((((a\to b)\to d)\to(c\to a))\wedge((a\to c)\to d))\wedge((d\to(a\to b))\wedge(b\to c))$
	$=(((c\to a)\to((a\to b)\to d))\wedge(d\to(a\to c)))\wedge((c\to b)\wedge(b\to a))$
10	$((d\to((a\to c)\to((b\to a)\to c)))\wedge(b\to d))\wedge((a\to b)\wedge(c\to a))$
	$=((((a\to c)\to((b\to a)\to c))\to d)\wedge(d\to b))$
	$\wedge((c\to(b\to a))\wedge(((b\to a)\to c)\to(a\to c)))$

Table 5.4: (Continuation of Table 5.3.)

Eq. #	MGE equation
11	$((b\to((a\to c)\to d))\wedge(d\to((b\to a)\to c)))\wedge((a\to b)\wedge(c\to a))$
	$=((((a\to c)\to d)\to b)\wedge(((b\to a)\to c)\to d))$
	$\wedge((c\to(b\to a))\wedge(d\to(a\to c)))$
12	$(((a\to b)\to(c\to b))\wedge((c\to d)\to(a\to d)))\wedge((b\to a)\wedge(d\to c))$
	$=(((c\to b)\to(a\to b))\wedge((a\to d)\to(c\to d)))\wedge((d\to a)\wedge(b\to c))$
13	$((((a\to b)\wedge c)\to_2((a\to c)\wedge d))\wedge(((a\to c)\wedge d)\to_2(a\wedge b)))$
	$\wedge(((c\to a)\to((a\to b)\to c))\wedge(c\to(a\to b)))$
	$=((((a\to c)\wedge d)\to_2((a\to b)\wedge c))\wedge((a\wedge b)\to_2((a\to c)\wedge d)))$
	$\wedge((((a\to b)\to c)\to(c\to a))\wedge((a\to c)\to d))$
14	$(((c\to a)\to(b\to a))\wedge((b\to c)\to(a\to c)))\wedge((a\to b)\to(c\to b))$
	$=(((b\to a)\to(c\to a))\wedge((a\to c)\to(b\to c)))\wedge((c\to b)\to(a\to b))$
15	$((c\to b)\to((a\to c)\to((b\to a)\to(c\to a))))\wedge((a\to b)\wedge(b\to c))$
	$=(((a\to c)\to((b\to a)\to(c\to a)))\to(c\to b))$
	$\wedge((((b\to a)\to(c\to a))\to(a\to c))\wedge((c\to a)\to(b\to a)))$
16	$((c\to b)\to((b\to a)\to(c\to a)))\wedge(((a\to c)\to(b\to c))\wedge(a\to b))$
	$=(((b\to a)\to(c\to a))\to(c\to b))$
	$\wedge(((b\to c)\to(a\to c))\wedge((c\to a)\to(b\to a)))$
17	$((((b\to a)\to(((a\to c)\to(b\to c))'\wedge(c\to b))')$
	$\wedge((c\to b)\to((a\to c)\to(b\to c))'))\wedge((((a\to c)\to(b\to c))'$
	$\to(c\to b))\wedge((c\to a)'\to(b\to a))))\wedge(a\to b)$
	$=((((a\to c)\to(b\to c))'\wedge(c\to b))'\to(b\to a))$
	$\wedge(((b\to a)\to(c\to a)')\wedge((b\to c)\to(a\to c)))$
18	$(((a\wedge(c\to b))\to_2(b\wedge d))\wedge(c\to(a\to(c\to b))))$
	$\wedge(((a\to b)\to(b\to c))\wedge(((c\to b)\to a)\wedge(b\to a)))$
	$=(((b\wedge d)\to_2(a\wedge(c\to b)))\wedge((a\to(c\to b))\to c))$
	$\wedge(((b\to c)\to(a\to b))\wedge(b\to d))$
19	$(((d\to a)\to(b\to c)')\wedge((c\to d)\to(a\to b)'))$
	$\wedge(((b\to a)'\to(d\to c))\wedge((a\to d)'\to(c\to b)))$
	$=(((b\to c)'\to(d\to a))\wedge((a\to b)'\to(c\to d)))$
	$\wedge(((d\to c)\to(b\to a)')\wedge((c\to b)\to(a\to d)'))$

Table 5.5: Simplified MGE equations derived from Table 5.3.

Eq. #	Simplified MGE equation
1	$((a\to b)\to(c\to b))\wedge((a\to c)\wedge(b\to a))\le c\to a$
2	$((d\to(c\to b))\wedge((a\to b)\to d))\wedge((b\to a)\wedge(a\to c))\le c\to a$
3	$((d\to(a\to b))\wedge((a\to c)\to d))\wedge((b\to c)\wedge(c\to a))\le b\to a$
4	$((d\to(c\wedge(a\to b)))\wedge((b\to a)\to d))\wedge((c\to a)\wedge(b\to d))\le a\to c$
5	$(b\to((a\to c)\to((b\to a)\to c)))\wedge((a\to b)\wedge(c\to a))\le c\to(b\to a)$
6	$(d\to e)\wedge(((e\to(a\to b))\wedge((a\to c)\to d))\wedge((b\to c)\wedge(c\to a)))\le b\to a$
7	$((b\to((a\to c)\to((b\to d)\to c)))\wedge((a\to d)\wedge(c\to a)))\wedge(d\to b)$ $\le d\to a$
8	$((((a\to b)\to d)\to((a\to c)\to d))\wedge(d\to(a\to b)))\wedge((b\to c)\wedge(c\to a))$ $\le b\to a$
9	$((((a\to b)\to d)\to(c\to a))\wedge((a\to c)\to d))\wedge((d\to(a\to b))\wedge(b\to c))$ $\le c\to b$
10	$((d\to((a\to c)\to((b\to a)\to c)))\wedge(b\to d))\wedge((a\to b)\wedge(c\to a))\le d\to b$
11	$((b\to((a\to c)\to d))\wedge(d\to((b\to a)\to c)))\wedge((a\to b)\wedge(c\to a))$ $\le c\to(b\to a)$
12	$(((a\to b)\to(c\to b))\wedge((c\to d)\to(a\to d)))\wedge((b\to a)\wedge(d\to c))\le d\to a$
13	$((((a\to b)\wedge c)\to_2((a\to c)\wedge d))\wedge(((a\to c)\wedge d)\to_2(a\wedge b)))$ $\wedge(((c\to a)\to((a\to b)\to c))\wedge(c\to(a\to b)))\le(a\to c)\to d$
14	$(((c\to a)\to(b\to a))\wedge((b\to c)\to(a\to c)))\wedge((a\to b)\to(c\to b))$ $\le(b\to a)\to(c\to a)$
15	$((c\to b)\to((a\to c)\to((b\to a)\to(c\to a))))\wedge((a\to b)\wedge(b\to c))$ $\le((a\to c)\to((b\to a)\to(c\to a)))\to(c\to b)$
16	$((c\to b)\to((b\to a)\to(c\to a)))\wedge(((a\to c)\to(b\to c))\wedge(a\to b))$ $\le(b\to c)\to(a\to c)$
17	$((((a\to c)\to(b\to c))'\wedge(c\to b))'\to(b\to a))$ $\wedge(((b\to a)\to(c\to a)')\wedge((b\to c)\to(a\to c)))\le a\to b$
18	$(((a\wedge(c\to b))\to_2(b\wedge d))\wedge(c\to(a\to(c\to b))))$ $\wedge(((a\to b)\to(b\to c))\wedge(((c\to b)\to a)\wedge(b\to a)))\le b\to d$
19	$(((d\to a)\to(b\to c)')\wedge((c\to d)\to(a\to b)'))$ $\wedge(((b\to a)'\to(d\to c))\wedge((a\to d)'\to(c\to b)))$ $\le(d\to c)\to(b\to a)'$

5.3 Mayet's E equations

In the three previous sections we have presented two apparently very different ways of generating Hilbert lattice equations. The first one was algebraic, utilizing an algebraic formulation of a geometric property possessed by any Hilbert space. The second one was based on the the properties of states (probability measures) one can define on any Hilbert space. Theorems 2.3.3 in Section 2.2 offers us a property of a third kind which any Hilbert space possesses and which can generate a class of Hilbert lattice equations and this is that each Hilbert space is defined over a particular field K.

The application to quantum theory uses the Hilbert spaces defined over real, $\mathbb{R}$, complex, $\mathbb{C}$, or quaternion (skew), $\mathbb{Q}$, fields. For these fields, in 2006, René Mayet [67] (see also [68]) used a technique similar to the one used for generating MGEs we presented in Sec. 5.2 (p. 81), to arrive at a new class of E equations we will present in this section. There are other fields over infinite-dimensional Hilbert spaces, for example a non-archimedean Keller field. [52, 32, 115], so, to get only the above three fields for an infinite-dimensional Hilbert space, we have to assume that an infinite orthonormal sequence of atoms exists in the Hilbert lattice (as well as a related harmonic conjugate condition) and invoke the theorem of Maria Pia Solèr [115] (Th. 2.3.5, p. 21). If, in a Hilbert space H over a (skew) field K, we do not have an infinite orthonormal sequence of vectors, then, for an arbitrary vector $a \in H$, there might not exist a vector $b \in Ka \stackrel{\text{def}}{=} \{x \cdot a \mid x \in K\}$ that satisfies $(b,b) = 1_K$ [where $(\,,)$ is the inner product in H]. If we have an orthonormal series of vectors, we will always have vectors satisfying the condition $(b,b) = 1_K$, and this enables us to introduce Hilbert-space-valued states,[12] as follows.

Definition 5.3.1. *A* **real Hilbert-space-valued state**—*we call it an* $\mathscr{RH}$ state—*on an orthomodular lattice L is a function* $s : L \longrightarrow \mathscr{RH}$*, where* $\mathscr{RH}$ *is a Hilbert space defined over a real field, such that*

$||s(1_L)|| = 1$*, where* $s(a)$ *is a state vector i.e.* $s(a) \in \mathscr{R}H$, $||s(a)|| = \sqrt{(s(a),s(a))}$ *is the Hilbert space norm, and* $a \in L$*; in this section we will not use the Dirac notation* $|s\rangle$ *for the state vector s, nor* $\langle s|t\rangle$ *for the inner product* (s,t)*;*

$(\forall a,b \in L)\,[\,a \perp b \;\Rightarrow\; s(a \vee b) = s(a) + s(b)\,]$*, where* $a \perp b$ *means* $a \le b'$*;*

$(\forall a,b \in L)\,[\,a \perp b \;\Rightarrow\; s(a) \perp s(b)\,]$*, where* $s(a) \perp s(b)$ *means the inner product* $(s(a), s(b)) = 0$.

[12] One could also name them *vector states* because they map elements of a Hilbert lattice to state vectors of the Hilbert space, but we decided to keep to the name introduced by Mayet [67]

Now, we select those Hilbert lattices in which we implement Definition 5.3.1 by the following definition.

Definition 5.3.2. *A* **quantum**[13] **Hilbert lattice**, $\mathcal{QHL}$, *is a Hilbert lattice orthoisomorphic to the set of closed subspaces of the Hilbert space defined over either a real field, or a complex field, or a quaternion skew field.*

In 1998 René Mayet [66] gave conditions that can be added to the orthomodular form constructed from a Hilbert lattice, although equivalent conditions that could be added to the Hilbert lattice definition are still unknown.

As with equations in the previous sections, we shall use only some properties related to states defined on a $\mathcal{QHL}$, in particular pairwise orthogonality of its elements—corresponding to pairwise orthogonality of vectors in the corresponding Hilbert space—to arrive at new equations.

We also define a complex and a quaternion Hilbert-space-valued state, called a $\mathcal{CH}$ state and a $\mathcal{QH}$ state, by mapping s to $\mathcal{CH}$ or $\mathcal{QH}$, i.e. a Hilbert space defined over a complex or quaternion field respectively.

This definition differs from Definition 2.4.1 in a crucial point, in that the state does not map the elements of the lattice to the real interval $[0,1]$ but instead to the real Hilbert space $\mathcal{RH}$. In particular, the property $a \perp b \ \Rightarrow \ s(a) \perp s(b)$ is a a restrictive requirement that allows us to define a strong set of $\mathcal{RH}$ states on a $\mathcal{QHL}$ but not on OMLs in general—even those admitting strong sets of real-valued states—nor even on all Hilbert lattices.

The conditions of Lemma 2.4.2 (p. 22) hold when we replace a real state value $m(a)$ with the square of the norm of the $\mathcal{RH}$ state value $s(a)$. For example, Eq. (2.28) becomes

$$||s(a)||^2 + ||s(a')||^2 = 1, \tag{5.29}$$

and so on. In addition, we can prove the following special properties that hold for $\mathcal{RH}$ states:

[13] Mayet [67] calls this lattice *classical Hilbert lattice* but since the real and complex fields as well as the quaternion skew filed over which the corresponding Hilbert space is defined are characteristic of its application in quantum mechanics we prefer to call the lattice quantum.

Lemma 5.3.3. *The following properties hold for any $\mathscr{RH}$ state s:*

$$s(0) = 0 \tag{5.30}$$

$$s(a) + s(a') = s(1) \tag{5.31}$$

$$||s(a)|| = 1 \qquad \Leftrightarrow \qquad s(a) = s(1) \tag{5.32}$$

$$||s(a)|| = 0 \qquad \Leftrightarrow \qquad s(a) = s(0) \tag{5.33}$$

$$s(a) \perp s(1) \qquad \Leftrightarrow \qquad s(a) = 0 \tag{5.34}$$

$$a \perp b \qquad \Rightarrow \qquad ||s(a \vee b)||^2 = ||s(a)||^2 + ||s(b)||^2 \tag{5.35}$$

$$a \le b \qquad \Rightarrow \qquad ||s(a)|| \le ||s(b)|| \tag{5.36}$$

$$a \le b \qquad \& \qquad ||s(a)|| = 1 \qquad \Rightarrow \qquad ||s(b)|| = 1 \tag{5.37}$$

$$\begin{aligned} a_i \perp a_j (1 \le i < j \le n) \quad \& \quad a_1 \vee \cdots \vee a_n = 1 \qquad \Rightarrow \\ s(a_1) + \cdots + s(a_n) = s(1) \end{aligned} \tag{5.38}$$

Proof. See Ref. [105, p. 783, Lemma 36] □

The conditions of Lemma 5.3.3, as well as the analogues of Lemma 2.4.2, also hold for $\mathscr{CH}$ and $\mathscr{QH}$ states.

The following definition of a strong set of $\mathscr{RH}$ states closely follows Definition 2.4.3, with an essential difference in the range of the states.

Definition 5.3.4. *A nonempty set S of $\mathscr{RH}$ states $s : L \longrightarrow \mathscr{RH}$ is called a* **strong set of $\mathscr{RH}$ states** *if*

$$(\forall a, b \in L)(\exists s \in S)((||s(a)|| = 1 \;\Rightarrow\; ||s(b)|| = 1) \;\Rightarrow\; a \le b)\,. \tag{5.39}$$

In an analogous manner, we define a **strong set of $\mathscr{CH}$ states** *and a* **strong set of $\mathscr{QH}$ states***.*

The following version of Theorem 5.1.1 holds. [67]

Theorem 5.3.5. *Any quantum Hilbert lattice admits a strong set of $\mathscr{RH}$ states.*

Proof. See Ref. [105, p. 784, Th. 38] □

Now, Mayet [67] showed that the lack of $\mathscr{RH}$ strong states for particular lattices, for example, the ones given in Ref. [105, p. 785, Fig. 13] gives the equations in the way similar to the one used by Megill and Pavičić [81]. For certain infinite sequences of equations, Mayet's method offers the advantage of providing a related infinite sequence of finite OMLs that violate the corresponding equation, analogous to the wagon-wheel series obtained by Godowski and presented in Section 5.1.

Let us first denote by Ω the following set of orthogonality conditions among the labeled atoms in Ref. [105, p. 785, Fig. 13(a)]: $\Omega = \{v \perp b_i,\ b_i \perp a_i,\ \ a_i \perp a_j\}$, $i, j = 1, \ldots, n$. Next, we define

$$a = a_1 \vee \cdots \vee a_n, \quad q = (a_1 \vee b1) \wedge \cdots \wedge (a_n \vee b_n), \quad b = b_1 \vee \cdots \vee b_n \,. \tag{5.40}$$

Now we are able to generate the following equations, i.e., to prove the following theorem.

Theorem 5.3.6. *In L_i, $i = 1, \ldots, n$, $n \geq 3$ given in Ref. [105, p. 785, Fig. 13(a),(b)] the following equations fail*

$$E_n: \quad \Omega \qquad \Rightarrow \quad a \wedge q = b \tag{5.41}$$

$$E'_n: \quad \Omega \quad \& \quad r \perp a \quad \Rightarrow \quad q \wedge (q \to r') \wedge (a \vee r) \leq b \tag{5.42}$$

respectively and they hold in any OML with a strong set of $\mathscr{RH}$ states.

Proof. See Refs. [105, p. 784, Th. 39] and [67]. ☐

The equations of Theorem 5.3.6, which hold in every $\mathscr{QHL}$, do not hold in every HL. Thus they are independent from all of the equations we have presented in Secs. 4.2, 5.1, and 5.2. In addition, they are independent of the modular law.

Theorem 5.3.7. *For any integer $n \geq 3$, the equation E_n does not hold in every* HL. *In particular, it is not a consequence of any n*OA *law, n*GO *law,* MGE*, or combination of them. In addition, it is not a consequence of these even in the presence of the modular law.*

Proof. See Ref. [105, p. 786, Th. 40] ☐

The two smallest equations from the class E_n, which are E_3 and E_4, respectively, read:

$$\begin{aligned}
& a \perp b \;\&\; a \perp c \;\&\; b \perp c \;\&\; a \perp d \;\&\; b \perp e \;\&\; c \perp f \\
& \quad \Rightarrow \ ((a \vee b) \vee c) \wedge (((a \vee d) \wedge (b \vee e)) \wedge (c \vee f)) \\
& \quad \leq (d \vee e) \vee f,
\end{aligned} \tag{5.43}$$

$$\begin{aligned}
& a \perp b \;\&\; a \perp c \;\&\; a \perp d \;\&\; b \perp c \;\&\; b \perp d \\
& \qquad \&\; c \perp d \;\&\; a \perp e \;\&\; b \perp f \;\&\; c \perp g \;\&\; d \perp h \\
& \Rightarrow (((a \vee b) \vee c) \vee d) \wedge ((((a \vee e) \wedge (b \vee f))
\end{aligned}$$

$$\wedge (c \vee g)) \wedge (d \vee h)) \leq ((e \vee f) \vee g) \vee h. \tag{5.44}$$

These equations pass in most OMLs that characterize properties of both quantum (Hilbert) and classical spaces including all our lattices with equal number of vertices (atoms) and edges (blocks) that we primarily consider in this paper. However, Eq. (5.43) fails in the OML (b) shown in Fig. 2 of Ref. [101, p. 102103-15, Fig. 2], and Eq. (5.44) fails in the OML (c) of that figure. Eq. (5.43) also fails in OML L42 of our Fig. 6.3 (p. 105), which is an OML that violates no other known Hilbert lattice equation (see Ref. [76, p. 2365, footnote 4]).

Chapter 6

OTHER $\mathscr{C}(H)$ PROPERTIES

There are several classes of lattices, specified by quantified conditions, that include Hilbert lattices but which are not currently known to be equational varieties. An open problem is whether equational conditions can be derived from them. In this chapter, we look at two such conditions.

6.1 Modular symmetry

Definition 6.1.1. *[59] Two elements a and b of a lattice* L *are a* **modular pair***, and we write* $M(a,b)$*, iff for every c in* L*,*

$$c \le b \quad \Rightarrow \quad (c \vee a) \wedge b = c \vee (a \wedge b). \tag{6.1}$$

Elements a, b are a **dual modular pair***, and we write* $M^*(a,b)$*,*[1] *iff for every c in* L*,*

$$b \le c \quad \Rightarrow \quad (c \wedge a) \vee b = c \wedge (a \vee b). \tag{6.2}$$

There are several equivalents to $M(a,b)$. From them we can also obtain their $M^*(a,b)$ analogues by duality, i.e. by interchanging $\vee$ and $\wedge$ as well as $\le$ and $\ge$.

Theorem 6.1.2. *The following conditions hold in any lattice:*

$$M(a,b) \quad \Leftrightarrow \quad \forall c(c \le b \Rightarrow (c \vee a) \wedge b = c \vee (a \wedge b)) \tag{6.3}$$

$$M(a,b) \quad \Leftrightarrow \quad \forall c(c \le b \Rightarrow (c \vee a) \wedge b \le c \vee (a \wedge b)) \tag{6.4}$$

[1] Other notations for $M(a,b)$ and $M^*(a,b)$ are $(a,b)M$ and $(a,b)M^*$ [59], and aMb and aM^*b [93] [116].

$$M(a,b) \quad \Leftrightarrow \quad \forall c(((c\wedge b)\vee a)\wedge b = (c\wedge b)\vee(a\wedge b)) \tag{6.5}$$

$$M(a,b) \quad \Leftrightarrow \quad \forall c(((c\wedge b)\vee a)\wedge b \le (c\wedge b)\vee(a\wedge b)) \tag{6.6}$$

$$M(a,b) \quad \Leftrightarrow \quad \forall c(a\wedge b\le c \Rightarrow ((c\wedge b)\vee a)\wedge b \le c). \tag{6.7}$$

Proof. **For Eqs. (6.3) through (6.6):** These are easily derived from Def. 6.1.1 using the modular law equivalents given in Th. 7.2.2 below (p. 113).

For Eq. (6.7): From Eq. (6.6),

$$M(a,b) \Rightarrow (c\wedge b)\vee a)\wedge b \le (c\wedge b)\vee(a\wedge b). \tag{6.8}$$

In any lattice, $a\wedge b\le c$ and $c\wedge b\le c$ imply $(c\wedge b)\vee(a\wedge b)\le c$. Since $c\wedge b\le c$ in any lattice, we have

$$a\wedge b\le c \Rightarrow (c\wedge b)\vee(a\wedge b)\le c. \tag{6.9}$$

From Eqs. (6.8) and (6.9) and transitivity of $\le$, then quantifying with c,

$$M(a,b) \Rightarrow \forall c(a\wedge b\le c \Rightarrow ((c\wedge b)\vee a)\wedge b\le c). \tag{6.10}$$

Conversely, define T as the term $(d\wedge b)\vee(a\wedge b)$. From the specialization rule of predicate calculus,

$$\forall c(a\wedge b\le c \Rightarrow ((c\wedge b)\vee a)\wedge b\le c) \Rightarrow (a\wedge b\le T \Rightarrow ((T\wedge b)\vee a)\wedge b\le T).$$

Since $a\wedge b\le T$ in any lattice,

$$\forall c(a\wedge b\le c \Rightarrow ((c\wedge b)\vee a)\wedge b\le c) \Rightarrow (T\wedge b)\vee a)\wedge b\le T. \tag{6.11}$$

In any lattice, $T\le b$ since $d\wedge b\le b$ and $a\wedge b\le b$, so $T\wedge b = T$. Thus

$$(T\wedge b)\vee a = T\vee a. \tag{6.12}$$

By the lattice absorption law, $(a\wedge b)\vee a = a$, so $T\vee a = (d\wedge b)\vee((a\wedge b)\vee a) = (d\wedge b)\vee a$. Combining with Eq. 6.12 and conjoining both sides with b, we have

$$((T\wedge b)\vee a)\wedge b = ((d\wedge b)\vee a)\wedge b. \tag{6.13}$$

Combining Eqs. (6.11) and (6.13), expanding the term T, and finally quantifying over d, we obtain

$$\begin{aligned} &\forall c(a\wedge b\le c\Rightarrow((c\wedge b)\vee a)\wedge b\le c)\\ &\qquad\Rightarrow\forall d((d\wedge b)\vee a)\wedge b\le(d\wedge b)\vee(a\wedge b)\\ &\qquad\Leftrightarrow M(a,b). \end{aligned} \tag{6.14}$$

where the last step is from Eq. (6.6). Eq. (6.7) follows from Eqs. (6.10) and (6.14). ☐

If all elements a and a' in an OL satisfy modular pair or dual modular pair condition, then the OL is an OML. In other words, $M(a,a')$ and $M^*(a,a')$ are equivalent to the OML law.

Theorem 6.1.3. *(a) An* OL *in which $M(a,a')$ is an* OML *and vice versa. (b) An* OL *in which $M^*(a,a')$ is an* OML *and vice versa.*

Proof. See Ref. [59, p. 132]. ☐

The modular law itself is simply expressed by the modular pair condition.

Theorem 6.1.4. *An lattice L is modular iff for all a,b in L, $M(a,b)$ or equivalently $M^*(a,b)$.*

Proof. This is obvious from the modular law equivalents Eqs. (7.5) and (7.7) (p. 114). ☐

The importance of modular pairs is that certain symmetry conditions hold in a Hilbert lattice (and thus in the subspace lattice $\mathscr{C}(H)$ of an infinite-dimensional Hilbert space H), even though the modular law itself does not.

Theorem 6.1.5. *For any elements a,b in an* HL*, the following conditions, hold:*

$$M(a,b)=M(b,a) \tag{6.15}$$

$$M^*(a,b)=M^*(b,a) \tag{6.16}$$

Proof. See Ref. [60, p. 168, Lemma 5]. (Note that Maeda defines $M^*(a,b)$ with the arguments reversed on p. 165, Def. 1, which he changes to our convention in subsequent literature. This does not affect the statement of this theorem, but the reader must be aware of it in order to follow the proof.) ☐

These conditions are called **modular symmetry** or **M-symmetry**, and **dual modular symmetry** or **M*-symmetry**, respectively. These are quantified conditions rather than equations.

For working with them, it can be convenient to express them in an expanded form. As a quantified inference, M-symmetry can be expressed as

$$\begin{aligned}&\forall c(c \le b \Rightarrow (c \vee a) \wedge b = c \vee (a \wedge b))\\ &\quad \Leftrightarrow \forall c(c \le a \Rightarrow (c \vee b) \wedge a = c \vee (b \wedge a))\end{aligned} \tag{6.17}$$

or in prenex normal form, which is useful for testing with a computer program such as our `lattice.c`, as

$$\begin{aligned}&\exists c \forall d((c \le a \Rightarrow (c \vee (b \wedge a) = (c \vee b) \wedge a))\\ &\quad \Rightarrow (d \le b \Rightarrow (d \vee (a \wedge b) = (d \vee a) \wedge b)))\end{aligned} \tag{6.18}$$

The M-symmetry condition is much stronger than any known Hilbert lattice equation. Of course, it is strictly weaker than the modular law, since it holds in any $\mathscr{C}(H)$ whereas the modular law fails in any (infinite-dimensional) $\mathscr{C}(H)$ [49, p. 67, Prop. 5]. On the other hand, it fails in all known non-modular lattices tested by this author. In particular, it fails in the lattice of Fig. 6.1[2] and the Greechie diagram of Fig. 6.2,[3] both of which satisfy all known equations that hold in HL (i.e. any finite- or infinite-dimensional $\mathscr{C}(H)$).

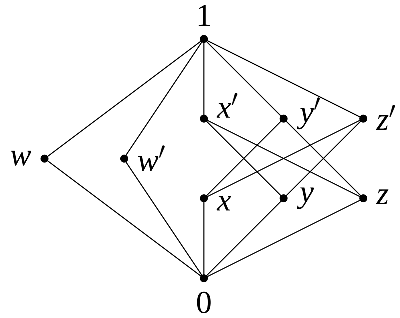


Figure 6.1: Hasse diagram for a non-modular, orthoarguesian lattice (from Ref. [6, p. 42, Fig. 12] or Ref. [49, p. 160, Fig. 11.2]) in which M-symmetry and M*-symmetry fail.

In a **relatively atomic** lattice ($a < b$ implies there is a $c \le b$ such that c covers a), M-symmetry is equivalent to the **exchange axiom**[4] (a covers $a \wedge b$ implies $a \vee b$ covers b) [49, p. 140, Prop. 1(iii)] which holds in a Hilbert lattice [4, p. 167, Th. 14.8.10]. Unlike the exchange axiom, M-symmetry involves no logical negation when expanded to lattice primitives. (The

[2] Because of the two-atom block $\{w, w'\}$, the MMP encoding of Fig. 6.1 cannot be used with the program `latticeg.c`, which currently handles only 3- and 4-atom blocks. However, it is hard-coded as "`Beran Fig. 12 (OA, non-modular)`" in the program `lattice.c` [Sec. A.4 (p. 151)].

[3] `123,345,567.` is an MMP encoding for Fig. 6.2.

[4] A lattice satisfying the exchange axiom is also called **semimodular** [15, p. 23]

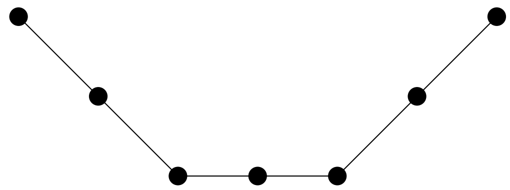


Figure 6.2: Greechie diagram for a non-modular, orthoarguesian lattice (from Ref. [49, p. 155, Fig. 10.2]) in which M-symmetry and M*-symmetry fail.

expression "*a* covers *b*" requires that *a* not be equal to *b*.) In this sense, it is one step closer to an equation than the exchange axiom, since an equation cannot involve logical negation.

6.1.1 The search for an equation

M-symmetry is still a quantified condition (in other words, it has an existential quantifier in prenex normal form) and thus does not necessarily generate an equational variety. An interesting open question is whether an equation—stronger than the OML law and ideally independent from any other known equation—can be derived from M-symmetry. An important result of Whitman [124] implies that an equation (identity) cannot be derived from M-symmetry alone. However, it does not eliminate the possibility of deriving an equation from M-symmetry together with other properties that hold in a Hilbert lattice. In any case, currently there is no known equation that has been derived exploiting the strength of M-symmetry.

To obtain such an equation, one possible approach (whose investigation is ongoing project of this author) is to find a quantifier-free expression (a set of polynomial equations connected with classical logical 'and') $E(a,b,\ldots)$ such that

$$E(a,b,\ldots) \quad \Rightarrow \quad M^*(b,a) \tag{6.19}$$

holds in OML (or in some other known HL condition). Then

$$E(a,b,\ldots) \quad \Rightarrow \quad M^*(a,b) \tag{6.20}$$

will also hold in HL and (after removal of $M^*(a,b)$ quantifier) will be an equational inference that holds in HL, hopefully stronger than the first condition.

The dual modular symmetry condition $M^*(a,b)$ can be expressed with the dual of Eq. (6.7) as follows.

$$M^*(a,b) \quad \Leftrightarrow \quad \forall c(c \le a \vee b \Rightarrow c \le ((c \vee b) \wedge a) \vee b). \tag{6.21}$$

This form allows us to express Eq. (6.19) as

$$E(a,b,\ldots) \Rightarrow (c \le b \vee a \Rightarrow c \le ((c \vee a) \wedge b) \vee a). \tag{6.22}$$

and Eq. (6.20) as

$$E(a,b,\ldots) \Rightarrow (c \le a \vee b \Rightarrow c \le ((c \vee b) \wedge a) \vee b). \tag{6.23}$$

Note that because the quantifier was removed, $E(a,b,\ldots)$ must not contain the variable c. To recap, because of the modular symmetry of HL [Th. 6.1.5 (p. 102)], Eq. (6.22) holds in any HL iff Eq. (6.23) holds in any HL.

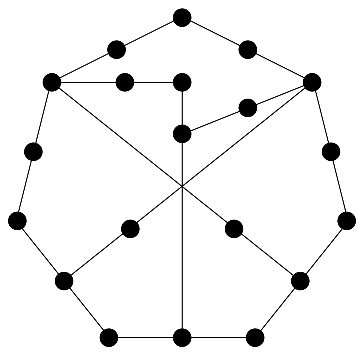


Figure 6.3: Greechie diagram for OML L42 (from Ref. [76, p. 2366, Fig. 7(b)]).

We will illustrate the procedure by showing an example of a conjecture. Empirically,[5] we found the following candidate for the expression $E(a,b,\ldots)$:

$$E(a,b,\ldots) \stackrel{\text{def}}{\Leftrightarrow} aCy \ \& \ y \wedge z \le a \ \& \ b \le z \ \& \ b \le (a \wedge z) \vee y. \tag{6.24}$$

This results in the inference

$$[aCy \ \& \ y \wedge z \le a \ \& \ b \le z \ \& \ b \le (a \wedge z) \vee y]$$

[5]This was found by experimentally adding conditions (hypotheses) to $E(a,b,\ldots)$ that were just strong enough so that Eq. (6.22) passed in all tested OMLs, but not so strong that Eq. (6.23) also passed.

$$\Rightarrow (c \le b \vee a \Rightarrow c \le ((c \vee a) \wedge b) \vee a), \tag{6.25}$$

which implies, in any HL (via modular symmetry), the inference

$$[aCy \ \&\ y \wedge z \le a \ \&\ b \le z \ \&\ b \le (a \wedge z) \vee y]$$
$$\Rightarrow (c \le a \vee b \Rightarrow c \le ((c \vee b) \wedge a) \vee b). \tag{6.26}$$

Eq. (6.25) does not fail in any finite OML (Greechie diagram) that we tried. On the other hand, Eq. (6.26) fails in OML lattice L42[6] (see Fig. 6.3), indicating non-OML behavior.

The interested reader can quickly perform a rough check of this result with the program `lattice.c` [Sec. A.4 (p. 151)] as follows. For Eq. (6.22), the condition

```
lattice 'a[y' '(y^z)<a' 'b<z' 'b<((a^z)vy)' 'c<(bva)' 'c<(((cva)^b)va)'
```

passes in all OML lattices tested by the program. But the modular symmetric (and thus HL-equivalent) condition of Eq. (6.23),

```
lattice 'a[y' '(y^z)<a' 'b<z' 'b<((a^z)vy)' 'c<(avb)' 'c<(((cvb)^a)vb)'
```

fails in OML lattice L42.

This result was unexpected and intriguing. It means that if Eq. (6.25) holds in all OMLs, then Eq. (6.26) would give us a rather strong and probably previously unknown equational condition that holds in all HLs.

The problem is that we have been unable to prove (or disprove) that Eq. (6.25) holds in all OMLs, in spite of considerable effort. So as of this writing it remains a conjecture.

Other similar experiments assigning $E(a, b, \ldots)$ have lead to the observation that the modular symmetry transformation from Eq. (6.22) to Eq. (6.23) tends to "strengthen" almost any OML or near-OML version of Eq. (6.22). Unfortunately, just as in the case above, we were unable to prove that any suitable version of Eq. (6.22) held in all OMLs (or even in all HLs, which would suffice). Nonetheless, it still seems that this method holds some promise for obtaining new HL equations and merits further study.

We mention that an inference resulting from an assignment to $E(a, b, \ldots)$ can sometimes be turned into or derived from an equation without hypotheses, by making appropriate substitution instances that eliminate hypotheses. For some purposes, it can be easier or more efficient to study the conjecture as a stand-alone equation. For example, the automated theorem prover EQP

[6] `123,145,167,189,2AB,4CD,6EF,8GH,ACE,BGI,DGJ,FGK.` is an MMP encoding for L42 (Fig. 6.3).

[70] requires an equation with no hypotheses. Of course, an equivalent equation is possible only if the inferential condition describes an equational variety and not just a quasi-variety (p. 63).

For example, as we show in the next lemma, Eq. (6.25) can be derived from the equation

$$c \wedge (a \vee (((a \wedge z) \vee (y \wedge (a \vee (y \wedge a')))) \wedge z \wedge b)) \\ \le ((((y \wedge z) \vee a) \vee c) \wedge b) \vee ((y \wedge z) \vee a). \tag{6.27}$$

Therefore if we can prove that Eq. (6.27) holds in all OMLs, it will follow that the conjectured Eq. (6.25) also holds in all OMLs, leading to a (likely) new HL condition in the form of Eq. (6.26). Thus the conjecture becomes whether Eq. (6.27) holds in all OMLs, and this problem has similarly eluded a proof or disproof so far.

Lemma 6.1.6. *In any* OML, *Eq. (6.25) follows from Eq. (6.27).*

Proof. The hypothesis aCy of Eq. (6.25) implies $y = (y \wedge (a \vee (y \wedge a')))$. Thus starting with Eq. (6.27) and making this substitution into it, we have

$$aCy \quad \Rightarrow \quad c \wedge (a \vee (((a \wedge z) \vee y) \wedge z \wedge b)) \\ \le ((((y \wedge z) \vee a) \vee c) \wedge b) \vee ((y \wedge z) \vee a). \tag{6.28}$$

The hypothesis $y \wedge z \le a$ implies $a = (y \wedge z) \vee a$. Substituting into Eq. (6.28),

$$[aCy \ \& \ y \wedge z \le a] \\ \Rightarrow \ c \wedge (a \vee (((a \wedge z) \vee y) \wedge z \wedge b)) \le ((a \vee c) \wedge b) \vee a. \tag{6.29}$$

The hypotheses $b \le z$ and $b \le (a \wedge z) \vee y$ imply $b \le ((a \wedge z) \vee y) \wedge z$, which in turn implies $b = ((a \wedge z) \vee y) \wedge z \wedge b$. Substituting this equality into Eq. (6.29),

$$[aCy \ \& \ y \wedge z \le a \ \& \ b \le z \ \& \ b \le (a \wedge z) \vee y] \\ \Rightarrow \ c \wedge (a \vee b) \le ((a \vee c) \wedge b) \vee a. \tag{6.30}$$

Finally, the hypothesis $c \le a \vee b$ implies $c = c \wedge (a \vee b)$. The substitution of this equality into Eq. (6.29) results in Eq. (6.25), as required. $\square$

6.1.2 O-symmetry

The closed subspaces of a Hilbert space H also satisfy a property even stronger than M-symmetry, called **O-symmetry**.

Definition 6.1.7. *A lattice is called* **O-symmetric** *iff for all a,b*

$$M(a,b) \quad \Leftrightarrow \quad M^*(b',a') \tag{6.31}$$

Unlike the relatively straightforward proof of M-symmetry in Ref. [60], the proof of the O-symmetry of $\mathscr{C}(H)$ is quite difficult, using deep topological facts in an apparently essential way [39, p. 1520]. The full development of this proof spans a significant portion of Maeda and Maeda's book [59] (and it references, but does not prove, these topological facts), culminating in the following theorem:

Theorem 6.1.8. *[59, p. 155, Th. 34.8] The set of closed subspaces of a Hilbert space is O-symmetric.*

To search for an equation derived from O-symmetry, a possible approach could be similar to that leading to Eq. (6.20), with a simple substitution of $M(a',b')$ for $M^*(a,b)$. Just as is the case for Eq. (6.20), it remains an open problem whether this approach will lead to an new equation holding in HL.

6.2 Superposition

The relationship between the superposition principle of a Hilbert lattice [Def. 2.3.1 (p. 19)] and the usual superposition in quantum mechanics can be understood intuitively as follows. In Hilbert space, the superposition of two vectors x and y (corresponding to pure states) is the vector sum $x+y$. In a Hilbert lattice (HL), this concept can be represented with atoms [Def. 2.3.1(2) (p. 19)], which correspond to one-dimensional subspaces (also called "rays"), as follows. Suppose x and y are non-zero vectors in the one-dimensional subspaces represented by atoms a and b, thus determining those subspaces. The superposition $x+y$ will be contained in the join $a\vee b$, which corresponds to a 2-dimensional subspace. Because of superposition property of the Hilbert lattice, there is an atom c that is covered by $a\vee b$ and which corresponds to the 1-dimensional subspace containing $x+y$.

While the superposition principle tells us that such a third 1-dimensional subspace (corresponding to atom c) exists, it does not tell us which one it is, i.e. c need not be unique. For example, the 1-dimensional subspace generated by the superposition $x+\frac{1}{2}y$ would correspond to a different atom than the subspace generated by $\frac{1}{2}x+y$.

The superposition principle is a distinctly "quantum mechanical" property of a lattice, as the following theorem shows.

Theorem 6.2.1. *[4, p. 165, Th. 14.8.2] An* OML *is classical (distributive) iff no pair of pure states admits (quantum) superpositions.*

It is instructive to look at the proof of a special case of this theorem.

Theorem 6.2.2. *No atomic distributive lattice with more than one atom satisfies superposition principle 3(a) of Def. 2.3.1 (p. 19).*

Proof. Let $a\neq b$ be two atoms. Suppose there is a third atom c such that $c\le a\vee b$. If the lattice is distributive, we would have:

$$c\wedge(a\vee b)=(c\wedge a)\vee(c\wedge b) \tag{6.32}$$

Since $c\le a\vee b$, we have $c\wedge(a\vee b)=c\neq 0$ since c is an atom. However, $c\wedge a=0$ (since c and a are different atoms) and similarly, $c\wedge b=0$, which contradicts the equation. □

We note that the 2^1 Boolean algebra satisfies this superposition principle vacuously, since there are no two different atoms a and b (1 is the only atom) to satisfy the hypothesis. However, this can be considered an artifact and not the intention of the definition, and if this is important for an application we can narrow the definition to exclude lattices with less than two atoms. The 2^1 BA is already excluded as a Hilbert lattice by condition 4 of Def. 2.3.1 (p. 19) (the minimum height requirement).

The superposition principle of Def. 2.3.1 can be formulated in prenex normal form (to make it easier to use in conjunction with certain first-order logic algorithms, including our `latticeg.c` program) as follows[7]

$$\begin{aligned}
&(\exists c)(\exists z)(\forall w)\\
&\quad((((\neg(a=0)\ \&\ ((\neg(z=0)\ \&\ (z\le a))\ \Rightarrow\ (z=a)))\ \&\ (\neg(b=0)\\
&\quad\&\ ((\neg(z=0)\ \&\ (z\le b))\ \Rightarrow\ (z=b))))\ \&\ \neg(a=b))\\
&\quad\ \Rightarrow\ ((\neg(c=0)\ \&\ ((\neg(w=0)\ \&\ (w\le c))
\end{aligned}$$

[7]In the format required by `latticeg.c`, this equation is expressed as `] c ] z @ w ( ( ( (~( a = 0 ) & ( ( ~(z=0)&(z<a))}(z =a)))&(~(b =0)&((~(z=0)&(z<b))}(z=b))))&~(a=b))}((~(c=0)&( (~(w=0)&( w<c))}(w=c))) & ((~(c=a)&~(c=b))&(c<(avb))))).`

$$\Rightarrow \ (w = c))) \ \& \ ((\neg(c = a) \ \& \ \neg(c = b)) \ \& \ (c \leq (a \vee b))))) \tag{6.33}$$

where $\neg$, &, and $\Rightarrow$ are classical meta-operations: negation, conjunction, and implication, respectively.

Not all OMLs satisfy the superposition principle, even non-distributive ones that admit states. Eq. (6.33), tested with the program `latticeg.c` [Sec. A.1 (p. 146)] against an exhaustive list of all Greechie diagrams with 3 atoms per block (obtained with Brendan McKay's program `nauty` mentioned in Sec. A.1), was used to find the smallest one in which superposition holds. It is shown in Fig. 6.4[8] and consists of an inverted pentagram inside of a pentagon.

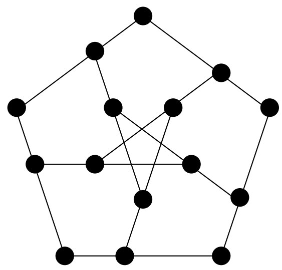


Figure 6.4: The smallest 3-atom-per-block Greechie diagram that admits superposition.)

Irreducibility (meaning 0 and 1 are the only lattice elements that commute with all other elements) and the **covering property** (for every a and every atom p such that a $a \wedge p = 0$, the element $a \vee p$ covers a) follow from the superposition principle [4, pp. 166, 167].

As can be seen from the lattice failures mentioned above, the superposition principle adds a strong property to an HL that is not present in the known equations. Thus it is natural to ask whether an equational property can be derived from it. The superposition principle is a quantified condition, not an equation, and moreover [unlike modular symmetry discussed in Sec. 6.1 (p. 100)] it requires logical negation, as does any condition involving the covering relation. However, superposition implies the exchange axiom (defined in the Sec. 6.1), which in turn is equivalent to modular symmetry in HL, which we showed to be "closer" to an equational condition. Open questions yet to be answered are (1) whether the superposition principle, or some reasonably strong condition derived from it, can be stated in a negation-free form (i.e. not requiring mention of atoms or the covering property), analogous to modular symmetry, and (2) whether the superposition principle, by itself or in conjunction with modular symmetry, can help us to find a new equation that holds in HL.

[8] `7BC,78F,6AD,69E,5CD,49B,38A,2EF,134,125.` is an MMP encoding for Fig. 6.4.

Chapter 7

FINITE-DIMENSIONAL HILBERT SPACE

Finite-dimensional Hilbert spaces are applicable to many problems in quantum mechanics, such as experiments involving particle spin states. In particular, most approaches to quantum computation involve finite dimensions.

The subspace lattice $\mathscr{C}(H)$ of a finite-dimensional Hilbert space satisfies a number of equations that are stronger than those holding in all Hilbert lattices (which include the $\mathscr{C}(H)$s for infinite-dimensional Hilbert spaces), in particular the modular law and Arguesian law which we will discuss in this chapter.

In Sec. 7.1, we show a concrete example of a finite-dimensional Hilbert space and briefly discuss the Hilbert lattice generated by its $\mathscr{C}(H)$. The theory of equations holding in the $\mathscr{C}(H)$ of a finite Hilbert space begins with Sec. 7.2 (p. 113).

7.1 Example: Hilbert lattice for a 2-qubit system

The definition of a Hilbert lattice [Def. 2.3.1 (p. 19)] requires that the lattice height be at least 4. This is the smallest height which allows a Hilbert space to be reconstructed from a Hilbert lattice [40, p. 215, Th. 3.5] and corresponds to a 4-dimensional Hilbert space. An example is a 2-qubit system used in quantum computing, using for example the spin states of two spin-$\frac{1}{2}$ particles or the polarizations of two photons.

In such a system, the state of each particle can be represented by a vector in a 2-dimensional Hilbert space H_2 with basis vectors $|0\rangle \stackrel{\text{def}}{=} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle \stackrel{\text{def}}{=} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. The compound system of two particles belongs to the 4-dimensional Hilbert space $H_4 = H_2 \otimes H_2$ where $\otimes$ is the tensor product [91, p. 71]. The basis vectors of H_4 are the tensor products of basis vectors from H_2, resulting in the H_4 basis vectors

$$|00\rangle \stackrel{\text{def}}{=} |0\rangle|0\rangle \stackrel{\text{def}}{=} |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$|10\rangle$, $|01\rangle$, and $|11\rangle$. The 1-dimensional subspaces spanned by these 4 basis vectors are atoms of the lattice $\mathscr{C}(H_4)$ and provide the basis for the projective subspaces [40, p. 212] constructed from the lattice.

We will let $L = \mathscr{C}(H_4)$, which will be our HL example in what follows. Following the discussion after Def. 2.3.1 (p. 19), it is easy to see that L is a Hilbert lattice i.e. $L \in$ HL.

First, let us look at some of the properties of L.

From the HL definition [Def. 2.3.1 (p. 19)] alone, L has an infinite number of atoms [44]. This is also obvious from the $\mathscr{C}(H_4)$ definition, because the sum of any two basis vectors in any proportion is a new vector (which defines a pure state[1]) that spans a new 1-dimensional subspace (atom).

The height of L is 4. This can be seen as follows. The join of two atoms corresponds to the subspace spanned by their corresponding Hilbert space vectors. More generally, the join of any two lattice elements corresponds to the subspace sum[2] of the subspaces corresponding to the $\mathscr{C}(H_4)$ lattice elements. In particular, the subspace spanned by all 4 basis vectors, corresponding to a lattice element of height 4, is all of H_4. There is no shorter chain that generates lattice 1 since H_4 needs at least 4 vectors to span all of H_4.

Entangled states (state vectors that cannot be expressed as a tensor product of vectors from H_2), such as the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ [91, p. 25], of course correspond to atoms in L since they are pure state vectors. Apparently there is no way to distinguish these from non-entangled states in a Hilbert lattice, since the Hilbert space reconstructed from L is simply a 4-dimensional Hilbert space without further structure (such as being a tensor product of two smaller spaces). Some preliminary work has been done on defining a "tensor product" for Hilbert lattices [63]; if such an effort is successful, it may be possible to add additional structure to a Hilbert lattice

[1]For the general definition of a pure state see Ref. [76, p. 2347, Def. 3.7]. Pure states are used to justify the n-Go equations [Th. 5.1.3 (p. 79)]; see e.g. the proof of Ref. [76, p. 2348, Th. 3.8].

[2]Since H_4 is finite-dimensional, the subspace sum of two subspaces equals their join in the lattice $\mathscr{C}(H_4)$. This is not necessarily true for infinite-dimensional subspaces.

that expresses some aspect of entanglement.

In accordance with the superposition principle [Def. 2.3.1 (p. 19)], the superposition of two state vectors in Hilbert space corresponds to an atom directly under (covered by) the join of the two state vectors. For example, the atom corresponding to the Bell state vector above is covered by the join of the basis atoms corresponding to $|00\rangle$ and $|11\rangle$. The superposition principle tells us that such an atom exists, but it does not specify it uniquely. For example, all atoms corresponding to vectors in the 2-dimensional subspace (plane) spanned by $|00\rangle$ and $|11\rangle$, except for those basis atoms themselves, satisfy the conditions of the superposition principle. At first glance, then, it may seem that the Hilbert lattice has "lost" information needed to specify particular atoms. But in fact, once the division ring (providing the numerical vector coefficients for a superposition) and the vectors themselves are constructed [40], it again becomes possible to specify specific vectors corresponding to a superposition.

The complete set of elements of Hilbert lattice L consists of the lattice element $\mathbf{0}_L$ (corresponding to the empty subspace of $\mathscr{C}(H_4)$), the atoms (corresponding to the 1-dimensional subspaces), the lattice elements that correspond to 2- and 3-dimensional subspaces, and the lattice element $\mathbf{1}_L$ (corresponding to the 4-dimensional full space). Since L is an HL, all of the equations of Sec. 1.3 (p. 7) hold. In addition, since the underlying Hilbert space is finite-dimensional, all of the equations described in Secs. 7.2, 7.3 (p. 123), and 7.4 (p. 140) hold. As we mentioned in Sec. 1.3, L is not an equational variety, meaning that it cannot be completely specified by equations, and a long-term goal is determining to what extent it can be thus specified.

7.2 Modular lattices

Definition 7.2.1. *[99, Def. 3.8, p. 193]. A* **modular lattice** *or* ML *is a lattice (a member of the class* Lat*) in which the following equation, called the* **modular law**, *holds.*

$$b \le a \quad \Rightarrow \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \tag{7.1}$$

and vice versa. A **modular ortholattice** *or* MOL *is an ortholattice (a member of the class* OL*) in which Eq. 7.1 holds and vice versa. An* MOL *is sometimes also called a* **modular orthocomplemented lattice**.

The following theorem lists some equivalent forms of the modular law.

Theorem 7.2.2. *A lattice in which any of the following conditions hold:*

$$a \le b \quad \Rightarrow \quad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \tag{7.2}$$

$$a \wedge (b \vee (a \wedge c)) = (a \wedge b) \vee (a \wedge c) \tag{7.3}$$

$$a \vee (b \wedge (a \vee c)) = (a \vee b) \wedge (a \vee c) \tag{7.4}$$

$$a \le c \quad \Rightarrow \quad a \vee (b \wedge c) = (a \vee b) \wedge c \tag{7.5}$$

$$a \le c \quad \Rightarrow \quad (a \vee b) \wedge c \le a \vee (b \wedge c) \tag{7.6}$$

$$a \le b \quad \& \quad a \vee c = b \vee c \quad \& \quad a \wedge c = b \wedge c \quad \Rightarrow \quad a = b \tag{7.7}$$

$$a \vee (b \wedge (a \vee c)) = a \vee (c \wedge (a \vee b)) \tag{7.8}$$

$$a \wedge (b \vee c) = a \wedge ((b \wedge (a \vee c)) \vee c) \tag{7.9}$$

$$(a \wedge (b \vee c)) \vee (b \wedge c) = (a \vee (b \wedge c)) \wedge (b \vee c) \tag{7.10}$$

is an ML *and vice versa.*

Proof. For Eq. (7.3), see Ref. [49, p. 14]. Eqs. (7.2) and (7.4) are obvious duals, and the principle of duality holds for modular lattices [1, p. 146]. For Eqs. (7.5) and (7.6), see Ref. [1, Def. 5-2, p. 146]. For Eq. (7.7), see Ref. [1, Th. 5-6, p. 146]. For Eq. (7.8), see Ref. [94, p. 41]. For Eq. (7.9), see Ref. [29, Th. 1, p. 211]. For Eq. (7.10), see Ref. [94, p. 40]. ☐

Eqs. (7.1) and (7.2) expresses the modular law in the form of the distributive law weakened by a hypothesis, thus showing that the class ML includes the class of all distributive lattices. Eq. (7.9) is useful because it can be directly applied to expressions of the form $a \wedge (b \vee c)$ (one side of the distributive law) with no preconditions. Note that Eq. (7.10) is self-dual. Eqs. (7.9) and (7.10) are variations of what is called the **shearing identity**.

The modular law holds in an HL (Hilbert lattice) iff the dimension of the Hilbert space H is finite [49, Prop. 5, p. 67]. We show one such proof below, which is very similar to the proof of the analogous **Dedekind's law** for projective subspaces [2, p. 9].

Lemma 7.2.3. *Let a, b, c be any subspaces of a vector space. Then*

$$a \subseteq c \quad \Rightarrow \quad (a + b) \cap c \subseteq a + (b \cap c) \tag{7.11}$$

where $\subseteq$, $\cap$, and $+$ are the subset relation, set intersection, and subspace sum respectively.

Proof. We will use $+$ and $-$ to denote vector sum and difference. Suppose $z \in (a + b) \cap c$. Then $z \in c$, and there exist $x \in a$, $y \in b$ such that $z = x + y$. By the hypothesis, $a \subseteq c$, we have

$x \in c$. Thus $y = z - x \in c$, so $y \in b \cap c$. Thus $z = x + (z - x) \in a + (b \cap c)$, establishing the conclusion. ☐

In any finite-dimensional subspace, $a + b = a \vee b$, where $\vee$ is the join of the lattice of subspaces of the vector space. Also, $\subseteq$ and $\cap$ correspond to lattice meet and ordering. Thus we have

Theorem 7.2.4. *The lattice of subspaces of a finite-dimensional Hilbert space is modular.*

Proof. We make the above operation and relation substitutions into Dedekind's law Eq. (7.11) to arrive at

$$a \leq c \quad \Rightarrow \quad (a \vee b) \wedge c \leq a \vee (b \wedge c)$$

which is the modular law in the form of Eq. (7.6). ☐

Some of the laws holding in $\mathscr{C}(H)$ for an infinite-dimensional Hilbert space H, such as the OML law and the 3OA law, hold in any MOL. Th. 4.4.8 (p. 56) shows the derivation of the 3OA law. Whether some others, such as the nOA law for $n > 3$, hold in MOL is, to our knowledge, an open problem. However, it is known that Mayet's E equations [Th. 5.3.6 (p. 98)] do not hold in all MOLs [67, p. 1264, Th. 4.2] even though it holds in $\mathscr{C}(H)$ for all infinite-dimensional Hilbert spaces. Importantly, this shows that HL equations, in general, are not merely consequences of the modular law that have been "weakened" to hold in infinite-dimensional Hilbert spaces.

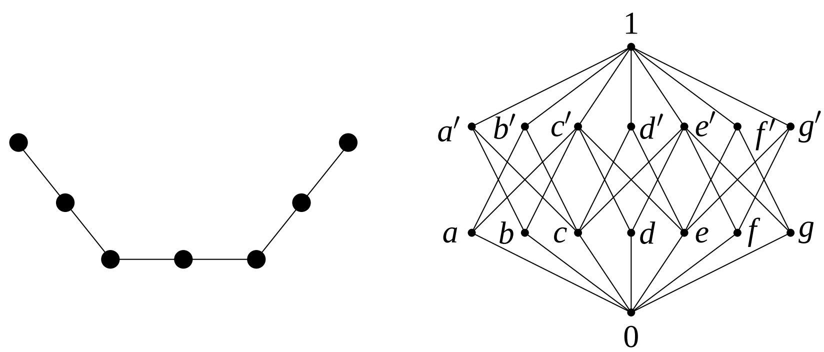


Figure 7.1: Greechie and Hasse diagrams for the non-modular OML with MMP encoding `123,345,567..`

7.2.1 Characterization of modular lattices

Modular lattices are characterized by the following theorem.

Theorem 7.2.5. *[49, p. 33] A lattice is modular iff it does not include a pentagonal sublattice.*

Proof. A pentagonal sublattice violates the modular law, as is easily shown, and if a sublattice violates an equation, so does the parent lattice [101, Lemma II.12, p. 102103-14]. For a proof of the converse, see Ref. [29, Th. 2, p. 80; Fig. I.2.3, p. 14]. □

Contrary to what might be naïvely expected, Th. 7.2.5 does not characterize the modular law in the sense that it can show that an equation derived from the modular law is equivalent to the modular law. We will prove this below in Th. 7.2.8 (p. 122).

The literature (e.g. Ref. [29, p. 211]) sometimes uses the informal but slightly ambiguous phrase "contains a pentagon" in the statement of Th. 7.2.5. For clarity, we will show an example using the lattice of Fig. 7.1 (called Dilworth's lattice D_{16} [6, p. 143]). This lattice is non-modular because it includes the pentagonal sublattice consisting of the nodes $\{0, a, b', 1, f\}$ along with the ordering relations from the parent lattice.

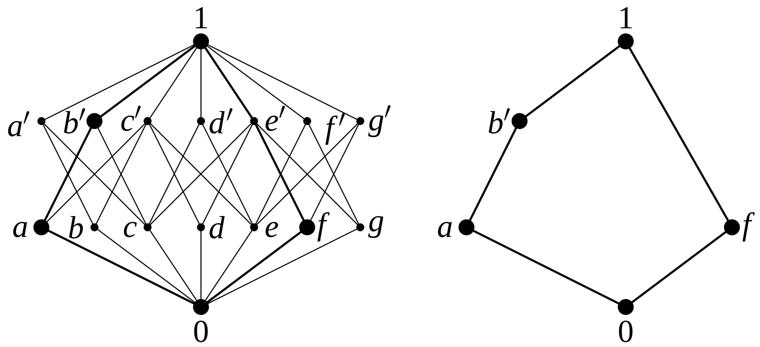


Figure 7.2: The set of nodes $\{0, a, b', 1, f\}$ forms a pentagon sublattice in the OML of Fig. 7.1, proving that the OML is non-modular.

For a subset of a lattice to be a sublattice, it must be closed under the parent lattice's operations $\wedge, \vee$ (although not necessarily closure under orthocomplement). To verify this is the case in Fig. 7.2, we can construct the truth-tables for the five sublattice elements:

$x \diagdown y$	$x\wedge y$: 0	1	a	b'	f
0	0	0	0	0	0
1	0	1	a	b'	f
a	0	a	a	a	0
b'	0	b'	a	b'	0
f	0	f	0	0	f

$x \diagdown y$	$x\vee y$: 0	1	a	b'	f
0	0	1	a	b'	f
1	1	1	1	1	1
a	a	1	a	b'	1
b'	b'	1	b'	b'	1
f	f	1	1	1	f

From this table we see that there is closure and that the ordering relations implied by the pentagonal sublattice in the right-hand side of Fig. 7.2 are satisfied.

It is important to note that not every embedded pentagon of nodes is a sublattice. For example, the pentagon shape $\{0,a,b',1,e'\}$ is not a sublattice: $b'\wedge e' = c \notin \{0,a,b',1,e'\}$, so the set of nodes is not closed under the $\wedge$ operation (see Fig. 7.3, which shows the six-node sublattice generated by the five nodes).

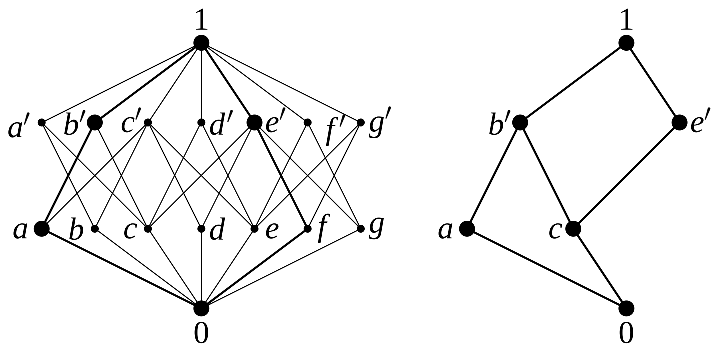


Figure 7.3: The pentagonal arrangement of nodes $\{0,a,b',1,e\}$ in the OML of Fig. 7.1 does not form a sublattice since it is not closed under the original lattice operations. On the right we show the six-node sublattice generated by the five nodes.

Moreover, closure alone does not guarantee a pentagon sublattice. For example, the pentagon shape $\{0,c,1,e',f\}$ is closed under the $\wedge$ and $\vee$ operations and thus is a sublattice, but it isn't a pentagonal sublattice, because $c\vee f = e' \neq 1$ (see Fig. 7.4).

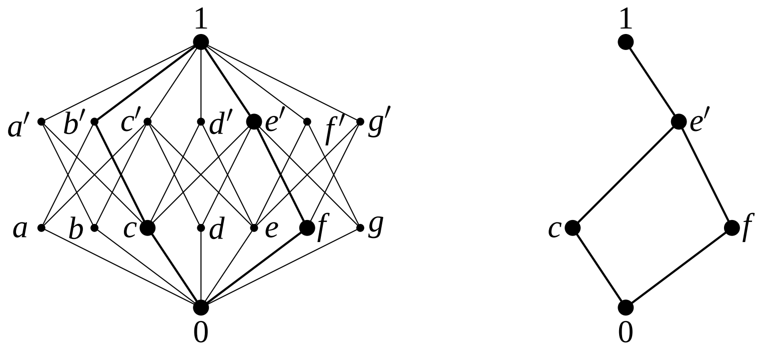


Figure 7.4: The pentagonal arrangement of nodes $\{0, c, 1, e', f\}$ in the OML of Fig. 7.1 forms a sublattice but not a pentagonal sublattice.

7.2.2 Consequences of the modular law

The following consequence of the modular law, apparently[3] due to von Neumann, is of special interest to us because an instance of it is the 3OA identity law in the form shown by Th. 4.5.8 (p. 72). In particular, some of the equations involved in its proof may suggest analogues holding in OML that could assist towards resolving the 3OA identity conjecture (p. 63).

Theorem 7.2.6. *The following condition holds in any* MOL *(and also in any* ML*):*

$$(a \vee b) \wedge (c \vee d) = 0 \quad \Rightarrow \quad (a \vee c) \wedge (b \vee d) = (a \wedge b) \vee (c \wedge d). \tag{7.12}$$

Proof. Ref. [49, Lemma 9, p. 96] gives a proof sketch, but since some details are omitted and the proof is not necessarily intuitive, we will give the full proof here. We will prove, in succession, the following steps:

$$((x \vee y) \vee u) \wedge w = ((x \vee y) \vee u) \wedge ((u \vee w) \wedge w) \tag{7.13}$$

$$(x \vee y \vee u) \wedge ((u \vee w) \wedge w) = (((x \vee y) \wedge (u \vee w)) \vee u) \wedge w \tag{7.14}$$

$$(x \vee y) \wedge (u \vee w) = 0$$

$$\Rightarrow \quad (((x \vee y) \wedge (u \vee w)) \vee u) \wedge w = u \wedge w \tag{7.15}$$

$$(x \vee y) \wedge (u \vee w) = 0$$

$$\Rightarrow \quad ((x \vee y) \vee u) \wedge w = u \wedge w \tag{7.16}$$

$$((x \vee y) \vee u) \wedge ((x \vee y) \vee w) = (x \vee y) \vee (((x \vee y) \vee u) \wedge w) \tag{7.17}$$

[3] Kalmbach [49, p. 96] states, above her Lemma 9, that it is due to von Neumann and cites Ref. [122] (with no page number given). However, this author was unable to find this theorem in Ref. [122].

$$\begin{aligned}
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad ((a \vee b) \vee d) \wedge ((b \vee c) \vee d) = (c \wedge a) \vee (b \vee d) && (7.18) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad (c \wedge a) \vee (b \vee d) = b \vee d && (7.19) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad ((a \vee b) \vee d) \wedge ((b \vee c) \vee d) = b \vee d && (7.20) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad ((a \vee b) \vee d) \wedge ((a \vee b) \vee c) = (c \wedge d) \vee (a \vee b) && (7.21) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad ((a \vee b) \vee c) \wedge ((a \vee c) \vee d) = a \vee c && (7.22) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad ((b \vee c) \vee d) \wedge ((a \vee c) \vee d) = (c \vee d) \vee (a \wedge b) && (7.23) \\
&((c \wedge d) \vee (a \vee b)) \wedge ((c \vee d) \vee (a \wedge b)) \\
&\qquad\qquad = (c \wedge d) \vee ((a \vee b) \wedge ((c \vee d) \vee (a \wedge b))) && (7.24) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad (c \wedge d) \vee ((a \vee b) \wedge ((c \vee d) \vee (a \wedge b))) = (c \wedge d) \vee (a \wedge b) && (7.25) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad ((c \wedge d) \vee (a \vee b)) \wedge ((c \vee d) \vee (a \wedge b)) = (c \wedge d) \vee (a \wedge b) && (7.26) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad (a \vee c) \wedge (b \vee d) \\
&\qquad\quad = (((a \vee b) \vee c) \wedge ((a \vee c) \vee d)) \wedge (((a \vee b) \vee d) \wedge ((b \vee c) \vee d)) && (7.27) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad (((a \vee b) \vee c) \wedge ((a \vee c) \vee d)) \wedge (((a \vee b) \vee d) \wedge ((b \vee c) \vee d)) \\
&\qquad\qquad = (a \wedge b) \vee (c \wedge d) && (7.28) \\
&(a \vee b) \wedge (c \vee d) = 0 \\
&\qquad \Rightarrow \quad (a \vee c) \wedge (b \vee d) = (a \wedge b) \vee (c \wedge d) && (7.29)
\end{aligned}$$

In the following derivations, we show all applications of the modular law explicitly. All other inferences hold in any lattice. "Rearrange" means apply commutative and associative laws.

For Eq. (7.13): Conjoin $x \vee y \vee u$ to both sides of the lattice absorption law $w = (u \vee w) \wedge w$.

For Eq. (7.14): An instance of the modular law Eq. (7.4) gives $(u \vee (x \vee y)) \wedge (u \vee w) =$

$u\vee((x\vee y)\wedge(u\vee w))$. Rearranging terms, $((x\vee y)\vee u)\wedge(u\vee w)=((x\vee y)\wedge(u\vee w))\vee u$. Conjoining w to both sides and again rearranging terms yields the result.

For Eq. (7.15): Disjoining u to both sides of the hypothesis, $((x\vee y)\wedge(u\vee w))\vee u=0\vee u=u$. Conjoining w to both sides yields the result.

For Eq. (7.16): Chain Eqs. (7.13), (7.14), and (7.15).

For Eq. (7.17): An instance of the modular law Eq. (7.4) gives $(x\vee y)\vee(w\wedge((x\vee y)\vee u))=((x\vee y)\vee w)\wedge((x\vee y)\vee u)$. Swap the two sides and rearrange.

For Eq. (7.18): For one direction, $(c\wedge a)\vee b\vee d\leq a\vee b\vee d$ and $(c\wedge a)\vee b\vee d\leq b\vee c\vee d$, so $(c\wedge a)\vee(b\vee d)\leq((a\vee b)\vee d)\wedge((b\vee c)\vee d)$. For the other direction, $((a\vee b)\vee d)\wedge((b\vee c)\vee d)=((b\vee d)\vee a)\wedge((b\vee d)\vee c)=(b\vee d)\vee(((b\vee d)\vee a)\wedge c)$ (i) by Eq. (7.17). From the hypothesis, $(b\vee a)\wedge(d\vee c)=0$ so $((b\vee d)\vee a)\wedge c=((b\vee a)\vee d)\wedge c=d\wedge c$ by Eq. (7.17); $d\wedge c\leq b\vee d\leq(c\wedge a)\vee(b\vee d)$, so $((b\vee d)\vee a)\wedge c\leq(c\wedge a)\vee(b\vee d)$ (ii). Also, $b\vee d\leq(c\wedge a)\vee(b\vee d)$, and combining with (ii) gives $(b\vee d)\vee(((b\vee d)\vee a)\wedge c)\leq(c\wedge a)\vee(b\vee d)$ (iii). Chaining (i) and (iii) gives $((a\vee b)\vee d)\wedge((b\vee c)\vee d)\leq(c\wedge a)\vee(b\vee d)$.

For Eq. (7.19): $c\wedge a\leq(a\vee b)\wedge(c\vee d)=0$ by the hypothesis, so $c\wedge a=0$. Disjoin $b\vee d$ to both sides.

For Eq. (7.20): Chain Eqs. (7.18) and (7.19).

For Eq. (7.21): Rearranging Eq. (7.17), $((a\vee b)\vee d)\wedge((a\vee b)\vee c)=(((a\vee b)\vee c)\wedge d)\vee(a\vee b)$. Disjoining $a\vee b$ to both sides of Eq. (7.16), $(((a\vee b)\vee c)\wedge d)\vee(a\vee b)=(c\wedge d)\vee(a\vee b)$. Chaining these yields the result.

For Eq. (7.22): Rearrange the left-hand side of Eq. (7.20).

For Eq. (7.23): Rearrange the sides of Eq. (7.21).

For Eq. (7.24): An instance of the modular law Eq. (7.4) gives $(c\wedge d)\vee((a\vee b)\wedge((c\wedge d)\vee((c\vee d)\vee(a\wedge b))))=((c\wedge d)\vee(a\vee b))\wedge((c\wedge d)\vee((c\vee d)\vee(a\wedge b)))$. Using $(c\wedge d)\vee(c\vee d)=c\vee d$, rearranging the right-hand side, and swapping sides yields the result.

For Eq. (7.25): From $a\wedge b=(a\vee b)\wedge(a\wedge b)$ we get $(a\vee b)\wedge((c\vee d)\vee(a\wedge b))=(a\vee b)\wedge((c\vee d)\vee((a\vee b)\wedge(a\wedge b)))$. An instance of the modular law Eq. (7.3) gives $(a\vee b)\wedge((c\vee d)\vee((a\vee b)\wedge(a\wedge b)))=((a\vee b)\wedge(c\vee d))\vee((a\vee b)\wedge(a\wedge b))$. Disjoining the hypothesis with both sides of $(a\vee b)\wedge(a\wedge b)=a\wedge b$, we get $((a\vee b)\wedge(c\vee d))\vee((a\vee b)\wedge(a\wedge b))=a\wedge b$. Chaining these three gives $(a\vee b)\wedge((c\vee d)\vee(a\wedge b))=a\wedge b$. Disjoining $c\wedge d$ to both sides yields the result.

For Eq. (7.26): Chain Eqs. (7.24) and (7.25).

For Eq. (7.27): Conjoin the sides of Eqs. (7.20) and (7.22) to get $(((a\vee b)\vee c)\wedge((a\vee c)\vee d))\wedge(((a\vee b)\vee d)\wedge((b\vee c)\vee d))=(a\vee c)\wedge(b\vee d)$, then swap the sides.

For Eq. (7.28): Conjoin the sides of Eqs. (7.21) and (7.23) then rearrange the left-hand side to get $(((a \vee b) \vee c) \wedge ((a \vee c) \vee d)) \wedge (((a \vee b) \vee d) \wedge ((b \vee c) \vee d)) = ((c \wedge d) \vee (a \vee b)) \wedge ((c \vee d) \vee (a \wedge b))$. Rearrange the right-hand side of Eq. (7.26) to get $((c \wedge d) \vee (a \vee b)) \wedge ((c \vee d) \vee (a \wedge b)) = (a \wedge b) \vee (c \wedge d)$. Chaining these two yields the result.

For Eq. (7.29): Chain Eqs. (7.27) and (7.28). ☐

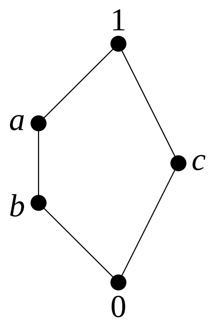


Figure 7.5: The pentagon lattice N_5.

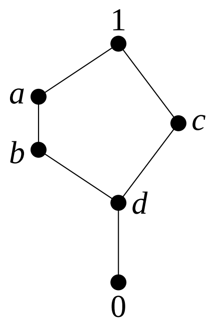


Figure 7.6: Counterexample for Th. 7.2.7.

However, the converse does not hold i.e. we cannot derive the modular law from the above condition added to Lat (the class of lattices).

Theorem 7.2.7. *The modular law consequence Eq. (7.12), when added to the equations for a lattice, is strictly weaker than the modular law.*

Proof. The lattice of Fig. 7.6 is non-modular, as can be shown by direct evaluation of the modular law Eq. (7.1) or by noticing that that it includes a pentagonal sublattice. On the other hand, it satisfies Eq. (7.12). ☐

It is tempting to think that the pentagon lattice N_5 [Fig. 7.5 (p. 121)] characterizes not only modular lattices but also the modular law, in a manner similar to how lattice O_6 characterizes the orthomodular law. The above result shows that this is not the case. The following theorem formalizes this.

Theorem 7.2.8. *In the presence of a lattice (member of class* Lat*), it is possible for a condition (equational inference) strictly weaker than the modular law to fail in lattice N_5. Therefore, lattice N_5 does not characterize conditions equivalent to the modular law in a lattice.*

Proof. Th. 7.2.7 shows that Eq. (7.12) is a condition strictly weaker than the modular law in the presence of a lattice. However, this condition fails in lattice N_5. □

It is apparently an open problem whether Th. 7.2.8 holds in the presence of an ortholattice, i.e. whether or not the addition of a condition that fails in N_5 will strengthen the OL laws to become the MOL laws. In particular, it is unknown whether Eq. (7.12) is equivalent to the modular law in the presence of an OL.

We can, however, derive the OML law from Eq. (7.12) in the presence of an OL.

Theorem 7.2.9. *In any* OL*, Eq. (7.12) implies the* OML *law.*

Proof. Substitute x' for a, y' for b, x for c, and 0 for d in Eq. (7.12). This results in the inference $(x' \vee y') \wedge x = 0 \Rightarrow (x \vee x') \wedge (y' \vee 0) = (x' \wedge y') \vee (x \wedge 0)$. Applying DeMorgan's laws, this is equivalent to $x \vee (x' \wedge y' = 1 \Rightarrow y = x \vee y$, so $x \rightarrow y = 1 \Rightarrow x \leq y$ by Def. 2.2.5 (p. 18). This is the OML law by Eq. (3.6) for $i = 1$.. □

Th. 4.5.8 (p. 72) showed that the 3OA identity law is a special case of Eq. (7.12), so it is possible that results about Eq. (7.12) could prove useful for proving or disproving the 3OA identity conjecture. However, the 3OA identity conjecture presupposes the equations for an OML. This provides additional motivation to prove or disprove Th. 7.2.8 in the presence of an OML (or equivalently, by Th. 7.2.9, an OL).

7.3 Arguesian lattices

Definition 7.3.1. *A lattice in which the following condition holds is an* **Arguesian lattice** (AL) *[17]:*

$$
\begin{aligned}
&(a \vee b) \wedge (c \vee d) \wedge (e \vee f) \\
&\quad \leq b \vee (a \wedge (c \vee (((a \vee c) \wedge (b \vee d)) \\
&\qquad\qquad \wedge (((a \vee e) \wedge (b \vee f)) \vee ((c \vee e) \wedge (d \vee f))))))
\end{aligned} \tag{7.30}
$$

The following theorem lists all of the known equivalent forms of the Arguesian law that have appeared in the literature (to this author's knowledge). These are often shown using abbreviations for some of the subformulas, but it is also useful to show them fully expanded, as we do: their sizes and some aspects of their structures are easier to compare, and it can be easier to encode them for a computer checking program. The reader who wishes to see the more compact forms can consult the original references. Recall that the dual of an equation has $\vee$ replaced with $\wedge$ and vice versa, and $\leq$ replaced with $\geq$, but $\Rightarrow$ (logical implication between hypothesis and conclusion) is unaffected.

Theorem 7.3.2. *A lattice in which any of the following condition (or its dual [47]) holds is an* AL*:*

$$
\begin{aligned}
&c \wedge (((a \vee d) \wedge (b \vee e)) \vee f) \\
&\qquad \leq a \vee ((((a \vee b) \wedge (d \vee e)) \vee ((b \vee c) \wedge (e \vee f))) \wedge (d \vee f))
\end{aligned} \tag{7.31}
$$

$$
\begin{aligned}
&(a \vee b) \wedge (c \vee d) \leq (e \vee f) \\
&\qquad \Rightarrow \quad (a \vee c) \wedge (b \vee d) \leq ((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f))
\end{aligned} \tag{7.32}
$$

$$
\begin{aligned}
&(a \vee b) \wedge (c \vee d) \wedge (e \vee f) \\
&\quad \leq (a \wedge (c \vee (((a \vee c) \wedge (b \vee d)) \\
&\qquad\qquad \wedge (((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f)))))) \\
&\qquad \vee (b \wedge (d \vee (((a \vee c) \wedge (b \vee d)) \\
&\qquad\qquad \wedge (((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f))))))
\end{aligned} \tag{7.33}
$$

$$
\begin{aligned}
&(a \vee b) \wedge (c \vee d) \wedge (e \vee f) \\
&\quad \leq a \vee (b \wedge (d \vee (((a \vee c) \wedge (b \vee d)) \\
&\qquad\qquad \wedge (((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f))))))
\end{aligned} \tag{7.34}
$$

$$
(a \vee b) \wedge (c \vee d) \wedge (e \vee f)
$$

$$\leq a \vee d \vee (((a \vee c) \wedge (b \vee d)) \\ \wedge (((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f)))) \tag{7.35}$$

$$(a \vee c) \wedge ((b \wedge (a \vee ((a \vee b) \wedge (c \vee d) \wedge (e \vee f)))) \vee d) \\ \leq ((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f)) \vee (d \wedge (a \vee c)) \tag{7.36}$$

$$(a \vee c) \wedge ((b \wedge (a \vee ((c \vee d) \wedge (e \vee f)))) \vee d) \\ \leq ((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f)) \vee (d \wedge (a \vee c)) \tag{7.37}$$

$$k \wedge ((((a \wedge b) \vee (c \wedge d)) \wedge ((e \wedge f) \vee (g \wedge h))) \vee (m \wedge j)) \\ \leq a \vee ((((k \vee f) \wedge (m \vee h)) \vee ((e \vee b) \wedge (g \vee d))) \wedge (c \vee j)) \tag{7.38}$$

$$(d \vee f) \wedge ((c \wedge (f \vee ((a \vee d) \wedge (b \vee e)))) \vee a) \\ \leq ((a \vee b) \wedge (d \vee e)) \vee ((b \vee c) \wedge (e \vee f)) \vee (a \wedge (d \vee f)) \tag{7.39}$$

$$(a \vee b) \wedge (c \vee d) \wedge (e \vee f) = (c \vee d) \wedge (e \vee f) \\ \wedge (a \wedge (c \vee (((a \vee c) \wedge (b \vee d)) \\ \wedge (((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f))))) \\ \vee (b \wedge (d \vee (((a \vee c) \wedge (b \vee d)) \\ \wedge (((c \vee e) \wedge (d \vee f)) \vee ((a \vee e) \wedge (b \vee f))))))) \tag{7.40}$$

$$(k \wedge ((((a \wedge b) \vee (c \wedge d)) \wedge ((e \wedge f) \vee (g \wedge h))) \vee (m \wedge j))) \\ \vee (a \wedge ((((k \wedge f) \vee (m \wedge h)) \wedge ((e \wedge b) \vee (g \wedge d))) \vee (c \wedge j))) \\ \leq (k \vee ((((a \vee b) \wedge (c \vee d)) \vee ((e \vee f) \wedge (g \vee h))) \wedge (m \vee j))) \\ \wedge (a \vee ((((k \vee f) \wedge (m \vee h)) \vee ((e \vee b) \wedge (g \vee d))) \wedge (c \vee j))) \tag{7.41}$$

Proof. For Eq. (7.31), see Ref. [33, Eq. (1), p. 167]. For Eq. (7.32), see Ref. [28] or Ref. [17, p. 67]. For Eq. (7.33), see Ref. [22, Eq. (2), p. 303] or Ref. [19, Th. 2.1(2), p. 337]. For Eq. (7.34), see Ref. [22, Eq. (3), p. 303] or Ref. [19, Th. 2.1(3), p. 337]. For Eq. (7.35), see Ref. [22, Eq. (4), p. 303], Ref. [19, Th. 2.1(4), p. 337], or Ref. [33, Eq. (4), p. 168]. For Eq. (7.36), see Ref. [19, Th. 2.1(5), p. 337]. For Eq. (7.37), see Ref. [22, Eq. (5), p. 303]. For Eq. (7.38), see Ref. [33, Eq. (2), p. 168]. For Eq. (7.39), see Ref. [33, Eq. (3), p. 168]. For Eq. (7.40), see Ref. [110, p. 4]. For Eq. (7.41), see Ref. [92]. ☐

Eq. (7.31) is the shortest known form of the Arguesian law. Eq. (7.41) shows that the Arguesian law can be expressed in a form which is self-dual.

To demonstrate the Arguesian law, we will consider the lattice formed by the projective subspaces of a 2-dimensional projective space (projective plane).

One way to construct a projective plane is using a height 3 Hilbert lattice, whose nodes are the subspaces of a 3-dimensional Hilbert space. We can define a **point** as the singleton of an atom.[4] By the axioms of projective geometry (e.g. [40, Sec. 3]; see also below), any two points (and thus any two atoms of the Hilbert lattice) determine a unique collection of atoms called a **line**. A **projective subspace** is a set of atoms such that the line determined by any two atoms in the set is included in the set. In the case of the lattice of subspaces of a projective plane (whether built from a Hilbert lattice or not), the only kinds of projective subspaces are the empty set (the lattice zero), points, lines, and the whole space (the lattice unit).

In the case of the lattice of closed subspaces of a 3-dimensional Hilbert space, the 1-dimensional and 2-dimensional subspaces correspond to the points and lines, respectively, of the projective lattice constructed from it.

We can also construct a projective plane by extending a Euclidean plane with points at infinity [15, p. 109]. The points are the singletons of the $\langle x, y\rangle$ coordinates, $x, y \in \mathbb{R}$, together with new points $\{\langle\infty,\infty\rangle\}$ and $\{\langle\infty, r\rangle\}$ for $r \in \mathbb{R}$. A non-vertical line consists of the Euclidean line together with the point $\{\langle\infty, r\rangle\}$, where r is the slope of the line, and a vertical line (parallel to the y-axis) consists of the Euclidean line together with the point $\{\langle\infty,\infty\rangle\}$. It can be verified that this construction satisfies the **axioms of a projective geometry** (two points determine one and only one line; every line contains at least three points; and if a line intersects two sides of a triangle at different points, then it intersects the third side). In addition, the Arguesian law holds (as well as the modular law which follows from the Arguesian law).

One instance of the Arguesian law in this extended Euclidean plane is illustrated in Fig. 7.7, where we have omitted the points at infinity for simplicity. We will use $\vee$ and $\wedge$ to denote the projective subspace sum (the union of all lines determined by a point from the first subspace and a point from the second) and meet (the set intersection of the two subspaces). Assume that the lines $a_0 \vee b_0$, $a_1 \vee b_1$, and $a_2 \vee b_2$ intersect at a common point d. Let c_0 be the point $(a_1 \vee a_2) \wedge (b_1 \vee b_2)$, c_1 the point $(a_0 \vee a_2) \wedge (b_0 \vee b_2)$, and c_2 the point $(a_0 \vee a_1) \wedge (b_0 \vee b_1)$. Then for the Arguesian law to hold, c_0, c_1, and c_2 must fall on the same line, which a detailed analysis using e.g. analytic geometry will show to be the case.

To show this with the Arguesian law, in we assign the points of Fig. 7.7 to Eq. (7.32) as follows: $a = a_0$, $b = b_0$, $c = a_1$, $d = b_1$, $e = a_2$, and $f = b_2$. The hypothesis of Eq. (7.32) is

[4] We define a point as a singleton of an atom, rather than the atom itself as usual in the literature. We do this because we can use inclusion as the sole ordering relation on projective subspaces, rather that the traditional but awkward context-dependent mixture of membership and inclusion, making a formal development easier.

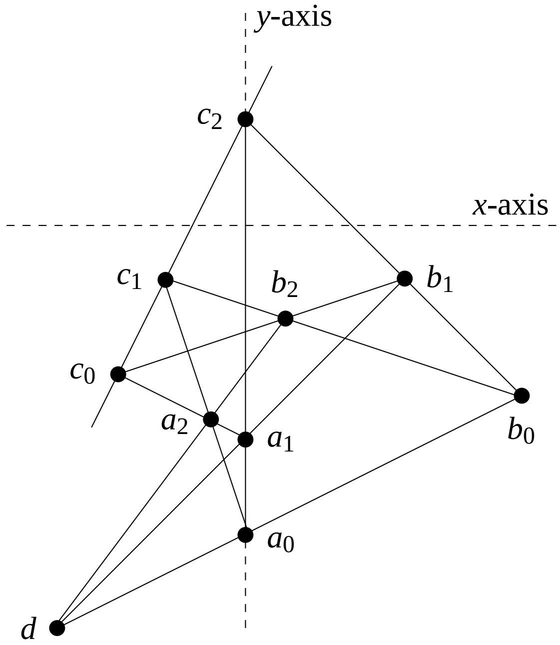


Figure 7.7: Example of the Arguesian law in the projective subspace lattice of a projective plane built from an extended Euclidean plane. (Note: this is not a Hasse lattice diagram. The lines represent projective subspaces generated by points; see text.)

satisfied:

$$\begin{aligned}(a_0 \vee b_0) \wedge (a_1 \vee b_1) = d& \\ &\leq (a_2 \vee b_2).\end{aligned}$$

Evaluating the conclusion,

$$\begin{aligned}(a_0 \vee a_1) \wedge (b_0 \vee b_1) &= c_2 \\ &\leq c_2 \vee c_0 \vee c_1 \\ &= c_0 \vee c_1 \\ &= ((a_1 \vee a_2) \wedge (b_1 \vee b_2)) \vee ((a_0 \vee a_2) \wedge (b_0 \vee b_2)),\end{aligned}$$

showing that this instance satisfies the Arguesian law. The penultimate equality follows because c_2 is on the same line as c_0 and c_1.

We will now modify the above example slightly to construct a projective plane in which the Arguesian law fails, but the modular law still holds. Our construction is a slight modification of the non-Arguesian projective geometry known as the "Moulton plane" [87]. For convenient reference, we name the lines as follows.

$$l_0 = a_0 \vee b_0 \tag{7.42}$$
$$l_1 = a_1 \vee b_1 \tag{7.43}$$
$$l_2 = a_2 \vee b_2 \tag{7.44}$$
$$l_3 = a_0 \vee a_1 \tag{7.45}$$
$$l_4 = a_0 \vee a_2 \tag{7.46}$$
$$l_5 = a_1 \vee a_2 \tag{7.47}$$
$$l_6 = b_0 \vee b_1 \tag{7.48}$$
$$l_7 = b_0 \vee b_2 \tag{7.49}$$
$$l_8 = b_1 \vee b_2 \tag{7.50}$$
$$l_c = c_0 \vee c_1 \tag{7.51}$$

Refer to Fig. 7.8.

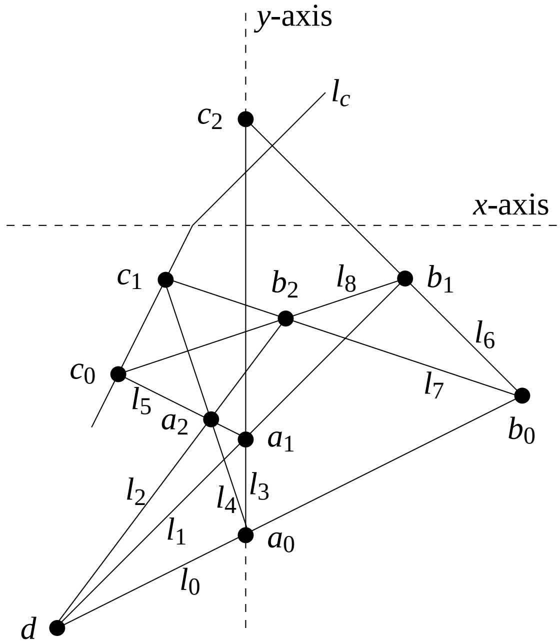


Figure 7.8: A modular, non-Arguesian projective subspace lattice of a projective plane built from an extended Euclidean plane.

The construction is the same as the one in Fig. 7.7 but with the following modification: any line with positive slope is bent at the x-axis so that it has slope r below the x-axis and slope $r/2$ above the x-axis. (Its point at infinity is not modified but continues to be $\{\langle\infty, r\rangle\}$.) Even with this "defect," it can be shown that the modified construction continues to be a projective plane [15, p. 110] and thus the modular law holds (since it holds in any projective plane [2, Th. IV, p. 259]). However, the Arguesian law fails: the point c_2 does not fall on the line determined by c_0 and c_1. Working out the assignment in the same way as we did for the previous example (with more detail, since we will be interested in what projective subspaces are visited), the hypothesis of Eq. (7.32) is satisfied exactly as before:

$$\begin{aligned}(a_0 \vee b_0) \wedge (a_1 \vee b_1) &= l_0 \wedge l_1 \\ &= d \\ &= d \wedge l_2 \quad (\text{i.e. } \leq l_2) \\ &= d \wedge (a_2 \vee b_2) \quad (\text{i.e. } \leq a_2 \vee b_2). \end{aligned} \tag{7.52}$$

Above we used the equivalence $a \leq b \Leftrightarrow a = a \wedge b$, which holds in any lattice. On the other hand, the conclusion evaluates as follows:

$$\begin{aligned}(a_0 \vee a_1) \wedge (b_0 \vee b_1) &= l_3 \wedge l_6 \\ &= c_2 \\ &\neq c_2 \wedge l_c = 0 \quad (\text{i.e. } \not\leq l_c) \\ \text{where } l_c &= c_0 \vee c_1 \\ &= (l_5 \wedge l_8) \vee (l_4 \wedge l_7) \\ &= ((a_1 \vee a_2) \wedge (b_1 \vee b_2)) \vee ((a_0 \vee a_2) \wedge (b_0 \vee b_2)), \end{aligned} \tag{7.53}$$

showing that the Arguesian law is violated by the assignment of Fig. 7.8.

Finite projective planes

The previous examples involved infinite projective geometries since the real number field used to construct the Euclidean plane has infinite members. Thus the lattice of their projective subspaces cannot be represented with a finite Hasse diagram.

However, there exist projective planes over finite fields. The smallest is the **Fano plane**, with 7 points and 7 lines, was discovered in 1892 by Gino Fano [123], and is shown in Fig. 7.9. The points and lines in the figure are labeled in order to see the correspondence to the Hasse diagram of its projective subspace lattice. This Hasse diagram is shown in Fig. 7.10. The lattice version of the projective plane is important for us because it makes automated verification of equations straightforward. This Hasse diagram, which appears in Ref. [116, p. 33, Fig. 1.18]

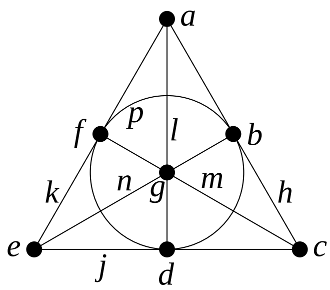


Figure 7.9: The Fano plane, which is the smallest non-trivial finite projective plane, with 7 points *a*–*g* and 7 lines (including the circle) h, j, k, l, m, n, p.

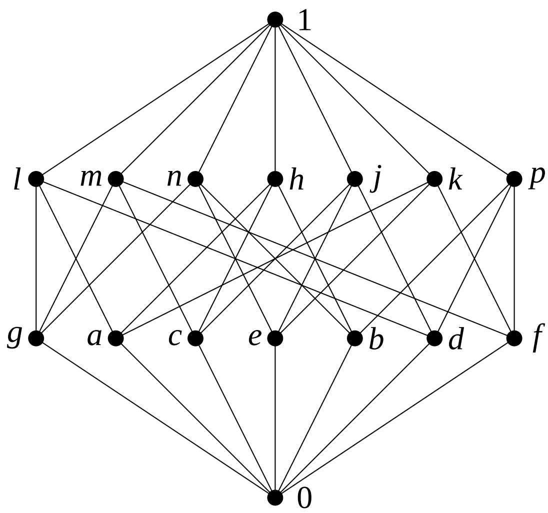


Figure 7.10: The projective subspace lattice of the Fano plane, with nodes labeled to correspond to the projective subspaces in Fig. 7.9. This lattice is modular and Arguesian.

(who calls it the "lattice of flats" for the Fano plane), is redrawn in our Fig. 7.10 to reveal an interesting symmetry: it remains the same when rotated by 180 degrees. The lattice is both modular and Arguesian and can be useful as a soundness check of, say, conjectured equivalents for these laws. It is also possible that it could serve as a component or starting point towards finding an Arguesian law counterexample.

Table 7.1: Covering table for the Hasse diagram of Fig. 7.10. Each node in the left-hand column is followed by the nodes that it covers.

$$
\begin{array}{l}
1 \gtrdot l,m,n,h,j,k,p \\
l \gtrdot g,a,d \\
m \gtrdot g,c,f \\
n \gtrdot g,e,b \\
h \gtrdot a,c,b \\
j \gtrdot c,e,d \\
k \gtrdot a,e,f \\
p \gtrdot b,d,f \\
g \gtrdot 0 \\
a \gtrdot 0 \\
c \gtrdot 0 \\
e \gtrdot 0 \\
b \gtrdot 0 \\
d \gtrdot 0 \\
f \gtrdot 0 \\
0 \gtrdot
\end{array}
$$

In Table 7.1 (p. 130) we show an alternate but equivalent representation of the Hasse diagram [Def. 2.5.1 (p. 22)], called a **covering table**, in which each lattice node is followed by a list of the nodes that it covers. Covering tables can provide a useful way to express the Hasse diagram in a machine-readable format. For our program `hasse.c`, table lines are separated by a semicolon, and the table ends with a period. Thus Table 7.1 would be expressed as "`1>l,m,n,h,j,k,p;l>g,a,d;m>g,c,f;n>g,e,b;h>a,c,b;j>c,e,d;k>a,e,f;-p>b,d,f;g>0;a>0;c>0;e>0;b>0;d>0;f>0;0>.`".

The smallest (finite) projective plane which is non-Arguesian (but modular, as all projective planes are [2, Th. IV, p. 259]) has 91 points and 91 lines. It was discovered in 1907 by Veblen and MacLagan-Wedderburn [121] [123]. Its projective subspace lattice thus has $91 \cdot 2 + 2 = 184$ nodes. Its Hasse diagram is too large and complex to be drawn in a meaningful way. Instead, we specify it with the covering table of Table 7.2.

Table 7.2: Covering table that specifies the 184-node non-Arguesian modular lattice corresponding to the projective subspaces of Velblen and MacLagan-Wedderburn's 91-point projective plane.

$1 \gtrdot l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}, l_{11}, l_{12}, l_{13}, l_{14}, l_{15}, l_{16}, l_{17}, l_{18}, l_{19}, l_{20}, l_{21}, l_{22}, l_{23}, l_{24}, l_{25}, l_{26}, l_{27}, l_{28}, l_{29}, l_{30}, l_{31}, l_{32}, l_{33}, l_{34}, l_{35}, l_{36}, l_{37}, l_{38}, l_{39}, l_{40}, l_{41}, l_{42}, l_{43}, l_{44}, l_{45}, l_{46}, l_{47}, l_{48}, l_{49}, l_{50}, l_{51}, l_{52}, l_{53}, l_{54}, l_{55}, l_{56}, l_{57}, l_{58}, l_{59}, l_{60}, l_{61}, l_{62}, l_{63}, l_{64}, l_{65}, l_{66}, l_{67}, l_{68}, l_{69}, l_{70}, l_{71}, l_{72}, l_{73}, l_{74}, l_{75}, l_{76}, l_{77}, l_{78}, l_{79}, l_{80}, l_{81}, l_{82}, l_{83}, l_{84}, l_{85}, l_{86}, l_{87}, l_{88}, l_{89}, l_{90}, l_{91}$

$l_1 \gtrdot a_0, a_1, a_3, a_9, b_0, c_0, d_0, e_0, f_0, g_0$

$l_2 \gtrdot a_0, b_1, b_8, d_3, d_{11}, e_2, e_5, e_6, g_7, g_9$

$l_3 \gtrdot a_0, c_1, c_8, e_7, e_9, f_3, f_{11}, g_2, g_5, g_6$

$l_4 \gtrdot a_0, b_7, b_9, d_1, d_8, f_2, f_5, f_6, g_3, g_{11}$

$l_5 \gtrdot a_0, b_2, b_5, b_6, c_3, c_{11}, e_1, e_8, f_7, f_9$

$l_6 \gtrdot a_0, c_7, c_9, d_2, d_5, d_6, e_3, e_{11}, f_1, f_8$

$l_7 \gtrdot a_0, b_3, b_{11}, c_2, c_5, c_6, d_7, d_9, g_1, g_8$

$l_8 \gtrdot a_1, a_2, a_4, a_{10}, b_1, c_1, d_1, e_1, f_1, g_1$

$l_9 \gtrdot a_1, b_2, b_9, d_4, d_{12}, e_3, e_6, e_7, g_8, g_{10}$

$l_{10} \gtrdot a_1, c_2, c_9, e_8, e_{10}, f_4, f_{12}, g_3, g_6, g_7$

$l_{11} \gtrdot a_1, b_8, b_{10}, d_2, d_9, f_3, f_6, f_7, g_4, g_{12}$

$l_{12} \gtrdot a_1, b_3, b_6, b_7, c_4, c_{12}, e_2, e_9, f_8, f_{10}$

$l_{13} \gtrdot a_1, c_8, c_{10}, d_3, d_6, d_7, e_4, e_{12}, f_2, f_9$

$l_{14} \gtrdot a_1, b_4, b_{12}, c_3, c_6, c_7, d_8, d_{10}, g_2, g_9$

$l_{15} \gtrdot a_2, a_3, a_5, a_{11}, b_2, c_2, d_2, e_2, f_2, g_2$

$l_{16} \gtrdot a_2, b_3, b_{10}, d_5, d_0, e_4, e_7, e_8, g_9, g_{11}$

$l_{17} \gtrdot a_2, c_3, c_{10}, e_9, e_{11}, f_5, f_0, g_4, g_7, g_8$

$l_{18} \gtrdot a_2, b_9, b_{11}, d_3, d_{10}, f_4, f_7, f_8, g_5, g_0$

$l_{19} \gtrdot a_2, b_4, b_7, b_8, c_5, c_0, e_3, e_{10}, f_9, f_{11}$

$l_{20} \gtrdot a_2, c_9, c_{11}, d_4, d_7, d_8, e_5, e_0, f_3, f_{10}$

$l_{21} \gtrdot a_2, b_5, b_0, c_4, c_7, c_8, d_9, d_{11}, g_3, g_{10}$

$l_{22} \gtrdot a_3, a_4, a_6, a_{12}, b_3, c_3, d_3, e_3, f_3, g_3$

$l_{23} \gtrdot a_3, b_4, b_{11}, d_6, d_1, e_5, e_8, e_9, g_{10}, g_{12}$

Continued on next page...

Table 7.2 – Continued

$l_{24} \geqslant a_3, c_4, c_{11}, e_{10}, e_{12}, f_6, f_1, g_5, g_8, g_9$
$l_{25} \geqslant a_3, b_{10}, b_{12}, d_4, d_{11}, f_5, f_8, f_9, g_6, g_1$
$l_{26} \geqslant a_3, b_5, b_8, b_9, c_6, c_1, e_4, e_{11}, f_{10}, f_{12}$
$l_{27} \geqslant a_3, c_{10}, c_{12}, d_5, d_8, d_9, e_6, e_1, f_4, f_{11}$
$l_{28} \geqslant a_3, b_6, b_1, c_5, c_8, c_9, d_{10}, d_{12}, g_4, g_{11}$
$l_{29} \geqslant a_4, a_5, a_7, a_0, b_4, c_4, d_4, e_4, f_4, g_4$
$l_{30} \geqslant a_4, b_5, b_{12}, d_7, d_2, e_6, e_9, e_{10}, g_{11}, g_0$
$l_{31} \geqslant a_4, c_5, c_{12}, e_{11}, e_0, f_7, f_2, g_6, g_9, g_{10}$
$l_{32} \geqslant a_4, b_{11}, b_0, d_5, d_{12}, f_6, f_9, f_{10}, g_7, g_2$
$l_{33} \geqslant a_4, b_6, b_9, b_{10}, c_7, c_2, e_5, e_{12}, f_{11}, f_0$
$l_{34} \geqslant a_4, c_{11}, c_0, d_6, d_9, d_{10}, e_7, e_2, f_5, f_{12}$
$l_{35} \geqslant a_4, b_7, b_2, c_6, c_9, c_{10}, d_{11}, d_0, g_5, g_{12}$
$l_{36} \geqslant a_5, a_6, a_8, a_1, b_5, c_5, d_5, e_5, f_5, g_5$
$l_{37} \geqslant a_5, b_6, b_0, d_8, d_3, e_7, e_{10}, e_{11}, g_{12}, g_1$
$l_{38} \geqslant a_5, c_6, c_0, e_{12}, e_1, f_8, f_3, g_7, g_{10}, g_{11}$
$l_{39} \geqslant a_5, b_{12}, b_1, d_6, d_0, f_7, f_{10}, f_{11}, g_8, g_3$
$l_{40} \geqslant a_5, b_7, b_{10}, b_{11}, c_8, c_3, e_6, e_0, f_{12}, f_1$
$l_{41} \geqslant a_5, c_{12}, c_1, d_7, d_{10}, d_{11}, e_8, e_3, f_6, f_0$
$l_{42} \geqslant a_5, b_8, b_3, c_7, c_{10}, c_{11}, d_{12}, d_1, g_6, g_0$
$l_{43} \geqslant a_6, a_7, a_9, a_2, b_6, c_6, d_6, e_6, f_6, g_6$
$l_{44} \geqslant a_6, b_7, b_1, d_9, d_4, e_8, e_{11}, e_{12}, g_0, g_2$
$l_{45} \geqslant a_6, c_7, c_1, e_0, e_2, f_9, f_4, g_8, g_{11}, g_{12}$
$l_{46} \geqslant a_6, b_0, b_2, d_7, d_1, f_8, f_{11}, f_{12}, g_9, g_4$
$l_{47} \geqslant a_6, b_8, b_{11}, b_{12}, c_9, c_4, e_7, e_1, f_0, f_2$
$l_{48} \geqslant a_6, c_0, c_2, d_8, d_{11}, d_{12}, e_9, e_4, f_7, f_1$
$l_{49} \geqslant a_6, b_9, b_4, c_8, c_{11}, c_{12}, d_0, d_2, g_7, g_1$
$l_{50} \geqslant a_7, a_8, a_{10}, a_3, b_7, c_7, d_7, e_7, f_7, g_7$
$l_{51} \geqslant a_7, b_8, b_2, d_{10}, d_5, e_9, e_{12}, e_0, g_1, g_3$
$l_{52} \geqslant a_7, c_8, c_2, e_1, e_3, f_{10}, f_5, g_9, g_{12}, g_0$
$l_{53} \geqslant a_7, b_1, b_3, d_8, d_2, f_9, f_{12}, f_0, g_{10}, g_5$
$l_{54} \geqslant a_7, b_9, b_{12}, b_0, c_{10}, c_5, e_8, e_2, f_1, f_3$
$l_{55} \geqslant a_7, c_1, c_3, d_9, d_{12}, d_0, e_{10}, e_5, f_8, f_2$

Continued on next page…

Table 7.2 – Continued

$l_{56} > a_7, b_{10}, b_5, c_9, c_{12}, c_0, d_1, d_3, g_8, g_2$
$l_{57} > a_8, a_9, a_{11}, a_4, b_8, c_8, d_8, e_8, f_8, g_8$
$l_{58} > a_8, b_9, b_3, d_{11}, d_6, e_{10}, e_0, e_1, g_2, g_4$
$l_{59} > a_8, c_9, c_3, e_2, e_4, f_{11}, f_6, g_{10}, g_0, g_1$
$l_{60} > a_8, b_2, b_4, d_9, d_3, f_{10}, f_0, f_1, g_{11}, g_6$
$l_{61} > a_8, b_{10}, b_0, b_1, c_{11}, c_6, e_9, e_3, f_2, f_4$
$l_{62} > a_8, c_2, c_4, d_{10}, d_0, d_1, e_{11}, e_6, f_9, f_3$
$l_{63} > a_8, b_{11}, b_6, c_{10}, c_0, c_1, d_2, d_4, g_9, g_3$
$l_{64} > a_9, a_{10}, a_{12}, a_5, b_9, c_9, d_9, e_9, f_9, g_9$
$l_{65} > a_9, b_{10}, b_4, d_{12}, d_7, e_{11}, e_1, e_2, g_3, g_5$
$l_{66} > a_9, c_{10}, c_4, e_3, e_5, f_{12}, f_7, g_{11}, g_1, g_2$
$l_{67} > a_9, b_3, b_5, d_{10}, d_4, f_{11}, f_1, f_2, g_{12}, g_7$
$l_{68} > a_9, b_{11}, b_1, b_2, c_{12}, c_7, e_{10}, e_4, f_3, f_5$
$l_{69} > a_9, c_3, c_5, d_{11}, d_1, d_2, e_{12}, e_7, f_{10}, f_4$
$l_{70} > a_9, b_{12}, b_7, c_{11}, c_1, c_2, d_3, d_5, g_{10}, g_4$
$l_{71} > a_{10}, a_{11}, a_0, a_6, b_{10}, c_{10}, d_{10}, e_{10}, f_{10}, g_{10}$
$l_{72} > a_{10}, b_{11}, b_5, d_0, d_8, e_{12}, e_2, e_3, g_4, g_6$
$l_{73} > a_{10}, c_{11}, c_5, e_4, e_6, f_0, f_8, g_{12}, g_2, g_3$
$l_{74} > a_{10}, b_4, b_6, d_{11}, d_5, f_{12}, f_2, f_3, g_0, g_8$
$l_{75} > a_{10}, b_{12}, b_2, b_3, c_0, c_8, e_{11}, e_5, f_4, f_6$
$l_{76} > a_{10}, c_4, c_6, d_{12}, d_2, d_3, e_0, e_8, f_{11}, f_5$
$l_{77} > a_{10}, b_0, b_8, c_{12}, c_2, c_3, d_4, d_6, g_{11}, g_5$
$l_{78} > a_{11}, a_{12}, a_1, a_7, b_{11}, c_{11}, d_{11}, e_{11}, f_{11}, g_{11}$
$l_{79} > a_{11}, b_{12}, b_6, d_1, d_9, e_0, e_3, e_4, g_5, g_7$
$l_{80} > a_{11}, c_{12}, c_6, e_5, e_7, f_1, f_9, g_0, g_3, g_4$
$l_{81} > a_{11}, b_5, b_7, d_{12}, d_6, f_0, f_3, f_4, g_1, g_9$
$l_{82} > a_{11}, b_0, b_3, b_4, c_1, c_9, e_{12}, e_6, f_5, f_7$
$l_{83} > a_{11}, c_5, c_7, d_0, d_3, d_4, e_1, e_9, f_{12}, f_6$
$l_{84} > a_{11}, b_1, b_9, c_0, c_3, c_4, d_5, d_7, g_{12}, g_6$
$l_{85} > a_{12}, a_0, a_2, a_8, b_{12}, c_{12}, d_{12}, e_{12}, f_{12}, g_{12}$
$l_{86} > a_{12}, b_0, b_7, d_2, d_{10}, e_1, e_4, e_5, g_6, g_8$
$l_{87} > a_{12}, c_0, c_7, e_6, e_8, f_2, f_{10}, g_1, g_4, g_5$

Continued on next page. . .

Table 7.2 – Continued

$l_{88} \gtrdot a_{12}, b_6, b_8, d_0, d_7, f_1, f_4, f_5, g_2, g_{10}$
$l_{89} \gtrdot a_{12}, b_1, b_4, b_5, c_2, c_{10}, e_0, e_7, f_6, f_8$
$l_{90} \gtrdot a_{12}, c_6, c_8, d_1, d_4, d_5, e_2, e_{10}, f_0, f_7$
$l_{91} \gtrdot a_{12}, b_2, b_{10}, c_1, c_4, c_5, d_6, d_8, g_0, g_7$
$a_0 \gtrdot 0;\ b_0 \gtrdot 0;\ c_0 \gtrdot 0;\ d_0 \gtrdot 0;\ e_0 \gtrdot 0;\ f_0 \gtrdot 0;\ g_0 \gtrdot 0$
$a_1 \gtrdot 0;\ b_1 \gtrdot 0;\ c_1 \gtrdot 0;\ d_1 \gtrdot 0;\ e_1 \gtrdot 0;\ f_1 \gtrdot 0;\ g_1 \gtrdot 0$
$a_2 \gtrdot 0;\ b_2 \gtrdot 0;\ c_2 \gtrdot 0;\ d_2 \gtrdot 0;\ e_2 \gtrdot 0;\ f_2 \gtrdot 0;\ g_2 \gtrdot 0$
$a_3 \gtrdot 0;\ b_3 \gtrdot 0;\ c_3 \gtrdot 0;\ d_3 \gtrdot 0;\ e_3 \gtrdot 0;\ f_3 \gtrdot 0;\ g_3 \gtrdot 0$
$a_4 \gtrdot 0;\ b_4 \gtrdot 0;\ c_4 \gtrdot 0;\ d_4 \gtrdot 0;\ e_4 \gtrdot 0;\ f_4 \gtrdot 0;\ g_4 \gtrdot 0$
$a_5 \gtrdot 0;\ b_5 \gtrdot 0;\ c_5 \gtrdot 0;\ d_5 \gtrdot 0;\ e_5 \gtrdot 0;\ f_5 \gtrdot 0;\ g_5 \gtrdot 0$
$a_6 \gtrdot 0;\ b_6 \gtrdot 0;\ c_6 \gtrdot 0;\ d_6 \gtrdot 0;\ e_6 \gtrdot 0;\ f_6 \gtrdot 0;\ g_6 \gtrdot 0$
$a_7 \gtrdot 0;\ b_7 \gtrdot 0;\ c_7 \gtrdot 0;\ d_7 \gtrdot 0;\ e_7 \gtrdot 0;\ f_7 \gtrdot 0;\ g_7 \gtrdot 0$
$a_8 \gtrdot 0;\ b_8 \gtrdot 0;\ c_8 \gtrdot 0;\ d_8 \gtrdot 0;\ e_8 \gtrdot 0;\ f_8 \gtrdot 0;\ g_8 \gtrdot 0$
$a_9 \gtrdot 0;\ b_9 \gtrdot 0;\ c_9 \gtrdot 0;\ d_9 \gtrdot 0;\ e_9 \gtrdot 0;\ f_9 \gtrdot 0;\ g_9 \gtrdot 0$
$a_{10} \gtrdot 0;\ b_{10} \gtrdot 0;\ c_{10} \gtrdot 0;\ d_{10} \gtrdot 0;\ e_{10} \gtrdot 0;\ f_{10} \gtrdot 0;\ g_{10} \gtrdot 0$
$a_{11} \gtrdot 0;\ b_{11} \gtrdot 0;\ c_{11} \gtrdot 0;\ d_{11} \gtrdot 0;\ e_{11} \gtrdot 0;\ f_{11} \gtrdot 0;\ g_{11} \gtrdot 0$
$a_{12} \gtrdot 0;\ b_{12} \gtrdot 0;\ c_{12} \gtrdot 0;\ d_{12} \gtrdot 0;\ e_{12} \gtrdot 0;\ f_{12} \gtrdot 0;\ g_{12} \gtrdot 0$
$0 \gtrdot$

Future work: search for a smaller non-Arguesian modular lattice

There have been many studies on the properties that a modular lattice must have in order to be non-Arguesian [46] [110] [18] [19] [20] [21] [17] [38] [123]. However, to this author's knowledge, no example of a specific finite lattice with that property has been published other than the 184-node lattice of Table 7.2, derived from the Veblen-MacLagan-Wedderburn 91-point geometry.

Unfortunately, a 184-node lattice is impractical as a counterexample for use with an automated equation checking tool. Thus it is desirable to find a smaller one. There are several possibilities for work in that direction. One is to search for specific lattices that result from the work mentioned above. This is not necessarily an easy task, since the conditions are often of a theoretical nature that do not lend themselves immediately to a computer algorithm, but it is probably a worthwhile effort for future work.

Here we will present another possible direction, based on the Moulton plane counterexample of Fig. 7.8 (p. 127). Of course, this plane is equivalent to a lattice with an infinite number of nodes, since each of the uncountable points on the Euclidean plane is the singleton of a lattice atom. In order to obtain a finite lattice, we can start with those points and lines used in the counterexample of Fig. 7.8, along with the instances of the join and meet operations that are used by the counterexample, that will assure us that the Arguesian law will fail. A lattice can be obtained by adding the lattice zero and unit then drawing a lattice Hasse diagram with only those subspaces as the nodes in between. This lattice will, however, also be non-modular.

The problem is whether we can add a finite number of additional nodes so that the modular law becomes satisfied. A related problem was considered in Ref. [101, p. 102103-20, Def. III.2], which defined so-called MMPL lattices in which finite extensions of an otherwise non-Hilbert lattice where added in order to satisfy more Hilbert lattice laws, so as to achieve an approximation sufficiently satisfactory for some some experimental purpose. Of course in the present case we want to find an exact result, not an approximation, since our problem is mathematical rather than experimental. Nonetheless, similar algorithms might be applicable to both approaches.

It may not be feasible to find such a finite lattice, if one exists, without the help of a computer-assisted search. Here we will describe the starting point for the problem that future work can be based on.

In Fig. 7.8, there are 11 points and 10 lines. A finite modular counterexample to the Arguesian law must have at least these points and lines. Adding the lattice zero (0) and unit (1), the starting lattice has $11+10+2=23$ nodes. The final finite lattice (if one exists) will have an unknown number of additional nodes.

Tables 7.3 and 7.4 show the join and meet function values necessary to ensure that the Arguesian law is violated and that the table (up to that point) represents a lattice. The unspecified entries, as well as possible additional rows and columns, would be filled in by a computer search that attempts to make the lattice modular.

We can also express the problem in terms of a starting minimal sublattice. The Hasse diagram for this starting sublattice is shown in Fig. 7.11. Of course, it is non-modular (as well as non-Arguesian). The problem is to extend this minimal sublattice with additional nodes and orderings until a modular lattice is built, if there is one. New orderings may be added to existing nodes (as well as new ones), i.e. more lines may be drawn on the Hasse diagram as long as a lattice still results, except that the ordering indicated by the dashed line may not be added in order to guarantee that the Arguesian law violation will be preserved.

If we add the dashed line to Fig. 7.11, we obtain the minimal sublattice for the projective geometry instance of Fig. 7.7 that demonstrates of the Arguesian law. It is interesting to note

Table 7.3: Join table for a starting lattice fragment in a search for a finite modular, non-Arguesian lattice. "…" means possible additional lattice nodes. The entries above the diagonal are omitted since they are just the reflection of the entries below. The bold entries indicate the lattice nodes involved in the Arguesian law violation, Eqs. (7.52) and (7.53). The remaining explicit entries are necessary for the table to represent a lattice. Entries with "·" would be determined by a future computer search to make the lattice modular.

$\vee$	0	a_0	a_1	a_2	b_0	b_1	b_2	c_0	c_1	c_2	d	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_c	1	$\cdots$
0	0																						
a_0	a_0	a_0																					
a_1	a_1	l_3	a_1																				
a_2	a_2	l_4	l_5	a_2																			
b_0	b_0	l_0	$\cdot$	$\cdot$	b_0																		
b_1	b_1	$\cdot$	l_1	$\cdot$	l_6	b_1																	
b_2	b_2	$\cdot$	$\cdot$	l_2	l_7	l_8	b_2																
c_0	c_0	$\cdot$	l_5	l_5	$\cdot$	l_8	l_8	c_0															
c_1	c_1	l_4	$\cdot$	l_4	l_7	$\cdot$	l_7	l_c	c_1														
c_2	c_2	l_3	l_3	$\cdot$	l_6	l_6	$\cdot$	$\cdot$	$\cdot$	c_2													
d	d	l_0	l_1	l_2	l_0	l_1	l_2	$\cdot$	$\cdot$	$\cdot$	d												
l_0	l_0	l_0	$\cdot$	$\cdot$	l_0	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_0	l_0											
l_1	l_1	$\cdot$	l_1	$\cdot$	$\cdot$	l_1	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_1	$\cdot$	l_1										
l_2	l_2	$\cdot$	$\cdot$	l_2	$\cdot$	$\cdot$	l_2	$\cdot$	$\cdot$	$\cdot$	l_2	$\cdot$	$\cdot$	l_2									
l_3	l_3	l_3	l_3	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_3	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_3								
l_4	l_4	l_4	$\cdot$	l_4	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_4	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_4							
l_5	l_5	$\cdot$	l_5	l_5	$\cdot$	$\cdot$	$\cdot$	l_5	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_5						
l_6	l_6	$\cdot$	$\cdot$	$\cdot$	l_6	l_6	$\cdot$	$\cdot$	$\cdot$	l_6	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_6					
l_7	l_7	$\cdot$	$\cdot$	$\cdot$	l_7	$\cdot$	l_7	$\cdot$	l_7	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_7				
l_8	l_8	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_8	l_8	l_8	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_8			
l_c	l_c	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_c	l_c	1	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	l_c		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$		$\cdot$

Table 7.4: Meet table for starting lattice fragment for a search for a finite modular, non-Arguesian lattice. See comments in caption for Table 7.3.

$\wedge$	0	a_0	a_1	a_2	b_0	b_1	b_2	c_0	c_1	c_2	d	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_c	1	$\cdots$
0	0																						
a_0	0	a_0																					
a_1	0	·	a_1																				
a_2	0	·	·	a_2																			
b_0	0	·	·	·	b_0																		
b_1	0	·	·	·	·	b_1																	
b_2	0	·	·	·	·	·	b_2																
c_0	0	·	·	·	·	·	·	c_0															
c_1	0	·	·	·	·	·	·	·	c_1														
c_2	0	·	·	·	·	·	·	·	·	c_2													
d	0	·	·	·	·	·	·	·	·	·	d												
l_0	0	a_0	·	·	b_0	·	·	·	·	·	d	l_0											
l_1	0	·	a_1	·	·	b_1	·	·	·	·	d	d	l_1										
l_2	0	·	·	a_2	·	·	b_2	·	·	·	d	d	d	l_2									
l_3	0	a_0	a_1	·	·	·	·	·	·	c_2	·	a_0	a_1	·	l_3								
l_4	0	a_0	·	a_2	·	·	·	·	c_1	·	·	a_0	a_1	a_2	·	l_4							
l_5	0	·	a_1	a_2	·	·	·	c_0	·	·	·	·	a_1	a_2	a_1	a_2	l_5						
l_6	0	·	·	·	b_0	b_1	·	·	·	c_2	·	b_0	b_1	·	c_2	·	·	l_6					
l_7	0	·	·	·	b_0	·	b_2	·	c_1	·	·	b_0	·	b_2	·	c_1	·	b_0	l_7				
l_8	0	·	·	·	·	b_1	b_2	c_0	·	·	·	·	b_1	b_2	·	·	c_0	b_1	b_2	l_8			
l_c	0	·	·	·	·	·	·	c_0	c_1	0	·	·	·	·	·	c_1	c_0	·	c_1	c_0	l_c		
1	0	a_0	a_1	a_2	b_0	b_1	b_2	c_0	c_1	c_2	d	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_c	1	·
$\vdots$	0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·		·

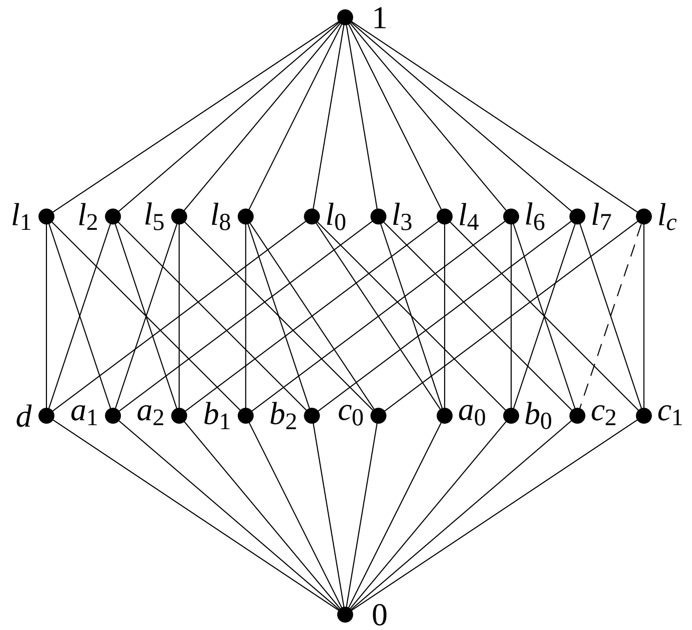


Figure 7.11: Hasse diagram of the minimal lattice corresponding to the projective geometry instance in Fig. 7.8 and Tables 7.3 and 7.4. The dashed line indicates an order that is *not* present. Any extension of this lattice (by adding additional nodes and orderings) will continue to fail the Arguesian law, except that the ordering corresponding to the dashed line must *not* be added. The goal of future work is to extend this lattice so that the modular law passes. (Note that the modular law fails in this minimal sublattice, of course, since it is not the solution to independence problem. In particular, it contains the pentagonal sublattice $\{0, d, l_1, 1, l_c\}$, making it non-modular by Th. 7.2.5.)

that the resulting Hasse diagram reveals a symmetry not apparent Fig. 7.7: if rotated 180°, the Hasse diagram is unchanged.

7.4 Higher-order Arguesian lattices

The Arguesian law also exists in higher-order forms, analogous to higher-order forms of the orthoarguesian law in the form of Eq. (4.18) (p. 42). These higher-order forms hold in all finite-dimensional Hilbert spaces, as we will prove below.

We are now ready to state our main theorem.

Theorem 7.4.1. (n-Arguesian Laws) *Let $a_0,\ldots,a_n$ and $b_0,\ldots,b_n$, $n\ge 1$, be (closed) subspaces of a finite-dimensional Hilbert space. We define the term $t_n^{\vee}(i_0,\ldots,i_n)$ by substituting $\vee$ for $+$ in the term $t_n(i_0,\ldots,i_n)$ from Theorem 4.1.1 (p. 38). Then following equation holds for $n\ge 1$:*

$$\begin{aligned}&(a_0\vee b_0)\cap\cdots\cap(a_n\vee b_n)\\&\le b_0\vee(a_0\cap(a_1\vee t_n^{\vee}(0,\ldots,n))).\end{aligned}\tag{7.54}$$

Proof. In any finite-dimensional subspace, $a+b=a\vee b$, where $\vee$ is the join of the lattice of subspaces of the vector space. Using this relationship along with the symbol substitutions mentioned into Eq. (4.12) (p. 40), the result follows. ☐

We call Eq. (7.54) the ***n*-Arguesian law**. The cases of $n=1$ and $n=2$ correspond to the modular and Arguesian laws, respectively. It is not known if these laws continue to be successively stronger for $n>2$. An open problem is whether these are equivalent to the higher-order forms of the Arguesian law mentioned in Ref. [34].

7.5 Pappian lattices

In projective geometry, **Pappus's postulate** states that if one is given one set of collinear points a, b, c, and another set of collinear points d, e, f, then the intersection points p, q, r of line pairs $\{a,e\}$ and $\{b,d\}$, $\{a,f\}$ and $\{c,d\}$, $\{b,f\}$ and $\{c,e\}$ are collinear [62]. This postulate, attributed to Pappus of Alexandria (c. 290-350), is illustrated in Fig. 7.12.

Pappus's postulate, like Desargue's, does not hold in all projective geometries. An outstanding feature of a **Pappian geometry** is contained in the following theorem [2, p. 71]:

Theorem 7.5.1. *Pappus's postulate holds in a projective geometry of projective dimension 2 or more iff the division ring constructed from the geometry is commutative i.e. a field.*

Obviously Pappus's postulate is independent of Desargue's, since the above theorem does not hold in all Desarguesian geometries. In fact, it implies that the geometry is Desarguesian [114] [118].

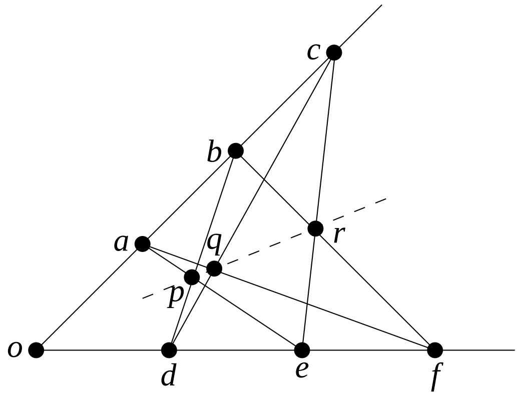


Figure 7.12: Illustration of Pappus's postulate in a projective plane.

Additional properties of Pappian geometries are discussed in Refs. [48] and [53].

From the above theorem, it follows that if the set of projective subspaces (see e.g. [40, Sec. 3]) constructed from a Hilbert lattice [Def. 2.3.1 (p. 19)] (to which an additional but currently unknown condition has been added) satisfies Pappus's postulate, then the multiplication operation in the division ring of final reconstructed Hilbert space will be commutative. When the harmonic conjugate condition [Def. 2.3.4 (p. 20)] is added to the Hilbert lattice to satisfy the conditions of Solèr's theorem, the only possible field of the resulting Hilbert space will be one of $\mathbb{R}$ or $\mathbb{C}$, since quaternionic multiplication is not commutative. This would bring us one step closer to the standard field $\mathbb{C}$ of quantum mechanics.

Therefore it is useful to search for a corresponding lattice identity. A partial result has been found by Day [16, Def. 4.7], who proposed the following condition:

Definition 7.5.2. *A modular lattice is called* **Pappian** *iff the following condition holds.*

$$\begin{aligned} & a\wedge(d\vee e)=b\wedge(d\vee e)\ \ \&\ \ a\wedge(d\vee e)=d\wedge(a\vee b) \\ &\qquad \&\ \ a\wedge(d\vee e)=e\wedge(a\vee b)\ \ \&\ \ c\le a\vee b\ \ \&\ \ f\le d\vee e \\ &\Rightarrow\quad (a\vee e)\wedge(b\vee d)\wedge(c\vee d\vee e)\wedge(f\vee a\vee b) \\ &\qquad \le((c\vee d)\wedge(a\vee f))\vee((c\vee e)\wedge(b\vee f)) \end{aligned} \tag{7.55}$$

This condition by itself does not imply the modular law, since it holds in the non-modular lattice of Ref. [6, Fig. 12, p. 42]. Moreover, when applied to the lattice of projective subspaces of a projective geometry, it holds only for the subspace lattice of vector spaces with dimension 2, or dimension 3 if its division ring is commutative [16, Cor. 5.3]. Thus it is of limited usefulness for Hilbert spaces generally, even those with commutative division rings.

It may be possible to weaken the condition in a way analogous to the weakening of the Arguesian law to obtain the orthoarguesian law, that might result in a more generally applicable "orthopappian law" that would hold in infinite-dimensional Hilbert space (and thus all dimensions). In order to serve as a useful condition to narrow down the Hilbert space division ring (for dim > 3), the main property needed is that it not hold in Hilbert spaces with non-commutative division rings (i.e. quaternions). This may be possible with a significantly weaker version of the law and is an open problem for future work.

Chapter 8

CONCLUSION

In Ch. 1 (p. 1) and specifically Sec. 1.3 (p. 7), we reviewed what is currently known about equations that hold in every $\mathcal{C}(H)$ (the lattice of closed subspaces of a finite- or infinite-dimensional Hilbert space). Aside from the OML law itself, these equations arise from three aspects of Hilbert space: geometry (nOA laws and Mayet's $\mathcal{E}_A$ equations), states (nGO laws and MGEs), and Hilbert-space valued states (Mayet's E equations), as summarized in Table 1.1 (p. 8). The discovery of these equations has been serendipitous, and it is open problem whether other aspects of Hilbert space will yield new equations.

The equational theory of OMLs, even though it has been known since 1937, is not known to be decidable. It remains a rich source of new results in itself, as our work in Ch. 3 (p. 27) showed.

Our investigation of the nOA laws [Ch. 4 (p. 37)] resulted in many new consequences and equivalences for those laws. An important open problem is the OA identity conjecture [Sec. 4.5 (p. 62)]. If this conjecture holds, it would prove to be a valuable tool for OA derivations, as Th. 4.5.3 (p. 64) shows. In particular, it would immediately prove the missing arrow directions in that theorem. In Sec. 4.5.2 (p. 74), we studied several possible approaches towards resolving this conjecture and showed specific equations that, if they hold in all OMLs, would prove the conjecture. Unexpectedly, the OA identity conjecture was found to be an instance of an inference due to von Neumann, Th. 7.2.6 (p. 118). Although von Neumann's inference does not itself hold in infinite dimensions, a study of its proof and consequences might eventually shed some light on the OA identity conjecture.

In Ch. 5 (p. 78), we reviewed in more depth the known equations based on states and vector-valued states. Assisted by several computer programs, a large number of finite OMLs (Greechie diagrams) was searched, resulting in 17 previously unknown MGEs (Mayet-Godowski equations) that are independent from all other known $\mathscr{C}(H)$ equations [Sec. 5.2.1 (p. 89)].

In Ch. 6 (p. 100), we explored two aspects of Hilbert space, modular symmetry and superposition, that might lead to new equations. While the existence of any such equations is still unknown, in Sec. 6.1.1 (p. 104) we outlined a possible technique and showed a specific conjecture which, if it holds in every OML, would lead to a new equation.

Finite-dimensional Hilbert spaces are important in quantum computation. In Ch. 7 (p. 111), we reviewed what is known about equations holding in finite dimensional $\mathscr{C}(H)$. In Th. 7.2.7 (p. 121), we proved that von Neumann's inference (mentioned above) is strictly weaker than the modular law. In Sec. 7.3 (p. 123), we reviewed the Arguesian law. An open problem is to find a smaller finite lattice counterexample showing that the Arguesian law is strictly stronger than the modular law. An apparently new result is that higher-order n-Arguesian equations [Sec. 7.4 (p. 140)] hold in finite dimensions, using a proof analogous to the one for the nOA laws. Finally, in Sec. 7.5 (p. 140), we speculated on the possibility of the existence of an equation based on Pappus' law and presented the known literature attempts in that direction.

Appendix A

COMPUTER PROGRAMS

This appendix summarizes the main computer programs that assisted with this work. The programs can be downloaded from the following web site: `http://us.metamath.org/#ql`.

With the exception of `shortdL` [mentioned in Sec. A.10 (p. 159)], `nauty` [mentioned in Sec. A.1 (p. 146)], and `metamath` [Sec. A.12 (p. 160)], each program's source code is self-contained in a single, stand-alone file with a `.c` extension. Only this source code file is publicly distributed and must be compiled to run on the specific platform of interest. Each program can be compiled with the `gcc` C-language compiler or equivalent, which is available for Linux, Unix, Windows, and Macintosh computers. For example, the program `latticeg.c` can be compiled using the command

```
gcc latticeg.c -o latticeg
```

from the computer's command-line shell (also called the command prompt or terminal window). Advanced users can apply various compiler optimization options to increase performance; these are described by the help documentation for the particular compiler version and platform.

Each program includes built-in documentation for its operation and options, which can be displayed with the `-help` option. For example, assuming `latticeg.c` was compiled as above into the user's current directory, the documentation can be invoked (on a Unix-type system) with

```
./latticeg --help
```

On some systems, the "`./`" prefix may not be needed.

In the main text and in the section titles below, we have appended "`.c`" to a program's name to indicate its source code file name, but in this appendix we will usually drop that suffix, which is not used to invoke the compiled version.

For brevity, we will not always show the detailed operation of all of these programs, but it should be straightforward to infer from their respective `-help` outputs along with studying the examples that follow. The style is very similar to that of the `latticeg` program that we describe in some detail below. In particular, the wff syntax for a program requiring an equation as an argument is identical to that for `latticeg`.

A.1 Program `latticeg.c`

The program `latticeg` was our primary tool for testing to see whether or not an equation holds (passes) or doesn't hold (fails) in each lattice in a list of lattices stored in an input file, with one line per lattice in MMP format, which is described by Def. 2.5.6 (p. 23) above. This program is described in Ref. [73, p. 2395]. Because it is so frequently used, we will go through it in some detail along with some examples. The usage of the other programs follows a similar style.

Before we discuss `latticeg`, it is useful to mention that an exhaustive, isomorphism-free list of all possible Greechie diagrams of a given size can be obtained with Brendan McKay's program `nauty` [71] [72] [73] [101]. This was often our starting point for finding lattices with desired characteristics. Typically we would pass the `nauty` output through a series of Unix pipe filters as described below.

The following listing is excerpted from the `-help` output of `latticeg`. For simplicity, we have shown only the options most frequently used. The `-help` output will show the complete set of options for the interested reader.

```
latticeg.c - Orthomodular Lattice Evaluator for Greechie Diagrams
Usage: latticeg [options] <hyp> <hyp> ... <conclusion>
     options:
       -a - test all lattices (don't stop after first failure)
       -v - show all visits to lattice points in a failure
       -n <integer> - test only the <integer>th lattice
       -f - show all failures in failing lattice
       -1 - print one formatted line per diagram, mainly for piping
       -i <file> - use Greechie lattices from <file> instead of the
                   built-in ones
       --i - same as -i but using standard input instead of a file
       -o (--o) <file> - write (append) output to <file> as well
            as screen
   latticeg -p <integer> - print the program's <integer>th lattice
       (you may use the -i and -o [or --o] options with -p)
   latticeg --help (or no argument) - print this message
```

For expressing equations, the variable names may be any lowercase letters other than `i`, `o`, and `v`. The built-in unary and binary operations are each expressed with one of the remaining characters. Wffs are expressed with ordinary notation, in which unary operations use prefix notation and binary operations use infix notation surrounded by parentheses. For example, the wff $a \wedge a = a \vee a''$ is expressed `(a^a)=(av--a)`. The full details of the syntax are given in the `-help` output, which we excerpt below:

```
Each <hyp> and the <conclusion> must be a <wff> defined as follows:

   <var> := a | b | c | d | e | f | g | h | j | k | l | m | n |
                p | q | r | s | t | u | w | x | y | z
   <opr> := ^ | v | # | O | I | 2 | 3 | 4 | 5
   <const> := 0 | 1          <uopr> := -
   <term> := <var> | <const> | <uopr> <term> | ( <term> <opr> <term> )
   <brel> := = | < | > | [    <ucon> := ~
   <bcon> := & | V | } | :
   <wff> := ( <term> <brel> <term> ) | <ucon> <wff>
             | ( <wff> <bcon> <wff> )

where a,b,c,... are variables (no i,o,v); 0,1 are constants; and
   - = negation (orthocomplement)
   ^ = conjunction (cap, meet, infimum)
   v = disjunction (cup, join, supremum)
   # = biimplication: ((x^y)v(-x^-y))
   O = ->0 = classical arrow: (-xvy)
   I = ->1 = Sasaki arrow: (-xv(x^y))
   2 = ->2 = Dishkant arrow: (-yI-x)
   3 = ->3 = Kalmbach arrow: (((-x^y)v(-x^-y))v(x^(-xvy)))
   4 = ->4 = non-tollens arrow: (-y3-x)
   5 = ->5 = relevance arrow: (((x^y)v(-x^y))v(-x^-y))
and = is equality, < is less-than-or-equal, > is g.e., [ is commutes:
   x<y is (xvy)=y; x>y is y<x; x[y is x=((x^y)v(x^-y)).
Metalogical connectives:  ~,&,V,},: are NOT,AND,OR,IMPLIES,EQUIVALENT.
The outermost parentheses of a <wff> are optional.

Predicate logic:

The present implementation has the following limitations:
1. No hypotheses may be present if quantifiers are used.
   Use & (AND) and } (IMPLIES) in the conclusion instead.
2. The conclusion must be a <qwff> as defined below.
```

```
3. No two quantifiers may be followed by the same variable.

We extend the wff syntax as follows:
    <qwff> := <wff> | @ <var> <qwff> | ] <var> <qwff>
where quantifier @ means "for all" and ] means "exists".

Thus the conclusion must be in prenex normal form, i.e. with all
quantifiers at the beginning of the expression.

Example:  'latticeg "]x@y(z<(xvy))"' means "for all z (implicitly),
there is an x s.t. for all y, z is l.e. xvy."
```

The connectives `0`, `I`, `2`, `3`, `4`, and `5` correspond to the implications of Def. 2.2.4 (p. 17). The meaning of the other connectives should be apparent from the listing above. A specific example of a complex predicate logic equation in this syntax is provided by the superposition condition given in the footnote to Eq. 6.33, p. 110.

We will next show a simple example of `latticeg` usage, which the reader may wish to reproduce to verify the program is working as expected. Suppose the file `godowski.oml` contains the following two lines:

```
123,345,567,789,9AB,BC1,2DG,6EG,AFG.
123,345,567,789,9AB,BCD,DEF,FG1,GHL,4IL,8JL,CKL.
```

These are the MMP encodings for the 3- and 4-spoke "wagon wheel" lattices of Fig. 5.1 (p. 79). We will test them against the 3-Go equation in the form of an instance of Eq. (5.8), p. 80:

$$((a\to b)\wedge(b\to c))\wedge(c\to a)\le b\to a \tag{A.1}$$

In the syntax of `latticeg`, this equation is expressed as

```
(((aIb)^(bIc))^(cIa))<(bIa).
```

The `latticeg` program invocation and output are as follows, where `$` indicates the shell input prompt:

```
$ latticeg -a -i godowski.oml '(((aIb)^(bIc))^(cIa))<(bIa)'
The input file has 2 lattice(s).
(((aIb)^(bIc))^(cIa))<(bIa)
FAILED #1 (16/9/34) at (((AIE)^(EIJ))^(JIA))<(EIA)
Passed #2 (21/12/44)
```

The `-a` option is important, for otherwise the program will stop after the first failing lattice is found. As expected, the 3-Go equations fails in the 3-spoke wheel and passes in the 4-spoke wheel. The failure message includes the nodal assignment to the variables where the first failure occurred. To interpret the node names, we can use the command

```
latticeg -p 1 -i godowski.oml
```

which will show the correspondence between the node names in the failing assignment and the atom names in the Greechie diagram.

The `-v` option shows all intermediate results of the failing assignment and lists the nodes, atoms, and blocks not "visited" during the evaluation of the failing assignment.

```
latticeg -a -v -n 1 -i godowski.oml '(((aIb)^(bIc))^(cIa))<(bIa)'
```

In particular, the listing lets us know which blocks can be stripped off of the Greechie diagram without affecting the failure in order to find a smaller counterexample. [Note that stripping blocks will not necessarily produce a sublattice of the original lattice; see Th. 2.5.8 (p. 26). Thus such smaller counterexamples must be carefully retested for other properties, such as continuing to hold for other equations when that is important.] For example, the `-v` option was used to help find the lattice of Fig. 4.2(b) (p. 50), which is a subset a much larger lattice of originally found by Peres to be a Kochen-Specker set, a purpose apparently completely unrelated to problem of 7OA independence [84].

In conjunction with Unix scripts and pipes, the `-1` option of `latticeg` and other programs here provides a powerful tool for automating massive searches of lattices with specific characteristics. The `-1` option outputs each MMP-encoded Greechie diagram prefixed with a pass/fail indicator (and some other information such as the atom, block, and node counts). This option can be used to filter a list of Greechie diagrams for certain characteristics (such as passing or failing the 3-Go equation). For example, in the above case we would see

```
$ latticeg -i 1.tmp -1 -a '(((xIy)^(yIz))^(zIx))<(yIx)'
#1 (21/12/44) passed: 123,345,567,789,9AB,BCD,DEF,FG1,GHL,4IL,8JL,CKL.
#2 (16/9/34) failed: 123,345,567,789,9AB,BC1,2DG,6EG,AFG.
```

A script to filter out failing lattices would search (e.g. `grep`) for the string "`passed: `" then remove the characters ending with that string, for passing to the next filter stage. (Other programs may have a different pass/fail prefix format with the `-1` option; see the `-help` for the individual program.) Certain older programs, including `latticeg`, were not initially designed with such piping in mind and differ slightly from newer ones in that the `-i` and `-o` options specify the input and output files. In newer programs, the input is usually taken from the standard input and the output sent to the standard output, as is the convention for most Unix command-line utilities. (The `-help` for each program will indicate the convention used.) However, such a “piped” mode can be emulated for `latticeg` with the `-i` option, as follows.

```
cat godowski.oml | latticeg -a --i -1 '(((xIy)^(yIz))^(zIx))<(yIx)' \
  | grep 'passed: ' | sed -e '/^.*: //' | latticeg -a --i -1 ...
```

If the `-i` or `-i` option is omitted, the program will test the equation against some built-in internal lattices. This behavior, which is normally never used, has its roots in early versions of `latticeg` which required hard-coded lattices before MMP encoding was devised. Although it is of historical interest only, we mention it so that the reader will not be confused if the `-i` or `-i` option is accidentally omitted.

A.2 Program `lattice2g.c`

The program `lattice2g` is identical to `latticeg` except that it incorporates an improved algorithm offering up to ten times speedup. From the user’s perspective, there is no difference from `latticeg`, and the two programs can be interchanged in any script without modification of the script. The reason for having `lattice2g` as a separate program is that the improved algorithm is very complex and thus somewhat “risky” (although no known bugs exist). The simpler `latticeg` provides an independent way to confirm the correctness of the algorithm (and also provides a way to benchmark the speedup of `lattice2g`).

The algorithm used in `lattice2g` is described in Ref. [84, Sec. 5].

A.3 Program `beran.c`

The program `beran` is used to simplify a one- or two-variable expression to a canonical form that is equivalent in any OML. For example, to simplify the expression $x' \vee (x \wedge (x' \vee (x \wedge y)))$, we can use

```
$ beran '(-xv(x^(-xv(x^y))))'
(-xv(x^(-xv(x^y)))) 78 (-xv(x^y)) (xIy)
```

This means the expression is equivalent to $x' \vee (x \wedge y)$ (which is the canonical form using primitive connectives $\vee$, $\wedge$, and $'$) and $x \to_1 y$ (which is an abbreviated form using the defined connective $\to_1$). The number 78 means that it is the 78th out of 96 possibilities [6, pp. 83–85].

In any OML, the validity of an equation with one or two variables is decidable. We can use `beran` to check the validity of such an equation. For example, to check that $x \wedge (x' \vee (x \wedge y))$ equals $x \wedge y$ in an OML, we can use

```
$ beran '((x^(-xv(x^y)))=(x^y))'
((x^(-xv(x^y)))=(x^y)) 96 1 1
```

If the result evaluates to 1, as above, the equation holds in any OML; any other result means that it doesn't hold. Note that the entire equation must be surrounded by parentheses since `beran` internally treats `=` as an operation rather than a binary relation.

A.4 Program `lattice.c`

The program `lattice` contains a series of built-in, hard-coded lattices that are counterexamples of successively more general classes of lattices. The first lattice that fails provides a rough indication of the "strength" of an equation given to it. This program is very useful for providing a crude, first-pass indication that, for example, a conjectured orthoarguesian law equivalent passes in a Boolean lattice (eliminating many kinds of typographical errors) and fails non-orthoarguesian counterexamples. While it of course does not prove the equivalence, it provides a useful filter for promising candidates for which we can search for a proof. For example, all of the 3OA equivalents in Sec. 4.4 above (p. 50) were first checked with `lattice` before their detailed proofs were worked out.

The syntax for equations, as well as many of the options, are the same as for `latticeg`, except that it does not have the ability to read lattices from an external file but can only make use of the built-in ones. For example, if we run it with the OML law $x \wedge (x' \vee (x \wedge y)) = x \wedge y$ as its equation argument, we see

```
$ lattice -a '(x^(-xv(x^y)))=(x^y)'
(x^(-xv(x^y)))=(x^y)
Passed 2-valued Boolean
Passed MO2 (modular)
Passed Beran Fig. 15 (modular)
Passed MO3 (modular)
Passed Dishkant (modular)
```

```
Passed Beran Fig. 12 (OA, non-modular)
Passed L42 (OA, non-modular)
Passed Mayet Fig. 5 (OM, OA, non-Go/Mayet)
Passed L38 (OM, non-OA)
Passed L36 (OM, non-OA)
Passed Godowski/Greechie L^ (OM, non-OA)
Passed L38M (OM, non-OA)
FAILED O6 (WOM, non-OM) at (b^(-bv(b^a)))=(b^a)
FAILED Beran Fig. 9h (WOM, non-OM) at (b^(-bv(b^a)))=(b^a)
FAILED Beran Fig. 9f (WOM, non-OM) at (b^(-bv(b^a)))=(b^a)
FAILED Beran Fig. 7b (WOM, non-OM) at (d^(-dv(d^a)))=(d^a)
FAILED Beran Fig. 11 (WOM, non-OM) at (d^(-dv(d^c)))=(d^c)
FAILED Rose-Wilkinson1 (WOM, non-OM) at (a^(-av(a^b)))=(a^b)
FAILED Beran Fig. 9g (non-WOM) at (c^(-cv(c^a)))=(c^a)
FAILED Beran Fig. 7c (non-WOM) at (A^(-Av(A^d)))=(A^d)
FAILED McCune (non-WOM) at (a^(-av(a^b)))=(a^b)
FAILED McCune2 (non-WOM) at (b^(-bv(b^a)))=(b^a)
FAILED McCune3 (non-WOM) at (e^(-ev(e^a)))=(e^a)
FAILED Rose-Wilkinson2 (non-WOM) at (a^(-av(a^b)))=(a^b)
```

The first 6 lattices above are OMLs, which the equation passes, and the rest are non-OMLs. Like with `latticeg`, the `-a` option means test against all lattices, otherwise it would stop on the first failure, lattice O6 [Fig. 2.1a, p. 18].

The contents of the lattices, including the node names shown in the failing assignments, can be listed with the `-p` option just as in `latticeg`. `OA`, `OM`, and `WOM` mean orthoarguesian [Ch. 4 above (p. 37)], orthomodular, and weakly orthomodular [102]. The Beran figures are found in Ref. [6]. For the Rose-Wilkinson lattices, see Refs. [100], [106], and [113]. For the McCune lattices, see Ref. [69], [102], [104], and [80]. For the Mayet lattice, see Ref. [65, p. 191]. For L36, see Ref. [76, p. 2360, Fig. 6(b)]. For L38m, see Ref. [76, p. 2366, Fig. 7(a)]. For L42, see our Fig. 6.3 (p. 105) or Ref. [76, p. 2366, Fig. 7(b)]. For L^, see Ref. [27, p. 247, Fig. (II)] or Ref. [76, p. 2366, Fig. 8(a)]. For L38, see Ref. [76, p. 2366, Fig. 8(b)]. For the Dishkant lattice, see Ref. [25, p. 16, Fig. 1]. For MO2 and MO3, see Ref. [14, p. 329, Figs. 1 and 2].

A useful feature of `lattice` is that its equation parser incorporates operation precedence (for example, `^` binds more tightly than `v`) and the backquote (`'`) may be used as a postfix operation in place of the prefix operation `-` for orthocomplementation. Since the other equation-handling programs such as `latticeg` (currently) accept only the strict syntax described in Sec A.1, `lattice` can be used to convert typed-in equations for use with `latticeg`. For example,

```
$ lattice -n 1 'xvy=xvx'^(xvy)'
(xvy)=(xv(-x^(xvy)))
Passed 2-valued Boolean
```

Here, `lattice` has internally converted the flexible syntax `xvy=xvx'^(xvy)` into the strict syntax `(xvy)=(xv(-x^(xvy)))`, which it prints out before testing. That line can be copied and pasted for use with `latticeg`. The operation precedence is documented in the last page of the `lattice -help` output (under the heading `Equation preprocessing`), but it can also be determined empirically just by looking at the converted equation that `lattice` prints. (The `-n 1` option above is used to suppress all lattice tests except the first.)

A.5 Program `hasse.c`

The program `hasse` is identical in behavior to `lattice`, but it takes the lattices (actually posets in general) from an input file instead of using built-in lattices. The input file encodes posets using the covering table notation described by Fig. 7.1 (p. 130). The `-help` option provides instructions for using it. (As of this writing, `hasse` is still undergoing development and is not yet ready for general use.)

A.6 Program `latticego.c`

For the general-purpose checking of whether an equation holds in a finite lattice, we primarily used `latticeg` (Sec. A.1), which tests an equation provided by the user against a list of MMP-encoded Greechie diagrams. While it has proved essential to our work, a drawback is that the run time increases quickly with the number of variables in and size of the input equation, making it impractical for huge equations.

But there is another limitation in principle, not just in practice, for the use of the `latticeg` program. In our work with MGEs [Sec. 5.2 (p. 81)], we were particularly interested in those lattices having no strong set of states but on which all of the successively stronger n-Gos pass, for all n less than infinity. This would prove that any MGE failing in that lattice is independent from all n-Gos and thus represents a new result. The `latticeg` program can, of course, check only a finite number of such equations, and when n becomes large the program is too slow to be practical. And in any case, it cannot provide a proof, but only evidence, that a particular lattice does not violate n-Go for any n.

Both of these limitations are overcome by a remarkable algorithm based on dynamic programming, that was suggested by Brendan McKay. This algorithm was incorporated into the `latticego` program, that is run against a set of lattices. No equation is given to the program; instead, the program tells the user the first n for which n-Go fails or whether it passes for all n. The program runs very quickly, depending only on the size of the input lattice, with a run time proportional to the fourth power of the lattice size (number of nodes) m, rather than increasing exponentially with the equation size (number of variables) n as with the `latticeg` program that checks against arbitrary equations.

A detailed description of the `latticego` algorithm can be found in Ref. [82, Sec. 6]. For an example of its use, assume the file `godowba.oml` contains the two 3- and 4-Go counterexamples described in Sec. A.1, along with a third line with a simple 3-atom Boolean algebra:

```
123,345,567,789,9AB,BC1,2DG,6EG,AFG.
123,345,567,789,9AB,BCD,DEF,FG1,GHL,4IL,8JL,CKL.
123.
```

This file can be tested with `latticego` as follows:

```
$ latticego -i godowba.oml 100
The input file has 3 lattice(s).
#1 (16/9/34) FAILED 3-Go
#2 (21/12/44) Passed 3-Go, FAILED 4-Go
#3 (3/1/8) Passed n-Go for all n (converged at 5-Go)
```

The output of `latticego` correctly identifies the first two lattices as 3- and 4-Go counterexamples and the Boolean algebra as satisfying n-Go for all n. The parameter $n = 100$ is simply an upper limit (the highest n-Go) at which to terminate the program if "convergence" hasn't yet occurred. More than $n = 10$ has rarely (if ever) been observed, and $n = 100$ provides a very safe margin.

A.7 Program `loop.c`

The program `loop` identifies loops [Def. 2.5.3 (p. 23)] that may occur in a Greechie diagram. The input to the program is a file containing a single Greechie diagram in MMP encoding. The program will list the loops that it finds.

For example, suppose the input file `go3.oml` contains the line

```
123,345,567,789,9AB,BC1,2DG,6EG,AFG.
```

This is the Greechie diagram for OML G3 in Fig. 5.3 (p. 81). The `loop` program is run as follows.

```
$ loop -i go3.oml
The input file has 1 lattice(s).
123,345,567,789,9AB,BC1,2DG,6EG,AFG. original
Starting block = 1
  6 123,345,567,789,9AB,BC1.  2*D.G. 6*E.G. A*F.G.
  7 213,345,567,789,9BA,AFG,GD2.  B*C.1* 6*E.G*
  5 213,345,576,6EG,GD2.  7*8.9. 9.A.B. B.C.1* A.F.G*
  7 123,345,576,6EG,GFA,A9B,BC1.  7*8.9* 2*D.G*
  7 231,1CB,BA9,987,756,6EG,GD2.  3*4.5* A*F.G*
  5 231,1CB,B9A,AFG,GD2.  3*4.5. 5.6.7. 7.8.9* 6.E.G*
Starting block = 2
Starting block = 3
  5 657,789,9BA,AFG,GE6.  1.2.3. 3.4.5* B*C.1. 2.D.G*
Starting block = 4
Starting block = 5
Starting block = 6
Starting block = 7
Starting block = 8
Starting block = 9
```

The three ways that this Greechie diagram is drawn in Fig. 5.3 was determined using the first three loops shown above. Let us look at the first loop, which corresponds to the center diagram in Fig. 5.3.

```
  6 123,345,567,789,9AB,BC1.  2*D.G. 6*E.G. A*F.G.
```

The `6` indicates the loop size is 6. The string `123,345,567,789,9AB,BC1.` is an MMP encoding for that loop, a hexagon. The next three strings, `2*D.G.`, `6*E.G.`, and `A*F.G.`, are the remaining lines (blocks) that are not part of the loop and are normally drawn inside of it. A `*` means that the line is connected to the loop itself. The atom `G` is obviously common to all three internal lines. The result is that the wagon wheel is essentially the only way that the Greechie diagram can be drawn given this outer loop.

Note: `loop` may occasionally be called `loopbig` in some documentation. The program `loopbig` was an enhancement to an older version of `loop` to handle larger Greechie diagrams, but it has been renamed `loop` and supersedes the original one.

A.8 Program oagen.c

The oagen program generates the nOA law in the format needed by programs such as latticeg. These equations are very long for large n, and this program eliminates the possibility of a typographical error when typing the nOA law by hand.

By default, oagen generates the nOA according to the recursive formula of Eq. (4.24), p. 44. Also by default, the output is a single long line, but the -in option indents the outer levels of the equation for easier reading. For example, the 5OA law can be produced as follows. Recall from Sec. A.1 (p. 147) that v, ^, -, and I mean $\vee$, $\wedge$, $'$, and $\to_1$ respectively.

```
$ oagen -n 5 -in
((aIe)^
  (
    (
      (((aIe)^(bIe))v((-aIe)^(-bIe)))v(
      (((aIe)^(cIe))v((-aIe)^(-cIe)))^
      (((bIe)^(cIe))v((-bIe)^(-cIe)))))v(
    (
      (((aIe)^(dIe))v((-aIe)^(-dIe)))v(
      (((aIe)^(cIe))v((-aIe)^(-cIe)))^
      (((dIe)^(cIe))v((-dIe)^(-cIe)))))^
    (
      (((bIe)^(dIe))v((-bIe)^(-dIe)))v(
      (((bIe)^(cIe))v((-bIe)^(-cIe)))^
      (((dIe)^(cIe))v((-dIe)^(-cIe)))))))))<(bIe)
```

For faster computations in latticeg, etc., the -sh option generates a shorter equivalent to the nOA law given by Eq. (4.68), p. 52.

```
$ oagen -n 5 -in -sh
(b^
  (
    (
      ((a^b)v((aIe)^(bIe)))v(
      ((a^c)v((aIe)^(cIe)))^
      ((b^c)v((bIe)^(cIe)))))v(
    (
      ((a^d)v((aIe)^(dIe)))v(
      ((a^c)v((aIe)^(cIe)))^
      ((d^c)v((dIe)^(cIe)))))^
    (
```

```
((b^d)v((bIe)^(dIe)))v(
((b^c)v((bIe)^(cIe)))^
((d^c)v((dIe)^(cIe))))))))<(-aIe)
```

A discussion comparing the sizes of the long and short versions can be found in Ref. [84, Sec. 5].

A.9 Program `states.c`

The `states` program is primarily used to check whether or not an input OML (in the form of an MMP-encoded Greechie diagram) admits a strong set of states [Def. 2.4.3 (p. 22)]. A description of its algorithm, which makes use of the linear programming algorithm, is described in Ref. [82, Secs. 4].

For example, suppose the file `statetest.oml` has the following lines:

```
123.
123,345,567,789,9AB,BC1,2DG,6EG,AFG.
```

corresponding to the 2^3 Boolean algebra and the wagon wheel of Fig. 5.1(a) (p. 79). We can test to see if these admit a strong set of states as follows:

```
$ ./states -i statestest.oml -m s -a
The input file has 2 lattice(s).
#1 (3/1/8) has a strong set of states
#2 (16/9/34) There is no state m s.t. (m(G) = 1 => m(4') = 1) => G =< 4'
```

The option `-m s` means “strong set of states” mode, and `-a` means don’t stop on the first Greechie diagram with admitting no strong set of states. As expected, the Boolean algebra admits a strong set of states, but the wagon wheel does not.

When an OML that does not admit a strong set of states is found, the `-qs` option can be used to generate a condensed state equation (p. 83). This feature is described in Ref. [82, Secs. 5] and was used to obtain the condensed state equations in, for example, Table 5.2 (p. 91). As an example of how this works for the wagon wheel lattice,

```
$ ./states -i statestest.oml -m s -a -qs
The input file has 2 lattice(s).
#1 (3/1/8) has a strong set of states
#2 (16/9/34) 123,345,567,789,9AB,BC1,2DG,6EG,AFG.
Raw st eq: 13+57+9B=35+79+B1
State eqn: ab+cd+ef=bc+de+fa
#2 (16/9/34) There is no state m s.t. (m(G) = 1 => m(4') = 1) => G =< 4'
```

The condensed state equation $ab + cd + ef = bc + de + fa$ corresponds exactly to the 3-Go equation [105, p. 776, Eq. (51)].

The `states` program also implements the detection of other kinds of states: a full set of states [64, p. 370] [5], a non-dispersive (0/1) set of states, no states at all, one state, and integer-valued (i.e. group-valued for group $\mathbb{Z}$) states [36] [36] [88] [90]. These other modes are documented in the `-help` listing. For the special case of non-dispersive states (the lack of which can indicate a Kochen-Specker set), the specialized program `states01` runs several orders of magnitude faster.

A.10 Program `subgraph.c`

An MMP encoding of a Greechie diagram [see Def. 2.5.6 (p. 23)] is not unique. For example, `123,345.` and `zA9,27A.` represent the same Greechie diagram. The program `subgraph` checks to see whether one MMP-encoded Greechie diagram (or more generally any hypergraph) is a subgraph of a "reference" diagram. In particular, when both diagrams have the same size, it checks to see whether they are isomorphic i.e. correspond to the same Greechie diagram.

The `subgraph` program has a number of modes allowing different combinations of inputs, to allow for example checking many potential subgraphs against a reference or checking one potential subgraph against many references. The `-help` option provides the details for the different modes. Here, we will show how to test the above example. Suppose the file `test.oml` has the two lines

```
zA9,27A.
123,456.
```

To find out if these are subgraphs (in this case, isomorphic to) `123,345.`, we can run `subgraph` as follows.

```
$ subgraph -r 123,345. < test.oml
The reference diagram is:
  123,345.
#1 zA9,27A.
  Isomorphism:  ref block #s, ref blocks, map to input block atoms:
    1   2
    123,345.
    z9A,A27.
  Backtrack count = 0
#2 123,456.
```

```
    The above input diagram is not a subgraph of the reference.
    Backtrack count = 3
  Total diagrams = 2  Total backtrack count = 3  CPU time =  0.03 s
```

Note that the atoms of the MMP encoding `zA9,27A.` are re-ordered under the reference diagram so that the one-to-one mapping of the isomorphism can easily be seen.

When isomorphic hypergraphs must be filtered from a very large collection, `subgraph` may be too slow, and Brendan McKay's much faster program `shortdL` [83] [108] can be used instead. An additional benefit is that all input hypergraphs are converted to a unique, canonical MMP encoding that can later be used to compare MMP encodings directly—indeed that is how `shortdL` works: after converting all input MMP encodings to a canonical form, it sorts them and filters out duplicates.

A.11 Program `mmpstrip.c`

The `mmpstrip` program is conceptually simple, in that produces all possible subgraphs of a Greechie diagram (or any hypergraph generally). It has a rich set of options such as random sampling when the output set would otherwise be too large. As usual, `mmpstrip -help` describes its operation and options. The algorithm and features are described in Ref. [83].

While its primary purpose is to assist in the search for new Kochen-Specker vector sets, it can also be useful for other purposes. For example, we can strip out blocks one at a time from an OML to see whether some desired property, such as providing a counterexample to an equation, continues to hold, in order to potentially reduce the size of the counterexample. It was used to assist the discovery [84] of a simpler counterexample that passes 6OA but fails 7OA, shown in Fig. 4.2(b) (p. 50).

A useful feature of `mmpstrip` is the `-b0` mode, meaning strip no blocks, which simply reproduces the input file with the side effect of renumbering the atoms in the MMP encoding, without any gaps in the numbering. For example, suppose the file `test.oml` has the two lines

```
zA9,27A.
123,456.
```

The first MMP encoding is not acceptable to the current version of certain programs such as `loop`, which require that atoms be numbered without gaps. (This deficiency is due to an early definition of MMP encoding that required gap-free atom numbering, and eventually it will be removed in future versions.)

```
      $ mmpstrip -b0 < test.oml
      123,452.
      123,456.
2 line(s) were output.
```

Here, `zA9,27A.` was renumbered to become `123,452.`, which has no gaps in the atom numbering and thus is acceptable to `loop`. It is important to be aware, though, that this is not a unique canonical form for the MMP encoding. To accomplish that, we can use the `shortdL` program, described in Sec. A.10 (p. 159).

The options of `mmpstrip` follow a slightly different convention from other programs in that there is no space between an option and any argument. For example, to strip 1 block we would use the option `-b1` and not `-b 1`. The `-help` option will clarify any such confusion.

A.12 Program `metamath`

A long-term goal is to formalize the proofs involved in the reconstruction of Hilbert space and verify them rigorously with an automated proof verifier. This would provide us with certainty that the construction is correct. The reconstruction is very complex, and errors exist in some of the literature. In addition, several pieces of the reconstruction exist only as informal proof sketches; while it is hoped that they have no gaps that can't be filled in, this can be known with certainty only by actually filling in those gaps.

The major theorem provers that exist today are outlined in the book The Seventeen Provers of the World [74]. Several can in principle be be used to verify the reconstruction, among them Metamath (which was developed by this author), Coq, HOL, Isabelle, and Mizar. (This is not necessarily an exhaustive list of suitable provers but represents some that this author has some knowledge of. Some of the 17 provers such as Otter, while important and useful in their own right, are primarily intended to prove stand-alone theorems of first-order logic rather than work with large integrated bodies of mathematical knowledge.)

In any case, it is possible that the reconstruction will be verified with Metamath at some point in the future. Most of the prerequisites, including a definition and development of Hilbert space, already exist in Metamath's set theory database, called `set.mm`.

There are several programs, developed by this author and others, that can be used to develop and verify Metamath proofs. The most important ones are `metamath` and `mmj2`. The first is described in depth in Ref. [75]; the second is a graphical interface program for developing Metamath proofs and is available, along with documentation, at the Metamath web site, `http://metamath.org`.

The `metamath` program is not normally run as a single command from the operating system's shell but has an interactive shell of its own with which the user builds and verifies proofs. It is invoked from the command line with a single argument consisting of the file name of an ASCII database of theorems written in the Metamath language. For example, the set theory database `set.mm` is opened as follows:

```
$ metamath set.mm
Metamath - Ver. 0.07.59 11-Dec-2010  Type HELP for help, EXIT to exit.
MM> read "../mm/set.mm"
Reading source file "set.mm"...
185648 lines (9720724 characters) were read from "1.tmp".
The source has 49102 statements; 908 are $a and 12411 are $p.
No errors were found.  However, proofs were not checked.  Type
VERIFY PROOF * if you want to check them.
MM>
```

The user can type `help` for a description of the many commands that are available. For example, to verify the proofs of all 12411 theorems currently in `set.mm`,

```
MM> verify proof *
0 10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
..................................................
All proofs in the database were verified in 4.29 s.
MM>
```

An example of a theorem in the `set.mm` database is called `uncom` and states that the union of two classes commutes. To see this theorem and its proof, we can use the following two commands.

```
MM> show statement uncom /comment
"Commutative law for union of classes.  Theorem 21 of [Suppes] p. 27."
8099 uncom $p |- ( A u. B ) = ( B u. A ) $= ... $.
MM> show proof uncom /lemmon/renumber
1 orcom          $p |- ( ( x e. A \/ x e. B ) <-> ( x e. B \/ x e. A ) )
2 elun           $p |- ( x e. ( A u. B ) <-> ( x e. A \/ x e. B ) )
3 elun           $p |- ( x e. ( B u. A ) <-> ( x e. B \/ x e. A ) )
4 1,2,3 3bitr4i  $p |- ( x e. ( A u. B ) <-> x e. ( B u. A ) )
5 4 eqriv        $p |- ( A u. B ) = ( B u. A )
MM>
```

In the above listing, `e.` is set membership $\in$. The tag `$p` means the statement to the left is a theorem (as opposed to `$a`, which means axiom or definition). In step 4, steps 1, 2, and 3 are assigned to the hypotheses of statement `3bitr4i`, which chains three logical equivalences. The proof can be drilled down as far as desired with successive `show statement` and `show proof` commands applied to the statements in the proof listing.

The above examples give a quick flavor of the `metamath` program, but it is not our purpose here to document it in detail. The interested reader can refer to Ref. [75].

Bibliography

[1] James C. Abbott. Sets, Lattices, and Boolean Algebras. Allyn and Bacon, Boston, 1969.

[2] Reinhold Baer. Linear Algebra and Projective Geometry. Academic Press, New York, 1952. Reprinted 2005 (Dover, New York).

[3] Jonathan Barrett and Matthew Leifer. The de Finetti theorem for test spaces. *New J. Phys.*, **11**:033024, 2009.

[4] Enrico G. Beltrametti and Gianni Cassinelli. The Logic of Quantum Mechanics. Addison-Wesley, 1981.

[5] Mary Katherine Bennett. A finite orthomodular lattice which does not admit a full set of states. *SIAM Rev.*, **12**:267–271, 1970.

[6] Ladislav Beran. Orthomodular Lattices; Algebraic Approach. D. Reidel, Dordrecht, 1985.

[7] Sterling K. Berberian. Introduction to Hilbert Space. Chelsea, New York, 1976.

[8] Garrett Birkhoff. Lattice Theory, volume XXV of American Mathematical Society Colloqium Publications. American Mathematical Society, New York, 2nd (revised) edition, 1948.

[9] Garrett Birkhoff. Lattice Theory, volume XXV of American Mathematical Society Colloquium Publications. American Mathematical Society, Providence, Rhode Island, 3rd edition, 1967.

[10] Samuel L. Braunstein. Error correction for continuous quantum variables. *Phys. Rev. Lett.*, **80**:4084–4087, 1998. *ArXiv:quant-ph/9711049*.

[11] Samuel L. Braunstein and H. J. Kimble. Dense coding for continuous variables. *ArXiv:quant-ph/9910010*, 1999.

163

[12] Samuel L. Braunstein and Arun K. Pati, editors. Quantum Information with Continuous Variables. Springer, Berlin–New York, 2003.

[13] Gunter Bruns. Free ortholattices. *Canad. J. Math.*, **28**:977–985, 1976.

[14] Günter Bruns and Gudrun Kalmbach. Varieties of orthomodular lattices. ii. *Canad. J. Math.*, **24**:328–337, 1972.

[15] Peter Crawley and Robert P. Dilworth. Algebraic Theory of Lattices. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[16] Alan Day. In search of a Pappian lattice identity. *Canad. Math. Bull.*, **24**:187–198, 1981.

[17] Alan Day, Christian Herrmann, Bjarni J'onsson, J. B. Nation, and Doug Pickering. Small non-Arguesian lattices. *Algebra Univers.*, **31**:66–94, 1994.

[18] Alan Day and Bjarni J'onsson. The structure of non-Arguesian lattices. *Bull. Amer. Math. Soc.*, **13**:157–159, 1985.

[19] Alan Day and Bjarni J'onsson. A structural characterization of non-Arguesian lattices. *Order*, **2**:335–350, 1986.

[20] Alan Day and Bjarni J'onsson. Non-Arguesian configurations in a modular lattice. *Acta Sci. Math. (Szeged)*, **51**:309–318, 1987.

[21] Alan Day and Bjarni J'onsson. Small non-Arguesian lattices. *Algebra Univers.*, **26**:208–215, 1989.

[22] Alan Day and Doug Pickering. A note on the Arguesian lattice identity. *Studia Sci. Math. Hungar.*, **19**:303–305, 1982.

[23] M. Dichtl. Astroids and pastings. *Algebra Univers.*, **18**:380–385, 1984.

[24] Markus Dichtl. There are loops of order three in orthomodular lattices. *Arch. Math.*, **37**:285–286, 1981.

[25] Hermann Dishkant. The first order predicate calculus based on the logic of quantum mechanics. *Rep. Math. Logic*, **3**:9–18, 1974.

[26] Radoslaw Godowski. Varieties of orthomodular lattices with a strongly full set of states. *Demonstratio Math.*, **14**:725–733, 1981.

[27] Radoslaw Godowski and Richard Greechie. Some equations related to the states on orthomodular lattices. *Demonstratio Math.*, **17**:241–250, 1984.

[28] G. Grätzer, H. Lakser, and Bjarni Jónsson. The amalgamation property for equational classes of modular lattices. *Pacif. J. Math.*, **45**:507–524, 1973.

[29] George Grätzer. General Lattice Theory. Birkhauser Verlag, Basel, 2nd edition, 1998.

[30] Richard J. Greechie. Orthomodular lattices admitting no states. *J. Comb. Theory A*, **10**:119–132, 1971.

[31] Richard J. Greechie. A non-standard quantum logic with a strong set of states. In Enrico G. Beltrametti and B. C. van Fraassen, editors, Current issues in quantum logic, (Proceedings of the Workshop on Quantum Logic held in Erice, Sicily, December 2–9, 1979, at Ettore Majorana Centre for Scientific Culture), pages 375–380. Plenum Press, New York, 1981.

[32] Herbert Gross. Hilbert lattices: New results and unsolved problems. *Found. Phys.*, **20**:529–559, 1990.

[33] Theodore Hailperin. A set of axioms for logic. *J. Symb. Logic*, **9**:1–19, 1944.

[34] Mark Haiman. Linear lattice proof theory: An overview. In Stephen D. Comer, editor, Universal Algebra and Lattice Theory, Proceedings of a Conference Held at Charleston, July 11–14, 1984, volume **1149** of Lecture Notes in Mathematics, pages 129–141. Springer, New York, 1985.

[35] Paul R. Halmos. Introduction to Hilbert Space and the Spectral Theory of Spectral Multiplicity. Chelsea, New York, 1957.

[36] John Harding, Ekaterina Jager, and Derek Smith. Group-valued measures on the lattice of closed subspaces of a Hilbert space. *Int. J. Theor. Phys.*, **44**:539–548, 2005.

[37] L. Herman, E. L. Marsden, and R. Piziak. Implication connectives in orthomodular lattices. *Notre Dame J. Formal Logic*, **16**:308–325, 1975.

[38] C. Herrmann. Alan Day's work on modular and Arguesian lattices. *Algebra Univers.*, **34**:35–60, 1995.

[39] Samuel S. Holland, Jr. Partial solution to mackey's problem about modular pairs and completeness. *Canad. J. Math.*, **21**:1518–1525, 1969.

[40] Samuel S. Holland, Jr. Orthomodularity in infinite dimensions; a theorem of M. Solèr. *Bull. Am. Math. Soc.*, **32**:205–234, 1995. *ArXiv:math/9504224v1*.

[41] Samuel S. Holland, Jr. Orthomodularity in infinite dimensions; a theorem of M. Solèr. *Bull. Am. Math. Soc.*, **32**:205–234, 1995.

[42] R. I. G. Hughes. The Structure and Interpretation of Quantum Mechanics. Harvard University Press, Cambridge, Massachusetts, 1989.

[43] Chris J. Isham. Lectures on Quantum Theory. Imperial College Press, London, 1995.

[44] P.-A. Ivert and T. Sjödin. On the impossibility of a finite propositional lattice for quantum mechanics. *Helv. Phys. Acta*, **51**:635–636, 1978.

[45] Peter Jipsen and Henry Rose. Varieties of Lattices. Springer-Verlag, New York, 1992.

[46] Bjarni Jónsson. Modular lattices and Desargues' theorem. *Math. Scand.*, **2**:295–314, 1954.

[47] Bjarni Jónsson. The class of Arguesian lattices is self-dual. *Algebra Univers.*, **2**:396, 1972.

[48] I. N. Kagno. Desargues' and Pappus' graphs and their groups. *Amer. J. Math.*, **69**:859–862, 1947.

[49] Gudrun Kalmbach. Orthomodular Lattices. Academic Press, London, 1983.

[50] Gudrun Kalmbach. Measures and Hilbert Lattices. World Scientific, Singapore, 1986.

[51] Gudrun Kalmbach. Quantum Measures and Spaces. Kluwer, Dordrecht, 1998.

[52] H. A. Keller. Ein nicht-klassischer Hilbertscher Raum. *Math. Z.*, **172**:41–49, 1980.

[53] Erwin Kleinfeld and Tuong Ton-That. An analytic proof of the theorems of Pappus and Desargues. *Note Mat.*, **22**:99–106, 2003.

[54] Erwin Kreyszig. Introductory Functional Analysis with Applications. John Wiley & Sons, New York, 1978.

[55] Seth Lloyd and Samuel L. Braunstein. Quantum computation over continuous variables. *Phys. Rev. Lett.*, **82**:1784–1787, 1999. *ArXiv:quant-ph/9810082*.

[56] George Whitelaw Mackey. The Mathematical Foundations of Quantum Mechanics. W. A. Benjamin, New York, 1963.

[57] M. Donald MacLaren. Atomic orthocomplemented lattices. *Pacif. J. Math.*, **14**:597–612, 1964.

[58] Maciej J. Mączyński. Hilbert space formalism of quantum mechanics without the Hilbert space axiom. *Rep. Math. Phys.*, **3**:209–219, 1972.

[59] Fumitomo Maeda and Shûichirô Maeda. Theory of Symmetric Lattices. Springer-Verlag, New York, 1970.

[60] Shûichirô Maeda. On the symmetry of the modular relation in atomic lattices. *J. Sci. Hiroshima Univ. Ser. A-I*, **29**:165–170, 1965.

[61] Shûichirô Maeda. On conditions for the orthomodularity. *Proc. Japan Acad.*, **42**:247–251, 1966.

[62] Elena Anne Marchisotto. The theorem of Pappus: A bridge between algebra and geometry. *Amer. Math. Monthly*, **109**:497–516, 2002.

[63] Tamás Matolcsi. Tensor product of Hilbert lattices and free orthodistributive product of orthomodular lattices. *Acta Sci. Math. (Szeged)*, **37**:263–272, 1975.

[64] René Mayet. Varieties of orthomodular lattices related to states. *Algebra Univers.*, **20**:368–396, 1985.

[65] René Mayet. Equational bases for some varieties of orthomodular lattices related to states. *Algebra Univers.*, **23**:167–195, 1986.

[66] René Mayet. Some characterizations of underlying division ring of a Hilbert lattice of authomorphisms. *Int. J. Theor. Phys.*, **37**:109–114, 1998.

[67] René Mayet. Equations holding in Hilbert lattices. *Int. J. Theor. Phys.*, **45**:1257–1287, 2006.

[68] René Mayet. Ortholattice equations and Hilbert lattices. In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Structures*, pages 525–554. Elsevier, Amsterdam, 2007.

[69] William McCune. Automatic proofs and counterexamples for some ortholattice identities. *Inform. Process. Lett.*, **26**:285–291, 1998.

[70] William W. McCune. Single axioms for groups and Abelian groups with various operations. Technical Report Preprint MCS-P270-1091, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, nov 1991.

[71] Brendan D. McKay. `nauty` user's guide (version 1.5). *Dept. Comput. Sci. Australian Nat. Univ. Tech. Rpt.*, TR-CS-90-02, 1990. `http://cs.anu.edu.au/˜bdm/nauty/`.

[72] Brendan D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, **26**:306–324, 1998.

[73] Brendan D. McKay, Norman D. Megill, and Mladen Pavičić. Algorithms for Greechie diagrams. *Int. J. Theor. Phys.*, **39**:2381–2406, 2000. *ArXiv:quant-ph/0009039*.

[74] Norman Megill. Metamath. In Freek Wiedijk, editor, The Seventeen Provers of the World, Lecture Notes in Artificial Intelligence, pages 88–95. Springer, Berlin, 2006. (Appears as pages 98–105 in `http://www.cs.ru.nl/~freek/comparison/comparison.pdf`).

[75] Norman D. Megill. Metamath: A Computer Language for Pure Mathematics. Lulu Publishing, Morrisville, North Carolina, 2007. `http://us.metamath.org/downloads/metamath.pdf`.

[76] Norman D. Megill and Mladen Pavičić. Equations, states, and lattices of infinite-dimensional Hilbert space. *Int. J. Theor. Phys.*, **39**:2337–2379, 2000. *ArXiv:quant-ph/0009038*.

[77] Norman D. Megill and Mladen Pavičić. Orthomodular lattices and a quantum algebra. *Int. J. Theor. Phys.*, **40**:1387–1410, 2001. *ArXiv:quant-ph/0103135*.

[78] Norman D. Megill and Mladen Pavičić. Deduction, ordering, and operations in quantum logic. *Found. Phys.*, **32**:357–378, 2002. *ArXiv:quant-ph/0108074*.

[79] Norman D. Megill and Mladen Pavičić. Equivalencies, identities, symmetric differences, and congruencies in orthomodular lattices. *Int. J. Theor. Phys.*, **42**:2797–2805, 2003. *ArXiv:quant-ph/0310063*.

[80] Norman D. Megill and Mladen Pavičić. Quantum implication algebras. *Int. J. Theor. Phys.*, **42**:2825–2840, 2003. *ArXiv:quant-ph/0310062*.

[81] Norman D. Megill and Mladen Pavičić. Mayet-Godowski Hilbert lattice equations. *ArXiv:quant-ph/0609192*, 2006.

[82] Norman D. Megill and Mladen Pavičić. Hilbert lattice equations. *Ann. Henri Poincaré*, **10**:1335–1358, 2010. *ArXiv:0912.1461*.

[83] Norman D. Megill and Mladen Pavičić. New class of tests of quantum contextuality. [unpublished], 2010.

[84] Norman D. Megill and Mladen Pavičić. Kochen-Specker sets and generalized Orthoarguesian equations. *Ann. Henri Poincaré*, **11**, 2011. *ArXiv:abs/1005.0016*.

[85] Peter Mittelstaedt. On the interpretation of the lattice of subspaces of hilbert space as a propositional calculus. *Z. Naturforsch.*, **27a**:1358–1362, 1972.

[86] Peter Mittelstaedt. Quantum Logic. Synthese Library; Vol. 126. Reidel, London, 1978.

[87] Forest Ray Moulton. A simple non-desarguesian plane geometry. *T. Am. Math. Soc.*, **3**:192–195, 1902.

[88] Mirko Navara. An orthomodular lattice admitting no group-valued measure. *Proc. Am. Math. Soc.*, **122**:7–12, 1994.

[89] Mirko Navara. Constructions of quantum structures. In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Structures*, pages 335–366. Elsevier, Amsterdam, 2007.

[90] Mirko Navara. Small quantum structures with small state spaces. *Int. J. Theor. Phys.*, **47**:36–43, 2008.

[91] Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, 2000.

[92] Douglas P1ckering. A selfdual Arguesian inequality. *Algebra Univers.*, **22**:99, 1986.

[93] R. Padmanabhan. On M-symmetric lattices. *Canad. Math. Bull.*, **17**:85–86, 1974.

[94] R. Padmanabhan and S. Rudeanu. Axioms for Lattices and Boolean Algebras. World Scientific, Singapore, 2008.

[95] Mladen Pavičić. Unified quantum logic. *Found. Phys.*, **19**:999–1016, 1989.

[96] Mladen Pavičić. Bibliography on quantum logics and related structures. *Int. J. Theor. Phys.*, **31**:373–461, 1992.

[97] Mladen Pavičić. Nonordered quantum logic and its YES-NO representation. *Int. J. Theor. Phys.*, **32**:1481–1505, 1993.

[98] Mladen Pavičić. Identity rule for classical and quantum theories. *Int. J. Theor. Phys.*, **37**:2099–2103, 1998.

[99] Mladen Pavičić. Quantum Computation and Quantum Communication: Theory and Experiments. Springer, New York, 2005.

[100] Mladen Pavičić. Quantum logic modelled by non-orthomodular lattices. *Int. J. Theor. Phys.*, **48**:[submitted], 2009.

[101] Mladen Pavičić, Brendan D. McKay, Norman D. Megill, and Krešimir Fresl. Graph approach to quantum systems. *J. Math. Phys.*, **51**:102103–1–31, 2010. *ArXiv:1004.0776*.

[102] Mladen Pavičić and Norman D. Megill. Binary orthologic with modus ponens is either orthomodular or distributive. *Helv. Phys. Acta*, **71**:610–628, 1998.

[103] Mladen Pavičić and Norman D. Megill. Quantum and classical implication algebras with primitive implications. *Int. J. Theor. Phys.*, **37**:2091–2098, 1998.

[104] Mladen Pavičić and Norman D. Megill. Non-orthomodular models for both standard quantum logic and standard classical logic: Repercussions for quantum computers. *Helv. Phys. Acta*, **72**:189–210, 1999. *ArXiv:quant-ph/9906101*.

[105] Mladen Pavičić and Norman D. Megill. Quantum logic and quantum computation. In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Structures*, pages 751–787. Elsevier, Amsterdam, 2007.

[106] Mladen Pavičić and Norman D. Megill. Standard logics are valuation-nonmonotonic. *J. Logic Comput.*, **18**:959–982, 2008. *ArXiv:0812.2702*.

[107] Mladen Pavičić and Norman D. Megill. Is quantum logic a logic? In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Logic*, pages 23–47. Elsevier, Amsterdam, 2009.

[108] Mladen Pavičić, Norman D. Megill, P. K. Aravind, and Mordecai Waegell. New class of 4-dim Kochen-Specker sets. *J. Math. Phys.*, **52**(*No. 2*):022104–1–9, 2011. *ArXiv:1004.1433* (2010).

[109] Mladen Pavičić, Jean-Pierre Merlet, Brendan D. McKay, and Norman D. Megill. Kochen-Specker vectors. *J. Phys. A*, **38**:1577–1592 and 3709 (corrigendum), 2005. *ArXiv:quant-ph/0409014*.

[110] Douglas Arthur Pickering. Minimal Non-Arguesian Lattices. PhD thesis, University of Hawaii, Honolulu, 1984.

[111] S. Ponnusamy. Foundations of Functional Analysis. Pangbourne, England, Alpha Science International Ltd., 2002.

[112] Alexander Prestel. Solèr's theorem. In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Logic*, pages 373–387. Elsevier, Amsterdam, 2009.

[113] Michael Rose and Kristin Wilkinson. Application of model search to lattice theory. *Association for Automated Reasoning Newsletter*, **52**, August, 2001.

[114] A. Seidenberg. Pappus implies Desargues. *Amer. Math. Monthly*, **83**:190–192, 1976.

[115] Maria Pia Solèr. Characterization of Hilbert spaces by orthomodular spaces. *Comm. Alg.*, **23**:219–243, 1995.

[116] Manfred Stern. Semimodular Lattice: Theory and Applications. Cambridge University Press, Cambridge, 1999.

[117] Karl Svozil. Quantum scholasticism: On quantum contexts, counterfactuals, and the absurdities of quantum omniscience. *Information Sciences*, **179**:535–541, 2009. *ArXiv:0711.1473*.

[118] G. Takách. Pappian implies Arguesian for modular lattices. *Algebra Univers.*, **42**:151–152, 1999.

[119] Gaisi Takeuti and Wilson M. Zaring. Introduction to Axiomatic Set Theory. Springer-Verlag, New York, 2nd edition, 1982.

[120] V. S. Varadarajan. Geometry of Quantum Theory, Vols. 1 & 2. John Wiley & Sons, New York, 1968, 1970.

[121] O. Veblen and J. H. Maclagan-Wedderburn. Non-desarguesian and non-pascalian geometries. *T. Am. Math. Soc.*, **8**:379–388, 1907.

[122] John von Neumann. Continuous Geometry. Princeton Univ. Press, Princeton, 1960.

[123] Charles Weibel. Survey of non-desarguesian planes. *Notices Am. Math. Soc.*, **54**:1294–1303, 2007.

[124] Philip M. Whitman. Lattices, equivalence relations, and subgroups. *Bull. Amer. Math. Soc.*, **52**:507–522, 1946.

[125] J. Jay Zeman. Quantum logic with implications. *Notre Dame J. Formal Logic*, **20**:723–728, 1979.

Index

173

Curriculum vitae

Norman Dwight Megill received the BSEE degree from the Massachusetts Institute of Technology in 1972. He studied for 2 years as a graduate student at MIT and passed the PhD qualifying exam. He left to work in industry before completing the program. At MIT he was a member of Tau Beta Pi, Etta Kappa Nu, and Sigma Xi honor societies, and was the recipient of a National Science Foundation Fellowship for his graduate studies.

From 1974 through 1976, he worked as a Senior Engineer at Digital Equipment Corporation. In 1976 he co-founded and worked for Boston Information Group, an electrical engineering consulting firm that remained in successful operation for 30 years.

Originally as a side interest, Megill spent many years studying and researching foundations of mathematics and quantum physics, supplemented by MIT courses taken as a special graduate student. This led to a long collaboration with Prof. Mladen Pavičić that has resulted in the publication of about two dozen papers on quantum logic and quantum structures.

Publication list

[1] Pierre B. Montandan, Norman D. Megill, Amy R. Kahn, William T. Peake, and Nelson Y.S. Kiang. Recording auditory-nerve potentials as an office procedure. Annals of Otology, Rhinology and Laryngology, **84**:1–9, 1975.

[2] Norman D. Megill. TDRs profile impedances of backplanes and PC boards. *Electronics*, pages 113–117, Jul. 14 1981.

[3] Norman D. Megill. A finitely axiomatized formalization of predicate calculus with equality. *Notre Dame J. Formal Logic*, **36**:435–453, 1995. `http://projecteuclid.org/euclid.ndjfl/1040149359`.

[4] Norman D. Megill and Martin W. Bunder. Weaker D-complete logics. *Logic J. IGPL*, **4**:215–225, 1996.

[5] Mladen Pavičić and Norman D. Megill. Binary orthologic with modus ponens is either orthomodular or distributive. *Helv. Phys. Acta*, **71**:610–628, 1998.

[6] Mladen Pavičić and Norman D. Megill. Quantum and classical implication algebras with primitive implications. *Int. J. Theor. Phys.*, **37**:2091–2098, 1998.

[7] Mladen Pavičić and Norman D. Megill. Non-orthomodular models for both standard quantum logic and standard classical logic: Repercussions for quantum computers. *Helv. Phys. Acta*, **72**:189–210, 1999. *ArXiv:quant-ph/9906101*.

[8] Brendan D. McKay, Norman D. Megill, and Mladen Pavičić. Algorithms for Greechie diagrams. *Int. J. Theor. Phys.*, **39**:2381–2406, 2000. *ArXiv:quant-ph/0009039*.

[9] Norman D. Megill and Mladen Pavičić. Equations, states, and lattices of infinite-dimensional Hilbert space. *Int. J. Theor. Phys.*, **39**:2337–2379, 2000. *ArXiv:quant-ph/0009038*.

[10] Norman D. Megill and Mladen Pavičić. Orthomodular lattices and a quantum algebra. *Int. J. Theor. Phys.*, **40**:1387–1410, 2001. *ArXiv:quant-ph/0103135*.

[11] Norman D. Megill and Mladen Pavičić. Deduction, ordering, and operations in quantum logic. *Found. Phys.*, **32**:357–378, 2002. *ArXiv:quant-ph/0108074*.

[12] Norman D. Megill and Mladen Pavičić. Equivalencies, identities, symmetric differences, and congruencies in orthomodular lattices. *Int. J. Theor. Phys.*, **42**:2797–2805, 2003. *ArXiv:quant-ph/0310063*.

[13] Norman D. Megill and Mladen Pavičić. Quantum implication algebras. *Int. J. Theor. Phys.*, **42**:2825–2840, 2003. *ArXiv:quant-ph/0310062*.

[14] Mladen Pavičić, Jean-Pierre Merlet, and Norman D. Megill. Exhaustive enumeration of Kochen-Specker vector systems. *The F*rench National Institute for Research in Computer Science and Control Research Reports, **RR-5388**, 2004. `http://www.inria.fr/rrrt/rr-5388.html`.

[15] Mladen Pavičić, Jean-Pierre Merlet, Brendan D. McKay, and Norman D. Megill. Kochen-Specker vectors. *J. Phys. A*, **38**:1577–1592 and 3709 (corrigendum), 2005. *ArXiv:quant-ph/0409014*.

[16] Norman D. Megill and Mladen Pavičić. Mayet-Godowski Hilbert lattice equations. *ArXiv:quant-ph/0609192*, 2006.

[17] Norman Megill. Metamath. In Freek Wiedijk, editor, The Seventeen Provers of the World, Lecture Notes in Artificial Intelligence, pages 88–95. Springer, Berlin, 2006. (Appears as pages 98–105 in `http://www.cs.ru.nl/~freek/comparison/comparison.pdf`).

[18] Norman D. Megill. Metamath: A Computer Language for Pure Mathematics. Lulu Publishing, Morrisville, North Carolina, 2007.
`http://us.metamath.org/downloads/metamath.pdf`.

[19] Mladen Pavičić and Norman D. Megill. Quantum logic and quantum computation. In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Structures*, pages 751–787. Elsevier, Amsterdam, 2007.

[20] Mladen Pavičić and Norman D. Megill. Standard logics are valuation-nonmonotonic. *J. Logic Comput.*, **18**:959–982, 2008. *ArXiv:0812.2702*.

[21] Mladen Pavičić and Norman D. Megill. Is quantum logic a logic? In Kurt Engesser, Dov Gabbay, and Daniel Lehmann, editors, Handbook of Quantum Logic and Quantum Structures, volume *Quantum Logic*, pages 23–47. Elsevier, Amsterdam, 2009.

[22] Norman D. Megill and Mladen Pavičić. Hilbert lattice equations. *Ann. Henri Poincaré*, **10**:1335–1358, 2010. *ArXiv:0912.1461*.

[23] Mladen Pavičić, Brendan D. McKay, Norman D. Megill, and Krešimir Fresl. Graph approach to quantum systems. *J. Math. Phys.*, **51**:102103–1–31, 2010. *ArXiv:1004.0776*.

[24] Mladen Pavičić, Norman D. Megill, and Jean-Pierre Merlet. New Kochen-Specker sets in four dimensions. *Phys. Lett. A*, **374**:2122–2128, 2010. *ArXiv:0910.1311*.

[25] Mordecai Waegell, P. K. Aravind, Norman D. Megill, and Mladen Pavičić. Parity proofs of the Bell-Kochen-Specker theorem based on the 600-cell. *Found. Phys.*, **41**:883–904, 2010. *ArXiv:1010.4353*.

[26] Mladen Pavičić, Norman D. Megill, P. K. Aravind, and Mordecai Waegell. New class of 4-dim Kochen-Specker sets. *J. Math. Phys.*, **52**(*No. 2*):022104–1–9, 2011. *ArXiv:1004.1433* (2010).

[27] Norman D. Megill and Mladen Pavičić. Kochen-Specker sets and generalized Orthoarguesian equations. *Ann. Henri Poincaré*, **11**, 2011. *ArXiv:abs/1005.0016*.

[28] Norman D. Megill, Krešimir Fresl, Mordecai Waegell, P. K. Aravind, and Mladen Pavičić. Probabilistic generation of quantum contextual sets. *Phys. Lett. A*, **375**:3419–3424, 2011. *ArXiv:1105.1840*.

[29] Norman D. Megill and Mladen Pavičić. Estimating Bernoulli trial probability from a small sample. *ArXiv:1105.1486*, 2011.