Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Algebraic computations in quantum logics

Doctoral Thesis

Jeannine J.M. Gabriëls

Supervisor: Prof. Ing. Mirko Navara, DrSc.

Ph.D. Programme: Electrical Engineering and Information Technology
Branch of Study: Mathematical Engineering

Prague, October 2015

To my parents Georgette Kerschot
and Constant Gabriëls

A person, who has seriously set a goal for himself, will definitely reach it.

Benjamin Disraeli (1804 – 1881), British politician

Thanks

This thesis is the final work of my study at the Czech Technical University in Prague[1]. There has been a lot of struggle and doubt along the way, I was working full–time in a software company while doing the research for this thesis. I would like to use these first pages to express my gratitude towards those who have helped me to find my path during the last years.

First of all, I want to express my deepest thanks to Prof. Ing. Mirko Navara, DrSc., who introduced me in the interesting topic of quantum logic and who supported me continuously in this challenging field. Thank you for your help and support. As you know, until very recently, I doubted that I could do this, and I could certainly not have done it without your help. It was great to work with you and I could not have wished for a better supervisor than you!

My son Joris Verschaeve, Ph.D., who not only introduced me to my first little steps in quantum mechanics, but he was a very great help in C^{++} programming.

Dr. Ulrike Vorhauer, who never doubted at my capabilities and taught me to write scientific work in English and corrected my writing style. Also my sons Joris and Peter gave me very useful tips in writing, "*Moeke, there is no Leitmotiv in your thesis!*". This made me thinking on it over and over.

Stephen Gagola III for his solutions at the moment I almost resigned to find an answer to a problem that kept me busy.

Prof. Pavel Pták with his questions, that made me almost lunatic but they were a challenge and a reason not to resign. He made me aware that I had to struggle more.

Then, I also want to thank all the staff members of the Faculty of Electrical Engineering. I have always appreciated the friendly and helpful atmosphere.

And last but not least I want to thank all my friends and family especially my husband, Romain Verschaeve, who supported me and tolerated the spoiled leisure time and ruined holidays.

[1]Supported by the Czech Technical University in Prague under project SGS12/187/OHK3/3T/13.

Anotace

Matematický popis kvantových jevů vyžaduje obecnější strukturu jevů, než je Booleova algebra. Birkhoff a von Neumann navrhli pro tento účel pojem ortomodulárního svazu. Jeho typickou vlastností je existence tzv. nekomutujících jevů, které nejsou současně pozorovatelné (např. poloha a hybnost, podle Heisenbergova principu neurčitosti).

Dlouho otevřenou otázkou zůstává, zda lze rozhodnout o ekvivalenci dvou formulí v ortomodulárních svazech ("word problem"). V Booleových algebrách to lze snadno docílit převedením formulí na jednoznačnou normální formu. K tomu je potřebná komutativita, asociativita a distributivita booleovských operací (konjunkce a disjunkce).

V ortomodulárních svazech odpovídající svazové operace (průsek a spojení) nesplňují distributivitu. To znemožňuje převod na normální formy. Disertace je věnována hledání alternativních postupů. Např. průsek je jen jedna ze 6 operací v ortomodulárních svazech, které zobecňují booleovskou konjunkci. V práci je studována otázka, zda některé z celkem 96 binárních operací v ortomodulárních svazech umožňují zavedení normálních forem podobně jako v klasické logice.

První otázkou bylo, které z operací splňují asociativní identitu, případně za předpokladu, že některé argumenty komutují. Dále je studována monotonie, která souvisí s distributivitou vzhledem k průseku a spojení. Závěr je, že neexistuje dvojice operací umožňujících normální formy jako v booleovském případě. Závěrečná kapitola se zabývá identitami podobnými těm, které studoval Moufang v souvislosti s algebrami kvaternionů a oktonionů. Jsou obecnější než asociativita, a tím otvírají možnosti pro další postup.

Vedlejším výsledkem práce je příspěvek k vlastnostem některých operací (např. Sasakiho projekce). Tyto nové nástroje na zjednodušení algebraických výpočtů dávají šanci na vypracování algoritmických postupů obecnějších než současné specializované programy.

Abstract

Mathematical description of quantum phenomena requires an event structure more general than a Boolean algebra. For this purpose, Birkhoff and von Neumann proposed the notion of an orthomodular lattice. Its typical feature is the existence of so–called non–commuting events, which are not simultaneously observable (like position and momentum, according to Heisenberg's uncertainty principle).

There is an old open problem whether the word problem for orthomodular lattices is solvable. Is it possible to decide whether two formulas are equivalent? In Boolean algebras, an easy positive answer is given by a transformation of the formula to a unique normal form. This requires the commutativity, associativity, and distributivity of the Boolean operations (disjunction and conjunction).

In orthomodular lattices, the corresponding lattice operations (join and meet) violate distributivity. This disables the use of normal forms. We looked for alternative approaches. E.g., the join is only one of six orthomodular lattice operations generalizing the disjunction. In the thesis, we study the question whether some of the 96 binary operations in orthomodular lattices admit normal forms similar to the classical logic.

The first question was which operations satisfy the associative identity, eventually under the assumption that some variables commute. Then we studied monotonicity because it is related to distributivity over the meet and join. The conclusion is that there is no pair of operations in orthomodular lattices admitting "Boolean–like" normal forms. In the last chapter we study "Moufang–like" identities, which were inspired by the algebras of quaternions and octonions. These identities generalize associativity and may enable further progress.

As a by–product, we proved interesting, yet unknown, properties of some orthomodular lattice operations (e.g., the Sasaki projection). These new tools simplify algebraic computations and give a chance to develop algorithms more general than the current specialized software.

Table of contents

List of tables

List of figures

Chapter 1

Contribution and outline

In abstract algebra, the answer to the question whether two words represent the same element can be non–trivial. Finding an algorithm to decide this question is referred as the word problem. One way to find such an algorithm is to bring the expressions to their normal forms. A *normal form* also called *canonical form* of an object is a standard way of presenting that object.

M. Dehn [9] was one of the first who described the word problem for groups. He was aware that the word problem would be complex to solve. Ph. Whitman [57, 58] not only proved the existence of normal forms in free lattices, he also gave an algorithm to calculate them. R. Freese and J. B. Nation [14] were able to implement Whitman's algorithms and developed also own algorithms for deciding if a lattice term has a lower cover.

The main trouble to find normal forms in orthomodular lattices is the shortage of distributivity, first positive results were found independently by D. Foulis [12] and S. Holland [30], let L be an orthomodular lattice and $a, b, c \in L$, such that any one of them commutes with the other two then, in this particular case, the distributive laws hold, this is known as the Foulis–Holland theorem. The free lattice generated by such three variables is distributive, this is called the *focussing technique* described by R. Greechie [21].

In Section 3 we summarise the basic theory and fundamental background, which will be used in the following chapters. This section is far from complete, our goal is to give the reader, not familiar with lattice theory, a brief introduction. The basic work par excellence for the theory of orthomodular lattice is the book by G. Kalmbach [37].

In the first part of Section 4 we describe the free algebras with non–free generators which we use frequently in this thesis. A second part of Section 4 contains a brief overview of the results obtained about the word problem.

In Section 5 we present the free orthomodular lattice $\mathrm{F}(x, y)$ which is an important instrument throughout our discussion. The complete description of this lattice is presented in the book by L. Beran [3].

The paper [47] by M. Navara describes how we can introduce the orthomodular lattice $\mathrm{F}(x, y)$ for our purposes, moreover he describes the possibility to use the free orthomodular lattices on more than two non–free generators. M. Hyčko implemented the method of M. Navara in a computer program [33] used throughout our work.

Not only the failing distributivity law makes it difficult to find normal forms but also the lack of associativity is a deficit. In Section 6 we inquire the associativity of orthomodular binary operations. First results can be found in the papers by H. Kröger [38] and L. Beran [2]. More recent publications on the associativity of orthomodular implications are the papers by B. D'Hooghe, J. Pykacz [10] and N. Megill, M. Pavičić [43] on the associativity of some orthomodular lattice implications.

We first identify the associative orthomodular lattice operations, there are only few fulfilling the associative law, therefore we continue the study of associativity in orthomodular lattices making some requirements under which the operations could fulfil the associativity equation.

In Section 7 we studied the monotonicity properties of orthomodular lattice operations. It is known that the Sasaki projection is monotone and preserves arbitrary suprema, but little results about monotonicity of the other binary orthomodular lattice operations are known.

In Section 8 we continue with some forms of weak associativity in orthomodular lattices. Weak associativity and alternative algebras appear frequently in the literature, but little is found about the special cases we studied in this thesis.

Parts of Sections 6 and 7 are based on joint research with M. Navara, [16] and [17] respectively. Section 8 falls partially back on the joint study with M. Navara and S. Gagola III [18].

The last Section 9 concludes our results and suggests directions for further study.

Chapter 2

Aims of the doctoral thesis

Mathematical description of quantum phenomena requires an event structure more general than a Boolean algebra. For this purpose, Birkhoff and von Neumann proposed the notion of an orthomodular lattice. Its typical feature is the existence of so-called non-commuting events, which are not simultaneously observable (like position and momentum, according to Heisenberg's uncertainty principle).

There is an old open problem whether the word problem for orthomodular lattices is solvable: Is it possible to decide whether two formulas are equivalent? In Boolean algebras, an easy positive answer is given by a transformation of the formula to a unique normal form. This requires the commutativity, associativity, and distributivity of the Boolean operations (disjunction and conjunction).

The scope we set ourselves in this thesis is following

1. Find operations in orthomodular lattices which satisfy the commutative or associative law, eventually other similar laws.

2. From the above operations, find pairs which satisfy the distributive laws or other identities which could be useful in simplification of formulas, in the best case, in finding normal forms in orthomodular lattices.

3. Discuss the possibilities of automated simplification of formulas in orthomodular lattices (extension of the current methods).

In orthomodular lattices, the corresponding lattice operations (join and meet) violate distributivity. This disables the use of normal forms. We looked for alternative approaches. E.g., the join is only one of six orthomodular lattice operations generalizing the disjunction. In the thesis, we study the question whether some of the 96 binary operations in orthomodular lattices admit normal forms similar to the classical logic.

The first question was which operations satisfy the associative identity, eventually under the assumption that some variables commute. Then we studied monotonicity because it is related to distributivity over the meet and join. The conclusion is that there is no pair of operations in orthomodular lattices admitting "Boolean-like" normal forms. In a later chapter we study "Moufang-like" identities, which were inspired by the algebras of quaternions and octonions. These identities generalize associativity and may enable further progress.

As a by-product, we proved interesting, yet unknown, properties of some orthomodular lattice operations (e.g., the Sasaki projection). These new tools simplify algebraic computations and give a chance to develop algorithms more general than the current specialized software.

Chapter 3

Previous results: Preliminaries

In this section, some fundamental theory and definitions are presented, as far as they are not defined later. We restrict ourselves to the concepts used in this thesis. The subject matter is based on the books by G. Kalmbach [37] and L. Beran [3] on orthomodular lattices. For lattices in general we used the book by G. Grätzer [20]. For theory of nonassociative algebras, quaternions and octonions, the books by R. Schafer [53], J. Conway and D. Smith [7] are used.

3.1 Fundamentals

Partial order, posets and lattices

A *partially ordered set* or *poset* is a pair $(P, \leqslant)$ with a set P and a reflexive, antisymmetric and transitive relation $\leqslant$ on the elements of P. We call a poset P *bounded* if P has a smallest element, the *zero* denoted as $\mathbf{0}$, and a maximal element, the *one* denoted as $\mathbf{1}$. An *orthocomplementation* on a bounded poset P is an operation

$$
\begin{aligned}
{}' : \quad & P \to P \\
& x \mapsto x'
\end{aligned}
$$

with the following properties

$$
\begin{array}{ll}
x' \text{ is the lattice–theoretical complement of } x & \left\{ \begin{array}{lcl} x \wedge x' & = & \mathbf{0}\,, \\ x \vee x' & = & \mathbf{1}\,, \end{array} \right. \\
{}' \text{ is order–reversing} & x \leqslant y \Rightarrow y' \leqslant x'\,, \\
{}' \text{ is an involution} & x'' = x\,.
\end{array}
$$

The binary operations $\wedge$ (meet) and $\vee$ (join) are defined as the infimum (greatest lower bound) and supremum (least upper bound) respectively.

If for all elements x, y of a partially ordered set C either $x \leqslant y$ or $x \geqslant y$ holds, then we call C a *chain.*

A poset for which every two–element subset has a supremum and an infimum is a *lattice*. We call it a *complete lattice* if this is the case for any subset. Similarly to bounded posets, a *bounded lattice* has a smallest element, the *zero*, and a maximum element, the *one*. A bounded poset P with an orthocomplementation is called an *orthoposet*. An *ortholattice* is a bounded lattice on which an orthocomplementation is defined. In the following we will denote a poset by P and a lattice by L.

A lattice $(L, \leqslant)$ can equivalently be defined as $(L, \wedge, \vee)$, with the same set L and two idempotent, commutative and associative operations, the join $\vee$ and the meet $\wedge$, for which the absorption laws hold

$$x \wedge (x \vee y) \;=\; x \quad \text{and} \quad x \vee (x \wedge y) \;=\; x$$

for all x and y in L.

A *sublattice* is a non–empty subset of a lattice, which is a lattice on its own with the restriction of the same meet and the same join as those of the original lattice.

Remark 3.1.1
We often write $(L, \wedge, \vee, {}', \mathbf{0}, \mathbf{1})$, or simply L, for an ortholattice.

A *distributive lattice* is a lattice in which the two distributive laws hold for every triple x, y and z

$$x \wedge (y \vee z) \;=\; (x \wedge y) \vee (x \wedge z) \quad \text{and} \quad x \vee (y \wedge z) \;=\; (x \vee y) \wedge (x \vee z)\,.$$

A *Boolean algebra* is a bounded distributive lattice in which every element has a unique complement.

Because of the symmetry in the definitions of supremum and infimum, zero and one, the following holds: if in an equation that holds for all posets, we interchange the symbols $\wedge$ and $\vee$, interchange zero and one and reverse all inequalities, then we obtain another valid statement, called the *dual*, which holds for all posets with zero and one. This property is called the *Principle of duality*.

The modular law

In an arbitrary lattice L the following inequality always holds for every $x, y, z \in L$:

$$x \leqslant z \Rightarrow x \vee (y \wedge z) \leqslant (x \vee y) \wedge z\,.$$

In a *modular lattice* also the inverse inequality holds. Thus a modular lattice is a lattice in which the *modular law* holds, which is defined as

$$x \leqslant z \Rightarrow x \vee (y \wedge z) = (x \vee y) \wedge z\,,$$

for all $x, y, z \in L$, it can be interpreted as a weak distributive law.

The modular law can also be written as $x \leqslant z$ implies $x \vee (y \wedge z) \geqslant (x \vee y) \wedge z$, or as an equation (the *modular identity*):

$$(x \wedge y) \vee (x \wedge z) = x \wedge \big(y \vee (x \wedge z)\big).$$

It is clear that every distributive lattice is a modular lattice. A *modular ortholattice* is a lattice which is orthocomplemented and modular.

Homomorphisms

An *order homomorphism* is a map between two partially ordered sets that preserves the order of the elements. Let P_1 and P_2 be posets and $\varphi : P_1 \to P_2$ a mapping, so that for all x and y in P

$$x \leqslant y \Rightarrow \varphi(x) \leqslant \varphi(y)$$

holds, then φ is an order homomorphism. If φ is bijective and the reverse mapping is still a homomorphism, then it is called an *isomorphism.* A homomorphism $\varphi : P_1 \to P_2$ is an *embedding* if for all x and y in P_1 if the following

$$x \leqslant y \Leftrightarrow \varphi(x) \leqslant \varphi(y)$$

holds. These definitions applied to lattices are equivalent, in general the definition on lattices is formulated as follows: let L_1 and L_2 be lattices. A map $\varphi : L_1 \to L_2$ is called a *lattice homomorphism* if φ respects the meet and join. That is, for all $x, y \in L_1$ both

$$\begin{aligned} \varphi(x \wedge y) &= \varphi(x) \wedge \varphi(y) \\ \varphi(x \vee y) &= \varphi(x) \vee \varphi(y) \end{aligned}$$

hold. A homomorphism $\varphi : L_1 \to L_2$ is called an *isomorphism* if φ is injective and the map φ^{-1} is also a homomorphism. If two lattices L_1 and L_2 are isomorphic, we write $L_1 \cong L_2$. A lattice homomorphism $\varphi : L_1 \to L_2$ is an *embedding* if $\varphi(L_1)$ is a sublattice of L_2 and $\varphi : L_1 \to \varphi(L_1)$ is an isomorphism [51].

Varieties

Let $\mathcal{I}$ be a set of lattice identities (equations), and denote by $\mathrm{Mod}(\mathcal{I})$ the *class* of all lattices that satisfy every identity in $\mathcal{I}$. A class $\mathcal{K}$ of lattices is a *lattice variety* if $\mathcal{K} = \mathrm{Mod}(\mathcal{I})$ for some set of lattice identities $\mathcal{I}$. The class of all lattices $\mathcal{L}$ is a lattice variety since $\mathcal{L} = \mathrm{Mod}(\varnothing)$. The following lattice varieties are frequently encountered:

$$\begin{aligned} \mathcal{T} &= \mathrm{Mod}\{x = y\} && \text{all trivial lattices,} \\ \mathcal{D} &= \mathrm{Mod}\{(x \wedge y) \vee (x \wedge z) = x \wedge (y \vee z)\} && \text{all distributive lattices,} \\ \mathcal{M} &= \mathrm{Mod}\{(z \wedge y) \vee (x \wedge y) = \big((z \wedge y) \vee x\big) \wedge y\} && \text{all modular lattices.} \end{aligned}$$

Algebras

The term *algebraic structure* or *algebra* refers to a pair (A, R) of a set A and a set R, where the set A is often called the *carrier* or *underlying set* and R is a set of a finite number of operations on the elements of A. By an *arity* of a function (operation) f we mean the number of arguments, or operands on which f operates on. The most used are *unary* operations, they have one argument, and *binary* operations with two operands. *Nullary* operations have no operands and are referred as constants.

Similar to sublattices, we can define a *subalgebra* as a non–empty subset of the carrier set of an algebra, which is an algebra on its own and which is closed under the same operations as those of the original algebra.

3.2 Quantum logic

G. Birkhoff and J. von Neumann [4] introduced quantum logic because the rules of classical Boolean logic could not explain the principles of quantum mechanics, mainly because the distributive law often fails in quantum mechanics.

A *Hilbert space* $\mathcal{H}$ is a linear space with scalar product, in which each Cauchy sequence has its limit value in it.

G. Birkhoff and J. von Neumann developed the logic of quantum mechanics by studying the structure of the lattice of projection operators $\mathbb{P}(\mathcal{H})$ on a Hilbert space $\mathcal{H}$. Therefore they are considered as the pioneers of a new field in algebra; the theory of quantum logic. From a strict formal view, quantum logic can be characterised as the logical structure based on the algebra of orthomodular lattices [11].

The lattice $\mathbb{P}(\mathcal{H})$ is an orthomodular lattice (see Section 3.2.1), also called a *Hilbert lattice*. It is not distributive, unless $\mathcal{H}$ is one–dimensional. The lattice $\mathbb{P}(\mathcal{H})$ is modular in case $\mathcal{H}$ has a finite dimension. It is orthomodular in case of infinite dimension. This was the base for the introduction of *orthomodular posets* and *orthomodular lattices* in quantum mechanics systems.

Let V and W be vector spaces over the same field K. A mapping $f : V \to W$ is said to be *linear* if for any two vectors x and y in V and any two scalars α and β in K, both additivity and homogeneity are satisfied.

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)\,.$$

The mapping is called *multilinear* if it is a function of several variables that is linear separately in each variable.

A function $f : \mathbb{R}_+ \to \mathbb{R}_+$, where $\mathbb{R}_+ := [0, \infty)$, is called *submultiplicative* if it satisfies the inequality

$$f(xy) \leqslant f(x)f(y) \quad \forall\, x, y \in \mathbb{R}_+\,.$$

It is called *multiplicative* if equality holds.

3.2.1 Orthomodular lattices

An *orthomodular lattice* L is an ortholattice in which, for each $x, y \in L$, the *orthomodular law*

$$x \leqslant y \quad \Rightarrow \quad y = x \vee (x' \wedge y)$$

holds. The orthomodular law can be seen as a weak distributive law, weaker than the modular law.

A characterization of quantum logic is the lack of the distributivity law in general, but also the absence of the associativity law, for operations other than meet or join, causes difficulties to perform calculations.

Commuting elements

An important relation in a lattice[1] L is the *commutation relation*, also called *compatibility relation*. For two elements x, $y \in L$, we say "*x commutes with y*" or "*x is compatible with y*" and write $x \,\mathrm{C}\, y$, it is defined as:

$$x \,\mathrm{C}\, y \quad \text{if} \quad x = (x \wedge y) \vee (x \wedge y')\,.$$

If L is an orthomodular lattice, then this relation is reflexive ($x \,\mathrm{C}\, x$) and symmetric ($x \,\mathrm{C}\, y \Rightarrow y \,\mathrm{C}\, x$). The commuting relation is transitive if and only if L is a Boolean algebra, in this case any pair of elements commutes. A Boolean algebra B can thus also be defined as an orthomodular lattice in which

$$x = (x \wedge y) \vee (x \wedge y')$$

for each x and y in B.

Some useful lemmas dealing with commuting elements are the following:

Lemma 3.2.1 (Beran, Theorem II.2.3 [3])
Suppose L is an ortholattice and x and y in L. If either $x \leqslant y$ or $x \leqslant y'$ then x and y commute.

Lemma 3.2.2 (Beran, Theorem II.3.7 [3])
If L is an orthomodular lattice, x, y in L then the following are equivalent:

(*i*)	x and y commute
(*ii*)	$x \wedge (x' \vee y) = x \wedge y$
(*iii*)	$x \vee (x' \wedge y) = x \vee y$.

Proposition 3.2.3 (Beran, Theorems II.4.2 and II.4.4 [3])
In any orthomodular lattice, if x commutes with y and with z, then x commutes with y', with $y \vee z$ and with $y \wedge z$, as well as with any (ortho–)lattice polynomial in variables y and z.

[1] The commutation relation is also defined on orthoposets.

The Foulis–Holland Theorem

Some methods to overcome the absence of distributivity were developed by D. Foulis [12] and S. Holland [30] independently, the so–called *Foulis–Holland Theorem* provides a weak but very useful alternative to the distributive law.

Theorem 3.2.4 (Foulis, Holland [12, 30])
Let L be an orthomodular lattice and $x, y, z \in L$ such that one of them commutes with the other two, then both distributive laws hold

$$\begin{aligned} x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z)\,, \\ x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z)\,. \end{aligned}$$

The proof can be found in the book by L. Beran [3, Theorem II.3.10, p. 48].

We call a *Foulis–Holland set* a non–empty subset of an orthomodular lattice S, such that for every triple (x, y, z) of distinct elements of S, one of them commutes with the other two.

A second tool to overcome the lack of distributivity in orthomodular lattices is the *Focusing technique* by R. Greechie [21, 22].

Theorem 3.2.5 (Greechie [21, 22])
If S is a Foulis–Holland set, then the sublattice generated by S is distributive.

A third technique was developed by M. Navara [47]. It uses the free orthomodular lattice $\mathrm{F}(x, y)$ generated by two free generators x and y. This technique will be discussed in detail in Section 5 and 5.2 and will be used throughout this thesis.

3.2.2 Intervals in orthomodular lattices

Intervals in orthomodular lattices are defined in the usual way:

$$[x, y] = \{a \in L : x \leqslant a \leqslant y\} \quad \text{for } x, y \in L \text{ and } x \leqslant y\,.$$

The interval $[\mathbf{0}, z]$, $z \in L$, is an orthomodular lattice on its own, if the orthocomplement of $r \in [\mathbf{0}, z]$ is defined as $r^{\sharp} := r' \wedge z$ [37].

An element κ of an orthomodular lattice L is called a *central element* if κ commutes with every element of L, and the set

$$C(L) := \{\kappa \in L : \kappa \,\mathrm{C}\, x, \forall x \in L\}$$

is called the *center* of L. Note that the zero and the one is always in the center.

The center of L is a subalgebra of L and a Boolean algebra on its own [37]. One can prove that for a central element $c \in C(L)$ and $z \in L$, $c \wedge z \in C([\mathbf{0}, z])$ holds. But this does not hold in the opposite direction: not every element of $C([\mathbf{0}, z])$ can be written as $c \wedge z$ for some $c \in C(L)$ [27].

The theorem of M. D. MacLaren [40] is important, it treats the decomposition of orthomodular lattices into a direct product of intervals. This theorem allows us to use calculations in the free orthomodular lattice $\mathrm{F}(x, y)$ on two free generators x and y. We spend special attention on $\mathrm{F}(x, y)$ in Section 5.

Theorem 3.2.6 (MacLaren [40])
Let κ be a central element of an orthomodular lattice L. Then

$$L \cong [\mathbf{0}, \kappa] \times [\mathbf{0}, \kappa'] .$$

This isomorphism is given by $x \mapsto (x \wedge \kappa\,,\, x \wedge \kappa')$, the orthocomplementation on $[\mathbf{0}, \kappa]$ is given by $x^{\sharp} := x' \wedge \kappa$ and the orthocomplementation on $[\mathbf{0}, \kappa']$ by $x^{\flat} := x' \wedge \kappa'$.

Note: this theorem holds also if L is an ortholattice. The proof and further details can be found in the book by G. Kalmbach [37, p. 20].

3.3 Division algebras

This section might appear handling a different matter as before; we introduce the division algebras, quaternions and octonions as context information for the next section about nonassociative algebras.

Roughly spoken a *division algebra* $(D, +, \cdot)$ is a unit ring and $D\backslash\{0\}$ is a set in which a multiplication and division is defined. For each pair of elements $a, b \in D$, $a \neq 0$, the equations

$$\begin{aligned} a \cdot x &= b \\ y \cdot a &= b \end{aligned}$$

have unique solutions $x, y \in D$.

A groupoid G (generalisation of the notion "group", with a (partial) function replacing the binary operation), is a *quasigroup* if for all $a, b \in G$, there exist unique elements x, y in G

$$\begin{aligned} a \cdot x &= b \\ y \cdot a &= b\,. \end{aligned}$$

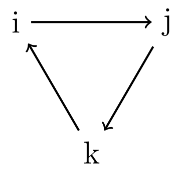
A quasigroup need not have an identity element, it does not need to be associative. Quasigroups are precisely groupoids whose multiplication tables are Latin squares. A quasigroup can be empty.

The algebra of *quaternions* is a skew four–dimensional algebra, in which the elements have the form

$$z := a + b \cdot \mathrm{i} + c \cdot \mathrm{j} + d \cdot \mathrm{k}\,.$$

The base of the quaternions is $(1, \mathrm{i}, \mathrm{j}, \mathrm{k})$, where $\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{i} \cdot \mathrm{j} \cdot \mathrm{k} = -1$ and the multiplication between these elements is defined as

$$\begin{array}{lclcl} \mathrm{i} \cdot \mathrm{j} & = & \mathrm{k} & = & -\mathrm{j} \cdot \mathrm{i} \\ \mathrm{j} \cdot \mathrm{k} & = & \mathrm{i} & = & -\mathrm{k} \cdot \mathrm{j} \\ \mathrm{k} \cdot \mathrm{i} & = & \mathrm{j} & = & -\mathrm{i} \cdot \mathrm{k} \end{array}$$



The *octonions* form an eight–dimensional algebra, with base $(1, \mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}, \mathrm{m}, \mathrm{n}, \mathrm{o})$ and $\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{l}^2 = \mathrm{m}^2 = \mathrm{n}^2 = \mathrm{o}^2 = \mathrm{i} \cdot \mathrm{j} \cdot \mathrm{k} \cdot \mathrm{l} \cdot \mathrm{m} \cdot \mathrm{n} \cdot \mathrm{o} = -1$. By defining $\mathrm{i}_0 = \mathrm{i}$, $\mathrm{i}_1 = \mathrm{j}$, $\mathrm{i}_2 = \mathrm{k}, \ldots, \mathrm{i}_6 = \mathrm{o}$, the multiplication rules between these elements are defined as follows, let $n = 0, \ldots, 6$:

$$\begin{array}{lclcl} \mathrm{i}_{n+1} \cdot \mathrm{i}_{n+2} & = & \mathrm{i}_{n+4} & = & -\mathrm{i}_{n+2} \cdot \mathrm{i}_{n+1} \\ \mathrm{i}_{n+2} \cdot \mathrm{i}_{n+4} & = & \mathrm{i}_{n+1} & = & -\mathrm{i}_{n+4} \cdot \mathrm{i}_{n+2} \\ \mathrm{i}_{n+4} \cdot \mathrm{i}_{n+1} & = & \mathrm{i}_{n+2} & = & -\mathrm{i}_{n+1} \cdot \mathrm{i}_{n+4} \end{array}$$

See also the book by J. H. Conway and D. A. Smith [7].

3.4 Non-associative algebras

In general, a *nonassociative algebra* is an algebra over a field, in some cases a ring, on which a, not necessarily associative, operation is defined. This operation is often called the multiplication. By associativity in an algebra, we mean the independence of the value of an operation on the distribution of parentheses within an expression [56]. The associative law for three elements of an algebra is written as

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z\,.$$

The topic of nonassociative algebras received attention after the discovery of the octonions. Following the investigation of the quaternions by W. Hamilton, his friend, J. Graves, described a new eight–dimensional algebra, which he called the “octaves”. Independently A. Cayley discovered the same algebra now called the octonions or *Cayley numbers*. This algebra satisfies the usual axioms for addition and multiplication. The multiplication however is not commutative, and does not necessarily satisfy the associative law. Addition and multiplication of this algebra do satisfy the distributive laws [1] in the usual way.

The main motivation of the work by W. Hamilton was the formulation of a three–dimensional normed division algebra. His endeavour was, however, unfruitful, since three–dimensional normed division algebras do not exist [1]. After having given up the quest of three dimensions and having the inspiration to go over to a four–dimensional algebra, he succeeded in the formulation of the quaternions.

There are exactly four division algebras with a (submultiplicative) norm: the real numbers $\mathbb{R}$, they are the best known, the complex numbers $\mathbb{C}$, they are not ordered, but algebraically complete, noncommutative quaternions $\mathbb{H}$, and the octonions $\mathbb{O}$. Among the normed division algebras the octonions are the largest, they are known to be nonassociative. There exist still larger non–normed division algebras but in contrast to the four normed ones, they are characterised by having *null divisors.* Two elements $a, b \in D$ are called null divisors if $a \cdot b = 0$, but $a \neq 0$ and $b \neq 0$.

Interest in the octonions in physics was first specified in the paper by P. Jordan, J. von Neumann and E. Wigner [34]. However their attempt to apply octonionic quantum mechanics to nuclear and particle physics had little success. It was only in the 1980s, that it was realized that the octonions explain some particular features of string theory[2] [1].

3.5 Hasse and Greechie diagrams

Throughout the thesis, we use Hasse and Greechie diagrams in order to create a graphical idea of a lattice. We say that an element y *covers* the element x, if $x \leqslant y$ and for an element z with $x < z \leqslant y$ it follows $z = y$. An *atom* is defined as the element which covers the zero. A *coatom* is the element which is covered by the one.

A *Hasse diagram* represents a finite poset S as a graph. The vertices represent the elements of S. The edges represent the covering order. If x, $y \in S$, and $x \leqslant y$, then the edge between x and y is drawn in a way that x is lower than y. If x, y and z are elements of S, so that $x \leqslant y$ and $y \leqslant z$, then by the transitivity property $x \leqslant z$. In a Hasse diagram this is shown by a path from x up to z, by passing through the elements between e.g. y. The diagram for a lattice need not to be unique, Figure 3.1 shows a rather "unusual" Hasse diagram of the Boolean algebra $\mathbf{2^4}$, the more used Hasse diagram of $\mathbf{2^4}$ can be seen in Figure 5.3.

Sometimes Hasse diagrams can be rather large and the diagram gets unclear, then we prefer the *Greechie diagram.* Let L be a lattice and B_i, $i = 1, \ldots, n$, its blocks (maximal Boolean subalgebras of L). A Greechie diagram associated with $L = \bigcup_i B_i$ consists of the set of vertices representing the atoms of L and a set of edges. The edges correspond to the blocks of L [37]. In Figure 3.1, on the right side, the Greechie diagram of $\mathbf{2^4}$ is depicted.

[2] A theoretical framework in which the point–like particles of particle physics are replaced by one–dimensional objects called strings; different types of observed elementary particles arise from the different quantum states of these strings.

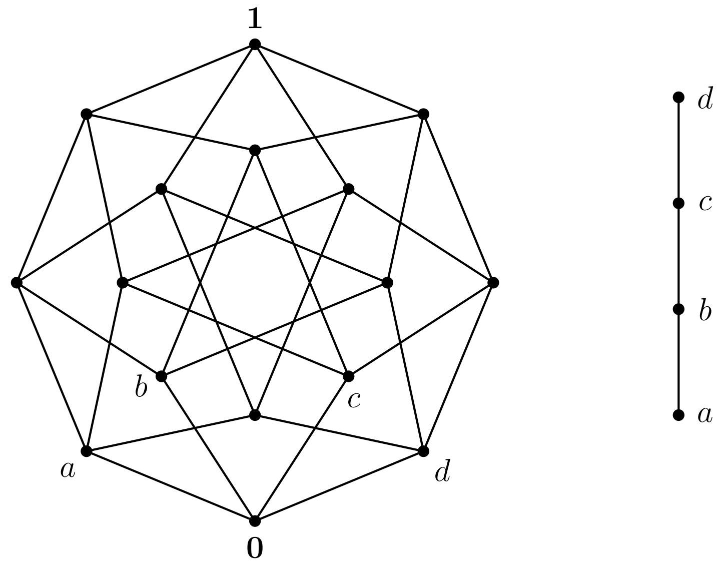


Figure 3.1: Hasse diagram and Greechie diagram of the Boolean algebra $\mathbf{2^4}$.

Chapter 4

Previous results: Free algebras

4.1 Free Algebras

Definition 4.1.1
Let $\mathcal{K}$ be a non–empty variety of algebras. We call F a *free algebra* in the variety $\mathcal{K}$, if F belongs to $\mathcal{K}$ and contains a set X of generators such that every mapping φ from the set of generators X into any algebra $A \in \mathcal{K}$ can be extended to a homomorphism

$$\psi : F \to A.$$

The map $\varepsilon : X \to F$ in Figure 4.1 is the so–called *inclusion map* that sends each element, x of X to x, treated as an element of F.

It is known that free algebras exist in a variety.

Theorem 4.1.2
The algebra from Definition 4.1.1 has the following property: any equation of the form

$$p(x_1, \ldots, x_n) = q(x_1, \ldots, x_n)\,,$$

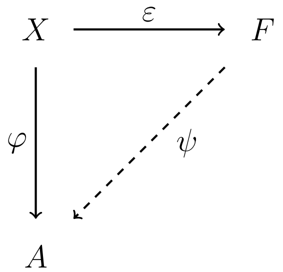


Figure 4.1: Diagram of the mappings from Definition 4.1.1

where $x_1, \dots, x_n \in X$, p and q are n–ary terms, holds in F, if and only if it holds in any algebra from the variety $\mathcal{K}$, this means, $p(y_1, \dots, y_n) = q(y_1, \dots, y_n)$ holds for any $A \in \mathcal{K}$ and any $y_1, \dots, y_n$ in A.

The reason for this is that the homomorphism φ preserves the validity of equations and any sequence $y_1, \dots, y_n$ can be chosen as $\varphi(x_1), \dots, \varphi(x_n)$.

In Definition 4.1.1, the elements of X are called *free generators* of the free algebra F. The choice of free generators is not important, as the following theorem explains:

Theorem 4.1.3
Algebras, from a given variety, freely generated by the same number of free generators, are isomorphic.

We sometimes need a free algebra with generators which are *not* free, i.e. they satisfy additional equations, this leads us to the following definition:

Definition 4.1.4
Let $\mathcal{K}$ be a non–empty variety of algebras, let $F \in \mathcal{K}$ and $\{x_1, \dots, x_n\} \subset F$. Let Q be a set of valid equations of the form

$$p_i(x_1, \dots, x_n) = q_i(x_1, \dots, x_n)\,,$$

$i \in I$, were p_i and q_i are n–ary terms. We say that F is freely generated by $x_1, \dots, x_n$ under constraints Q if every mapping $\varphi : \{x_1, \dots, x_n\} \to A$ such that

$$p_i\big(\varphi(x_1), \dots, \varphi(x_n)\big) = q_i\big(\varphi(x_1), \dots, \varphi(x_n)\big)$$

extends to a homomorphism $\psi : F \to A$.

4.2 The word problem

Let S be a set of letters, the finite sequences of n letters, elements of S^n, $n \in \mathbb{N}$, are called *words* of length n over the *alphabet* S. The set S is identified with S^1. The *empty word* is also admitted, it is the element of S^0. The set

$$M(S) := \bigcup_{n \in \mathbb{N}} S^n$$

is the set of all words over the set S. Let $x = (w_1 w_2 \dots w_n) \in S^n$ and $y = (v_1 v_2 \dots v_m) \in S^m$ be two words, the operation

$$x \cdot y = (w_1 w_2 \dots w_n) \cdot (v_1 v_2 \dots v_m) := (w_1 w_2 \dots w_n v_1 v_2 \dots v_m)$$

forms a new word of length $n + m$, this operation is often called the *concatenation* of the words x and y. The set $M(S)$ with the concatenation and with the empty word forms a *monoid*, i.e. a set that is closed under an associative binary operation (the concatenation) and has an identity element (the empty word).

The *normal form* also called *canonical form* of an expression is a standard way of presenting it. The normal form of elements is a substantial tool in solving the word problem, by bringing the elements to normal forms, the comparison whether two words correspond or not is then trivial.

A well–known problem is the *word problem*, i.e. does there exist an algorithm to decide whether or not two expressions in an algebra represent the same element? If there exists such an algorithm, we say the word problem is *decidable*, or *solvable*, otherwise it is *undecidable* or *unsolvable*.

M. Dehn [9] was one of the first who realised that the word problem was an important area of study. He formulated three general problems, which are of fundamental importance when handling with the presentations of groups:

1. The *identity problem*, now called the word problem. Each element of a group is defined by its composition of the generating elements. Is it possible to develop a method with a finite number of steps to decide whether an element equals the unit element or not?

2. The *transformation problem*, also called the *conjugacy problem*: given two arbitrary group elements a and b, is it possible to develop a method to decide whether one of these elements can be transformed to the other one, in other words: is it possible to find an element u of the group, so that

   $$a = u\,b\,u^{-1}$$

   holds?

3. The *group isomorphism problem*: Given two groups, are these isomorphic or not? And given a mapping of elements of one group to elements of the other group, is this mapping an isomorphism or not?

Remark 4.2.1
The conjugacy problem and the word problem are related; if in a group the conjugacy problem is solved, then the word problem can also be solved.

The Novikov–Boone Theorem

P. Novikov [50] showed that there exists a finitely generated group G such that the word problem for G is unsolvable. A different proof was obtained by W. Boone [5]. In other words, the word problem for groups is in general unsolvable.

The solvability of the word problem in a single group can be extended to a criterion for the solvability of the *uniform word problem* for a class of finitely presented groups by a straightforward argument.

4.3 Free lattices

Similar statements exist concerning lattices and lively publications on this subject can be found. We give a short summary of the main results, see also [52] and [15].

4.3.1 First results

Boolean and distributive algebras.

Every element of a Boolean algebra can be transformed to its normal form, which is unique, the word problem in Boolean algebras is thus solvable. This is also the case in distributive lattices.

Free modular lattices.

R. Dedekind [8] showed that the free modular lattice generated by three free generators is finite and has 28 elements, the word problem is thus solvable for this lattice. C. Herrmann [28] and R. Freese [13] proved the word problem is undecidable for free modular lattices with more than three free generators.

Free lattices.

The word problem for finitely presented free lattices was first solved by Skolem [54]. P. Whitman [57, 58] proved that for all equal elements in a free lattice, there is one of shortest length. He solved the word problem in free lattices, and developed an algorithm for finding normal forms. R. Freese and J.B. Nation implemented Whitman's algorithm in a computer program, see [14].

Ortholattices.

G. Bruns [6] extended the algorithm of P. Whitman to ortholattices and proved that the word problem is solvable for ortholattices.

Related to the word problem, however different, is the uniform word problem, A. Meinander [46] proved the solvability of the uniform word problem for free ortholattices.

Modular lattices.

L. Lipshitz [39] showed that the word problem for finite–dimensional projective geometries and finite modular lattices and the word problem for modular lattices are undecidable.

Orthomodular and modular–ortho lattices

Finding a solution to the word problem remains an open challenge in the *orthomodular* case as well as in the *modular–ortho* case [29]. The free orthomodular lattice over two free generators (see Figure 5.1) has 96 elements numbered by L. Beran [3]. Hence the word problem for free orthomodular lattices over two free generators is decidable. Unfortunately, this argument cannot be extended to the free lattice over three free generators.

The free orthomodular lattice on two free generators

Calculations in the free orthomodular lattice on two free generators $\mathrm{F}(a, b)$ will be our main tool throughout this research. In the next section we deal with the theory of the free orthomodular lattice on two free generators $\mathrm{F}(a, b)$. A simplified Hasse diagram of this lattice can be seen in Figure 5.1. The Greechie diagram is presented in Figure 5.4. A more exhaustive theory is treated in the book by L. Beran [3].

Chapter 5

Previous results: The free orthomodular lattice on two free generators

5.1 Fundamentals

The free orthomodular lattice on one generator is isomorphic to the Boolean algebra with four elements: $\{\mathbf{0}, x, x', \mathbf{1}\}$, written $\mathbf{2^2}$.

The free orthomodular lattice $\mathrm{F}(x,y)$ generated on two free generators x and y has 96 elements, listed and numbered by L. Beran [3]. These numbers are called *Beran's numbers* in the paper by N. Megill and M. Pavičić [43] and their subsequent papers. Beran's numbers are of practical use, e.g. for comparison. We use them throughout this thesis also as a place–holder for the corresponding operations. Sometimes we write Bn, $n = 1, \ldots, 96$, as abbreviation for "operation with Beran's number n". A complete list of all elements of $\mathrm{F}(x,y)$ can be found in the book by L. Beran [3], we put a copy in the Appendix.

Definition 5.1.1
The *lower commutator* of two lattice elements x and y is defined as

$$x \,\underline{\mathrm{com}}\, y = (x \wedge y) \vee (x' \wedge y) \vee (x \wedge y') \vee (x' \wedge y'),$$

also denoted by $c(x,y)$. Its complement, the *upper commutator* of x and y, is defined as:

$$x \,\overline{\mathrm{com}}\, y := (x \vee y) \wedge (x' \vee y) \wedge (x \vee y') \wedge (x' \vee y')$$

and denoted by $c'(x,y)$. In Boolean algebras the relation between both, upper and lower commutators is:

$$\mathbf{0} = x \,\overline{\mathrm{com}}\, y \leqslant x \,\underline{\mathrm{com}}\, y = \mathbf{1}.$$

It can be easily shown that in orthomodular lattices $x \leqslant y$ implies $x \,\mathrm{C}\, y$ (Lemma 3.2.1), and $x \,\mathrm{C}\, y$ is equivalent to $x \,\overline{\mathrm{com}}\, y = \mathbf{0}$ and $x \,\underline{\mathrm{com}}\, y = \mathbf{1}$ [37]. The elements $x \,\underline{\mathrm{com}}\, y$ and $x \,\overline{\mathrm{com}}\, y$ have the Beran's numbers 16 and 81 respectively. They have a special role which is described in [47].

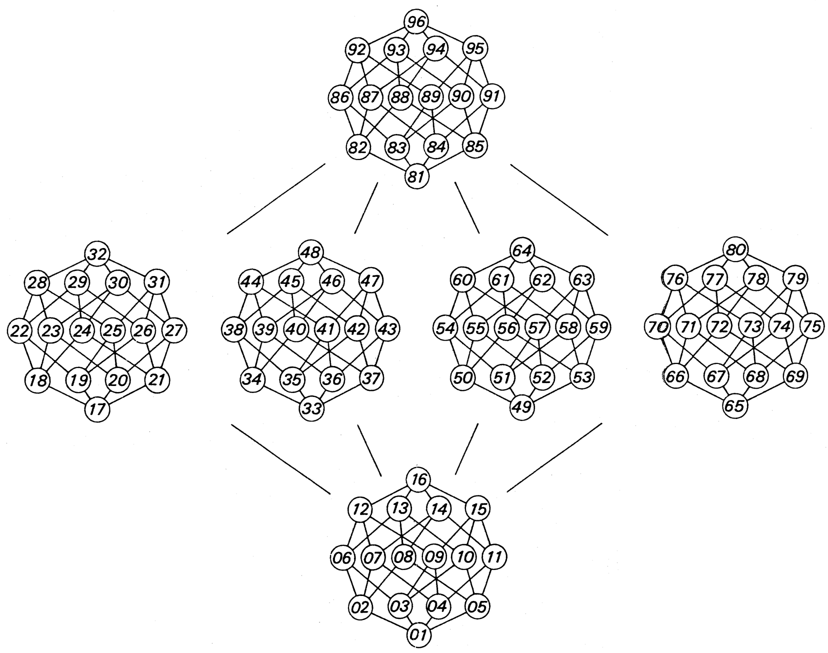


Figure 5.1: The (simplified) Hasse diagram of $\mathrm{F}(x, y)$.

Figure 5.1 represent a "simplified" Hasse diagram of $\mathrm{F}(x, y)$, *simplified* because in Hasse diagrams only the covering is denoted, here each of the "meta–edges" between six sets of elements represents sixteen coverings between the corresponding elements.

In Section 3.2.2 we introduced the Theorem of MacLaren (Theorem 3.2.6), saying that an orthomodular lattice L is isomorphic to the direct product of two intervals:

$$L \cong [\mathbf{0}, \kappa] \times [\mathbf{0}, \kappa'],$$

where κ is a central element.

An adept choice is $L := \mathrm{F}(x, y)$, $\kappa := x \,\underline{\mathrm{com}}\, y$, and $\kappa' = x \,\overline{\mathrm{com}}\, y$; since $\mathrm{F}(x, y)$ is generated by x and y, the homomorphic image $p_\kappa(\mathrm{F}(x, y)) = [\mathbf{0}, \kappa]$ is thus generated by $p_\kappa(x) = (x \wedge y) \vee (x \wedge y')$ and $p_\kappa(y) = (x \wedge y) \vee (x' \wedge y)$, they commute.

The interval $[\mathbf{0}, \kappa]$, on its turn, is isomorphic to the Boolean algebra $\mathbf{2^4}$ with two free generators, four atoms, and sixteen elements. The four atoms are: $x \wedge y$, $x' \wedge y$, $x \wedge y'$, and $x' \wedge y'$. The interval $[\mathbf{0}, \kappa']$ has four atoms $x \wedge \kappa'$, $x' \wedge \kappa'$, $y \wedge \kappa'$, $y' \wedge \kappa'$ and six elements, it is isomorphic to the orthomodular lattice MO_2 (Figure 5.2). These realisations are the cornerstone of the method of M. Navara [47].

Other elements which appear often in this thesis are the *Sasaki projection*, Beran's number 18, the element with Beran's number 23, and their analogues, like their duals and operations where the arguments are interchanged.

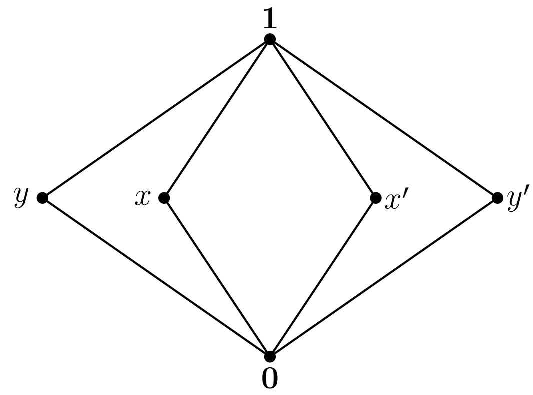


Figure 5.2: Hasse diagram of MO_2.

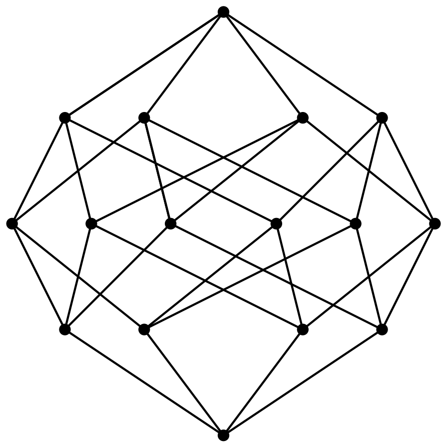


Figure 5.3: Hasse diagram of the Boolean algebra $\mathbf{2^4}$.

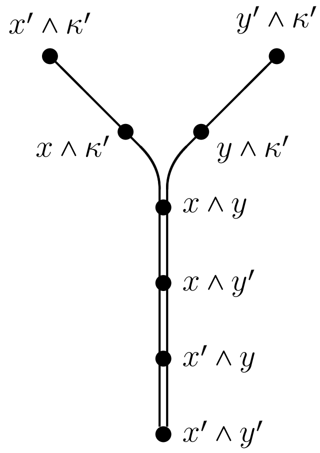


Figure 5.4: Greechie diagram of $\mathrm{F}(x, y)$.

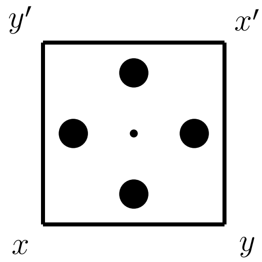


Figure 5.5: The symbol for $\mathrm{F}(x, y)$.

5.2 Computation in $\mathbf{F(x, y)}$ and the role of computers in proofs

Because the orthomodular lattice $\mathrm{F}(x, y)$ is finite, the whole computation can be made automatically, e.g., by the use of a computer program. Megill implemented this idea in a computer program[1] which, given a formula in two variables, returns the Beran's number of the corresponding element in $\mathrm{F}(x, y)$. This program was presented in [43] and extensively used in [44, 45] and other papers.

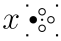
An implementation of the method of M. Navara [47] has been presented in [31] and described in [32]. It returns the Beran's number, as well as the graphical representation according to [47] and the corresponding TEX macro for its typesetting[2]. Beside simplification of formulas and testing equalities, it allows to answer questions with more complicated logical structure. It admits to introduce further variables commuting with all other variables, see Section 5.2.1. We make extensive use of the program by M. Hyčko [33], throughout our research. We automatised the program, allowing us to compute large numbers of operations in a short time.

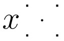
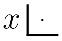
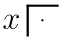
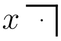
5.2.1 Computation in $\mathbf{F(x, y)}$ and its set representation

The representation of $\mathrm{F}(x, y)$, see Figures 5.1 and 5.4, is not so easy to handle. In [47] a set illustration of $\mathrm{F}(x, y)$ is described (see Figure 5.5); the orthomodular lattice MO_2 can be characterised by subsets of a set with four elements. The Boolean algebra $\mathbf{2^4}$ can be represented by the power set of a four–element set. The orthomodular lattice $\mathrm{F}(x, y)$ is then depicted as in Figure 5.5. The four discs correspond to the Boolean part and the four corners depicted by the two adjacent bars represent the MO_2 part of $\mathrm{F}(x, y)$. Full or empty discs refer to the presence or absence of the corresponding atoms in the Boolean part. The Boolean algebra $\mathbf{2^4}$ generated by two free generators x, y has atoms $x \wedge y$, $x \wedge y'$, $x' \wedge y$, and $x' \wedge y'$. It can be presented by all subsets of a four element set ⁘, the discs represent its atoms and a full or empty disc pictures the presence or absence of the corresponding atom, respectively. The generators x and y of the Boolean algebra $\mathbf{2^4}$

[1] http://us.metamath.org/downloads/quantum–logic.tar.gz

[2] The style file can be downloaded from ftp://math.feld.cvut.cz/pub/navara/foml2.sty

(or the Boolean part of $\mathrm{F}(x,y)$) are displayed by $x = x$  y and $y = x$  y, respectively. The atoms are x  y, x  y, x  y, and x  y in the same order as above.

The orthomodular lattice MO_2 with generators x, y (or the MO_2 part of $\mathrm{F}(x,y)$) is represented by the four angles , where the presence or absence of a bar indicates the presence or absence of the respective point of the set representation of MO_2. Except for the constants $\mathbf{0} = x$  y and $\mathbf{1} = x$  y, only couples of neighbouring bars are admitted, the atoms are represented by x  $y = x$, x  $y = y$, x  $y = y'$, and x  $y = x'$.

In the whole free orthomodular lattice $\mathrm{F}(x,y)$, the two parts (Boolean and MO_2) are combined so that the generators are represented by x  $y = x$ and x  $y = y$, respectively. The computation is done in both parts independently.

In orthomodular lattices the following implication holds (Lemma 3.2.1):

$$x \leqslant y \quad \Rightarrow \quad x \,\mathrm{C}\, y\,.$$

This property allows the use of the free orthomodular lattice $\mathrm{F}(x,y,c_1,\ldots,c_n)$, for a natural number n, generated by two free generators x and y and n non–free, mutually compatible, generators $c_1,\ldots,c_n$, and $c_i \mathrm{C} x$ and $c_i \mathrm{C} y$ for $i = 1,\ldots,n$. This orthomodular lattice $\mathrm{F}(x,y,c_1,\ldots,c_n)$ is isomorphic to the direct product of 2^n factors $\mathrm{F}(x,y)$ [47]

$$\mathrm{F}(x,y,c_1,\ldots,c_n) \cong \underbrace{\mathrm{F}(x,y) \times \ldots \times \mathrm{F}(x,y)}_{2^n \text{times}}\,.$$

Its operations are represented by 2^n–tuples of graphical symbols. E.g. for $n = 1$

$$\mathrm{F}(x,y,c) \cong \mathrm{F}(x,y) \times \mathrm{F}(x,y)\,.$$

The generators x, y, and c are thus expressed by the graphical notation:

$x = x\,($ ,  $)_c\, y$,

$y = x\,($ ,  $)_c\, y$,

$c = x\,($ ,  $)_c\, y$.

In Sections 6 and 7 we will need a free orthomodular lattice with three generators, two free generators, x and y, and one non–free generator c which commutes with x as well as with y. This orthomodular lattice is also described in [47] and the program by M. Hyčko allows to use up to nine non–free generators.

5.3 Orthomodular lattices versus Boolean algebras

The free orthomodular lattice $\mathrm{F}(x,y)$ is isomorphic to the product of the Boolean algebra $\mathbf{2^4}$ and the orthomodular lattice MO_2. Each Boolean operation has thus six corresponding orthomodular lattice operations, in $\mathrm{F}(x,y)$ the corresponding elements form an interval isomorphic to MO_2.

Boolean operation	Beran's number of the corresponding $\mathrm{F}(x,y)$ operations					
$\mathbf{0}$	1	17	33	49	65	81
$x \wedge y$	2	18	34	50	66	82
$x \wedge y'$	3	19	35	51	67	83
$x' \wedge y$	4	20	36	52	68	84
$x' \wedge y'$	5	21	37	53	69	85
x	6	22	38	54	70	86
y	7	23	39	55	71	87
$(x \wedge y) \vee (x' \wedge y')$	8	24	40	56	72	88
$(x \wedge y') \vee (x' \wedge y)$	9	25	41	57	73	89
y'	10	26	42	58	74	90
x'	11	27	43	59	75	91
$x \vee y$	12	28	44	60	76	92
$x \vee y'$	13	29	45	61	77	93
$x' \vee y$	14	30	46	62	78	94
$x' \vee y'$	15	31	47	63	79	95
$\mathbf{1}$	16	32	48	64	80	96

Table 5.1: Boolean operations and the Beran's numbers of their six orthomodular counterparts.

We listed the Boolean operations, together with the Beran's numbers of their corresponding orthomodular lattice counterparts in Table 5.1. E.g., the Boolean implication ($x \to y = x' \vee y$) gives rise to six quantum implications $\to_i$, $i = 0, \ldots, 5$. For $i = 1, \ldots, 5$ these can be characterized as those binary orthomodular lattice operations which satisfy the Birkhoff–von Neumann requirement [4]:

$$x \to_i y = 1 \quad \Leftrightarrow \quad x \leqslant y$$

see [37]. For $i = 0$ the implication corresponds to the classical one, for which this property fails, e.g. for the atoms x, x', y, y' of MO_2 the property $x \to_0 y = 1$ holds but not $x \leqslant y$. The implications are defined as the orthomodular lattice polynomials listed in Table 5.2. The enumeration is taken from [43] and will be maintained throughout this section.

Symbol	definition	Beran's number, name & graphical symbol		
$x \to_0 y$	$x' \vee y$	94	classical arrow	
$x \to_1 y$	$x' \vee (x \wedge y)$	78	Sasaki arrow	
$x \to_2 y$	$y \vee (x' \wedge y')$	46	Dishkant arrow	
$x \to_3 y$	$(x' \wedge y) \vee (x' \wedge y') \vee \big(x \wedge (x' \vee y)\big)$	30	Kalmbach arrow	
$x \to_4 y$	$(x \wedge y) \vee (x' \wedge y) \vee \big(y' \wedge (x' \vee y)\big)$	62	non–tolens arrow	
$x \to_5 y$	$(x \wedge y) \vee (x' \wedge y) \vee (x' \wedge y')$	14	relevance arrow	

Table 5.2: Orthomodular implications.

In Boolean algebras, all right–hand sides of these six equations coincide, and the same happens in orthomodular lattices if x, y commute.

The six implications $\to_i$, $i = 0, \ldots, 5$, give rise to the corresponding (quantum) conjunctions $\wedge_i$ and disjunctions $\vee_i$ [43] defined as:

$$\begin{aligned} x \vee_i y &= x' \to_i y\,, \\ x \wedge_i y &= (x \to_i y')'\,. \end{aligned}$$

symbol	definition	Beran's number	graphical symbol
$x \vee_0 y$	$x \vee y$	92	
$x \vee_1 y$	$x \vee (x' \wedge y)$	28	
$x \vee_2 y$	$y \vee (x \wedge y')$	44	
$x \vee_3 y$	$(x \vee y) \wedge \big(x' \vee (x \wedge y') \vee (x \wedge y)\big)$	76	
$x \vee_4 y$	$(x \vee y) \wedge \big(y' \vee (x' \wedge y) \vee (x \wedge y)\big)$	60	
$x \vee_5 y$	$(x \wedge y) \vee (x' \wedge y) \vee (x \wedge y')$	12	
$x \wedge_0 y$	$x \wedge y$	2	
$x \wedge_1 y$	$x \wedge (x' \vee y)$	18	
$x \wedge_2 y$	$y \wedge (x \vee y')$	34	
$x \wedge_3 y$	$(x \wedge y) \vee \big(x' \wedge (x \vee y') \wedge (x \vee y)\big)$	66	
$x \wedge_4 y$	$(x \wedge y) \vee \big(y' \wedge (x' \vee y) \wedge (x \vee y)\big)$	50	
$x \wedge_5 y$	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y)$	82	

Table 5.3: Orthomodular disjunctions and conjunctions.

We have presented our most important tool, the free orthomodular lattice $\mathrm{F}(x, y)$. Now, we can pass to our main work. We start, in Section 6, with the study of associativity in orthomodular lattices.

Chapter 6

Orthomodularity and associativity

6.1 First results – skew Boolean algebras

In orthomodular lattices, not only the distributive law fails, but also the commutative and associative law is not applicable in general to some binary operations which extend the Boolean join and meet. Some results for associativity in *skew Boolean algebras* can be found in the works by H. Kröger [38] and L. Beran [3].

Let $(M, \wedge, \vee, {}', \mathbf{0}, \mathbf{1})$ be an orthomodular lattice, we define the *skew meet* ($\curlywedge$) and *skew join* ($\curlyvee$) for the two elements x and $y \in M$ as follows:

$$x \curlywedge y \ := \ (x \vee y') \wedge y \qquad \text{Beran's number 34,}$$
$$x \curlyvee y \ := \ (x \wedge y') \vee y \qquad \text{Beran's number 44,}$$

respectively.
These two skew operations give rise to a new algebra $(M, \curlywedge, \curlyvee, {}', \mathbf{0}, \mathbf{1})$, which is called the *skew Boolean algebra* associated to the orthomodular lattice $(M, \wedge, \vee, {}', \mathbf{0}, \mathbf{1})$.

Remark 6.1.1
In Boolean algebras both skew operations are equal to the classical ones, $x \curlywedge y = x \wedge y$ and $x \curlyvee y = x \vee y$.

Kröger [38] showed that the presence of commuting elements is a sufficient condition for associativity in skew Boolean lattices.

Proposition 6.1.2 (Kröger [38])
If both $x \,\mathrm{C}\, y$ and $y \,\mathrm{C}\, z$ hold, then

$$(x * y) * z = x * (y * z)$$

where $*$ is the skew meet or the skew join.

We will prove later (Corollary 6.3.2) that $y \, \mathrm{C} \, z$ is already sufficient for associativity of the skew join and the skew meet.

Proposition 6.1.3 (Megill, Pavičić [43] and D'Hooghe, Pykacz [10])
Let x, y and z be elements of an orthomodular lattice such that one of them commutes with the other two. If $* \in \{\wedge_i, \vee_i \mid i = 0, \ldots, 5\}$ (see Section 5.3, Table 5.3), then the associativity identity $(x * y) * z = x * (y * z)$ holds.

Proof:
See [10] for $i = 1, 2, 5$ and [43] for $i = 3, 4$. The case $i = 0$ leads to the lattice operations $\wedge_0 = \wedge$ and $\vee_0 = \vee$ which are known to be associative.

q.e.d.

6.2 Associativity of binary operations

In this section, we will explore which binary orthomodular lattice operations among the 96 ones are associative.

Let us start by summarizing the situation in Boolean algebras.

Proposition 6.2.1
There are $\mathbf{2^4} = 16$ binary Boolean operations. Eight of them are associative: The disjunction $\vee$, the conjunction $\wedge$, the equivalence $\leftrightarrow$ and its complement $\nleftrightarrow$ (XOR),

$$x \leftrightarrow y = \begin{cases} 1 & \text{if } x = y \,, \\ 0 & \text{otherwise} \,, \end{cases}$$

$$x \nleftrightarrow y = \begin{cases} 0 & \text{if } x = y \,, \\ 1 & \text{otherwise} \,, \end{cases}$$

also the two constants $0, 1$, and the two projections $\triangleleft$, $\triangleright$,

$$\begin{aligned} x \triangleleft y &= x \,, \\ x \triangleright y &= y \,. \end{aligned} \tag{6.1}$$

In general, we have to distinguish a Boolean operation from its six corresponding orthomodular lattice operations, they have the same appearance as one of the six orthomodular counterparts, e.g., the projections $\triangleleft$, $\triangleright$, defined by (6.1) are also defined in orthomodular lattices. Table 6.1 summarizes these cases.

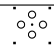
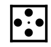
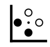
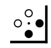
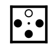
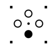
Symbol	Orthomodular lattice operation	Beran's number	Graphical symbol
0	least element	1	
1	greatest element	96	
◁	left projection	22	
▷	right projection	39	
∨	join	92	
∧	meet	2	

Table 6.1: The six associative orthomodular lattice operations.

Proposition 6.2.2
The six orthomodular operations in Table 6.1 are associative.

Proof:
The lattice operations ∨, ∧ are known to be associative by definition. The other four operations are trivially associative.

q.e.d.

The following observation reduces our work to one half:

Proposition 6.2.3
In the special case of a Boolean algebra, each orthomodular binary operation $*$ reduces to a unique Boolean operation $\square$. If $*$ is associative (in orthomodular lattices), so is $\square$ (in Boolean algebras).

This means also that the orthomodular counterparts of a nonassociative Boolean operation cannot be associative in an orthomodular lattice. There are eight nonassociative Boolean operations and as many associative ones, thus, in an orthomodular lattice there are up to 48 $(= 96 - 8 \times 6)$ possibly associative operations. They are listed in Table 6.2. Most of them are nonassociative, as will be proven later. The period 16 is due to the system of Beran's numbers.

Boolean operation	Beran's numbers of corresponding orthomodular lattice operations					
constant 0	1	17	33	49	65	81
constant 1	16	32	48	64	80	96
left projection ◁	6	22	38	54	70	86
right projection ▷	7	23	39	55	71	87
conjunction ∧	2	18	34	50	66	82
equivalence ↔	8	24	40	56	72	88
non–equivalence ↮	9	25	41	57	73	89
disjunction ∨	12	28	44	60	76	92

Table 6.2: The orthomodular operations extending the eight associative Boolean operations.

For the special case that only two variables are present, six different cases of associativity can be found:

$$(x * x) * y = x * (x * y)\,, \tag{6.2}$$
$$(x * x') * y = x * (x' * y)\,, \tag{6.3}$$
$$(x * y) * y = x * (y * y)\,, \tag{6.4}$$
$$(x * y') * y = x * (y' * y)\,, \tag{6.5}$$
$$(x * y) * x = x * (y * x)\,, \tag{6.6}$$
$$(x * y) * x' = x * (y * x')\,. \tag{6.7}$$

An example of such approach is as follows:

Example 6.2.4
Let x, y be non–commuting atoms of MO_2. Let $*$ be one of the orthomodular binary operations which, restricted to x and y, acts as the left projection,

$$x * y = x\,.$$

This occurs only if the Beran's number of $*$ is in $\{17, 18, \ldots, 32\}$. Then the right–hand side of (6.6) is

$$x * (y * x) = x * y = x$$

and (6.6) reduces to

$$x * x = x\,.$$

This means that $*$, acting on the sub–orthomodular lattice generated by x, must be idempotent. This sub–orthomodular lattice is $\{0, x, x', 1\}$, i.e., the free Boolean algebra $\mathbf{2^2}$ with one generator x. We see that the Boolean counterpart of $*$ must be idempotent.
Among associative operations on Boolean algebras, idempotence holds for $\wedge, \vee, \lhd, \rhd$ and does not hold for $0, 1, \leftrightarrow, \not\leftrightarrow$. For the corresponding orthomodular lattice operations (denoted by Beran's numbers) this means that each of the operations with Beran's numbers 18, 22, 23 and 28 may (and do) satisfy (6.6), while those with Beran's number 17, 24, 25 and 32 violate (6.6) and are nonassociative. Other operations from $\{17, 18, \ldots, 32\}$ were already excluded because their Boolean counterparts are not associative.

In view of Example 6.2.4, among operations with Beran's numbers in $\{17, 18, \ldots, 32\}$, only four of them, B18, B22, B23 and B28, remain candidates which have to be tested further. For this, we may use the equations (6.2) to (6.5) or (6.7). This requires only computations in two variables which can be performed by a program [31]. Only those operations which pass these tests need to be checked by more general associativity criteria. Among them, the left projection $\lhd$ (B22), is known to be associative.

Proposition 6.2.5
The following operations violate at least one of equations (6.2) to (6.7):

1. The operations $*$ with Beran's numbers 6, 12, 24, 25, 50, 54, 72 violate (6.2).
2. The operations $*$ with Beran's numbers 55, 60, 66, 70, 76, 89 violate (6.3).
3. The operations $*$ with Beran's numbers 7, 32, 65, 71, 80, 82 violate (6.4).
4. The operations $*$ with Beran's numbers 8, 9, 40, 41, 73, 87, 88 violate (6.5).
5. The operations $*$ with Beran's numbers 17, 33, 48, 49, 56, 57, 64 violate (6.6).
6. The operations $*$ with Beran's numbers 18, 23, 28, 34, 38, 44, 86 violate (6.7).

Remark 6.2.6
To prove this and also some following theorems in this work, we use M. Hyčko's program [33]. The input line we used here is `B3(*,x,B3(*,x,y)) = B3(*,B3(*,x,x),y)` for the equation (6.2) and suitable for other five equations. The letter "`B`" stands for Beran, "`3`" indicates the arity of the operation[1], "$*$" stands for the Beran's number of the operation on the two following arguments. The program returns "`True`" if the equation holds, "`False`" otherwise.

Proof:
As described in Remark 6.2.6, the proof uses the computational tool [33], as well as arguments analogous to those of Example 6.2.4. The results are summarised in Table 6.3. There are eight operations, marked with ☆, which fulfil all six equations. Operations which do not appear in Table 6.2 or for which none of the six equations holds are not listed. The coloured entries indicate pairs of "`True`" values, i.e. equations (6.2) and (6.3), equations (6.4) and (6.5) and the equations (6.6) and (6.7). This will be dealt with in the next Section 6.3.1.

		Equation					
	Beran's number	(6.2) xxy	(6.3) xx'y	(6.4) xyy	(6.5) xy'y	(6.6) xyx	(6.7) xyx'
☆	1	True	True	True	True	True	True
☆	2	True	True	True	True	True	True
	6	False	False	True	True	False	False
	7	True	True	False	False	False	False
	8	False	False	False	False	True	False
	9	False	False	False	False	True	False
	12	False	False	False	False	True	False
☆	16	True	True	True	True	True	True
	17	True	True	False	False	False	False
	18	True	True	True	False	True	False
☆	22	True	True	True	True	True	True
	23	True	True	True	True	True	False

[1]Here we mean the arity from the point of view of the user of the program, i.e., the number of entries following a call of a function, including the Beran's number. From the mathematical point of view, this represents a binary operation (whose entries are the second and third arguments in the expression).

	Beran's number	Equation (6.2) xxy	(6.3) xx'y	(6.4) xyy	(6.5) xy'y	(6.6) xyx	(6.7) xyx'
	24	False	False	True	False	False	False
	25	False	False	True	False	False	False
	28	True	True	True	False	True	False
	32	True	True	False	False	False	False
	33	False	False	True	True	False	False
	34	True	False	True	True	True	False
	38	True	True	True	True	True	False
☆	39	True	True	True	True	True	True
	40	True	False	False	False	False	False
	41	True	False	False	False	False	False
	44	True	False	True	True	True	False
	48	False	False	True	True	False	False
	49	False	False	True	True	False	False
	50	False	False	False	False	False	True
	54	False	False	True	True	False	True
	55	False	False	False	False	True	True
	56	True	False	False	False	False	False
	57	True	False	False	False	False	False
	60	False	False	False	False	False	True
	64	False	False	True	True	False	False
	65	True	True	False	False	False	False
	66	False	False	False	False	False	True
	70	False	False	False	False	True	True
	71	True	True	False	False	False	True
	72	False	False	True	False	False	False
	73	False	False	True	False	False	False
	76	False	False	False	False	False	True
	80	True	True	False	False	False	False
☆	81	True	True	True	True	True	True
	82	False	False	False	False	True	False
	86	False	False	True	True	False	False
	87	True	True	False	False	False	False
	88	False	False	False	False	True	False
	89	False	False	False	False	True	False
☆	92	True	True	True	True	True	True
☆	96	True	True	True	True	True	True

Table 6.3: Results for the proof of Proposition 6.2.5.

q.e.d.

The approach of Proposition 6.2.5 disproved associativity of 40 orthomodular operations, but it is not applicable to the lower commutator $x \underline{\mathrm{com}}\, y$, and the upper commutator $x \overline{\mathrm{com}}\, y$. Both satisfy all equations (6.2) to (6.7) and another approach is needed.

Proposition 6.2.7
The lower and upper commutator, $\underline{\mathrm{com}}$, respectively $\overline{\mathrm{com}}$, are not associative.

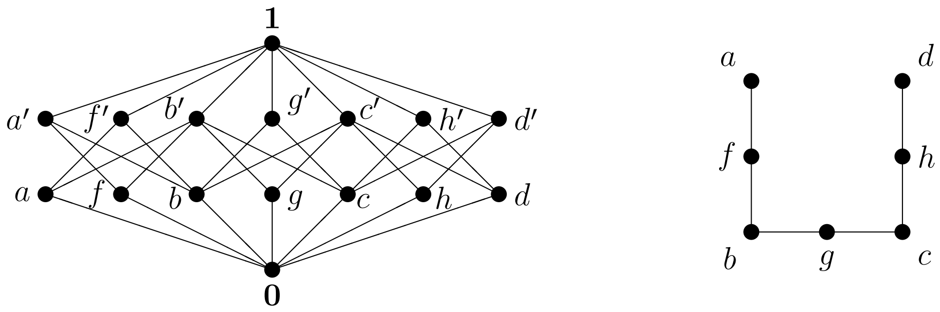


Figure 6.1: Hasse and Greechie diagram of the orthomodular lattice D_{16}.

Proof:
To prove this proposition, the Dilworth orthomodular lattice D_{16} is used (Figure 6.1). Then

$$\begin{aligned}(a \,\underline{\mathrm{com}}\, g) \,\underline{\mathrm{com}}\, d &= b \,\underline{\mathrm{com}}\, d = c\,,\\ a \,\underline{\mathrm{com}}\, (g \,\underline{\mathrm{com}}\, d) &= a \,\underline{\mathrm{com}}\, c = b \neq (a \,\underline{\mathrm{com}}\, g) \,\underline{\mathrm{com}}\, d\,.\end{aligned}$$

A similar argument works for the upper commutator:

$$\begin{aligned}(a \,\overline{\mathrm{com}}\, g) \,\overline{\mathrm{com}}\, d &= b' \,\overline{\mathrm{com}}\, d = c'\,,\\ a \,\overline{\mathrm{com}}\, (g \,\overline{\mathrm{com}}\, d) &= a \,\overline{\mathrm{com}}\, c' = b' \neq (a \,\overline{\mathrm{com}}\, g) \,\overline{\mathrm{com}}\, d\,.\end{aligned}$$

q.e.d.

The result of this section is:

Theorem 6.2.8
The only associative operations in orthomodular lattices are the six operations in Table 6.1.

Proof:
This is just a combination of the previous propositions which cover all 96 possible cases. Based on Proposition 6.2.1, Proposition 6.2.3 excludes 48 operations. Proposition 6.2.5 rejects further 40 possibilities and Proposition 6.2.7 disproves associativity of the two commutators (not covered by the preceding propositions). What remains are the six associative operations from Proposition 6.2.2.

q.e.d.

6.3 Associativity of binary operations, by using commutation

We found six operations being associative without any constraints. The next question which comes up is: are there other operations which satisfy the associativity identity under the assumption that some arguments commute?

The operations listed in Table 6.3 are associative in at least one of the equations (6.2) to (6.7), some only in one equation, others in all six of them. Six operations are known to be associative. Then we may impose any of the three commutation assumptions, $x \mathrel{C} y$, $x \mathrel{C} z$ and $y \mathrel{C} z$. All subsets of these three conditions form a lattice (the Boolean algebra $\mathbf{2^3}$ with eight elements, see Figure 6.2).

An *ideal* in a bounded lattice L is a non–empty subset I such that $x, y \in I$ implies $x \vee y \in I$ and from $x \in I$, $z \leqslant x$ follows $z \in I$. An ideal generated by a single element of L is called a *principal ideal*. The dual concept of ideal is *filter*, a *principal filter* is defined analogously.

Each positive result obtained for some subset of commutation assumptions applies to the respective principal filter (of assumptions stronger than or equal to the original one). Each negative result obtained for some subset of assumptions applies to the respective principal ideal (of assumptions weaker than or equal to the original one).

The aim is to describe the situation for all eight possible sets of compatibility assumptions. To shorten their description, we cover the whole of $\mathbf{2^3}$ by principal filters and ideals and we only mention the weakest sufficient conditions for positive results and the strongest insufficient conditions for negative results (i.e., the strongest assumptions which still admit a counterexample). E.g., if we find out a counterexample for $y \mathrel{C} z$ and positive results assuming $x \mathrel{C} y$ or $x \mathrel{C} z$, this gives answer to all possible combinations of compatibility assumptions. The filled circles in Figure 6.2 denote the assumptions under which equalities hold, empty circles the remaining cases. The **1** represents the case that all variables commute, then all 48 operations of Table 6.2 satisfy the associativity identity. The **0** represents the case that none of the variables commute namely the six associative operations of Theorem 6.2.8.

6.3.1 Conditional associativity by using commutation of one pair of arguments

We intend to find those operations from Table 6.2 which fulfil the associative identity when only two of the three arguments commute. In Table 6.3 there are 42 orthomodular lattice operations (without the six associative operations) for which at least one of the six equations (6.2) to (6.7) holds.

Having a closer look at the six equations (6.2) to (6.7), we remark that we need not consider all 42 operations. The six equations can be grouped in three pairs having the commuting elements at the same place. In other words, for some cases the calculations in $\mathrm{F}(x, y)$ itself give some counterexamples. The equations (6.2) and (6.3), equations (6.4) and (6.5) and the equations (6.6) and (6.7) form three pairs. If the results contain at

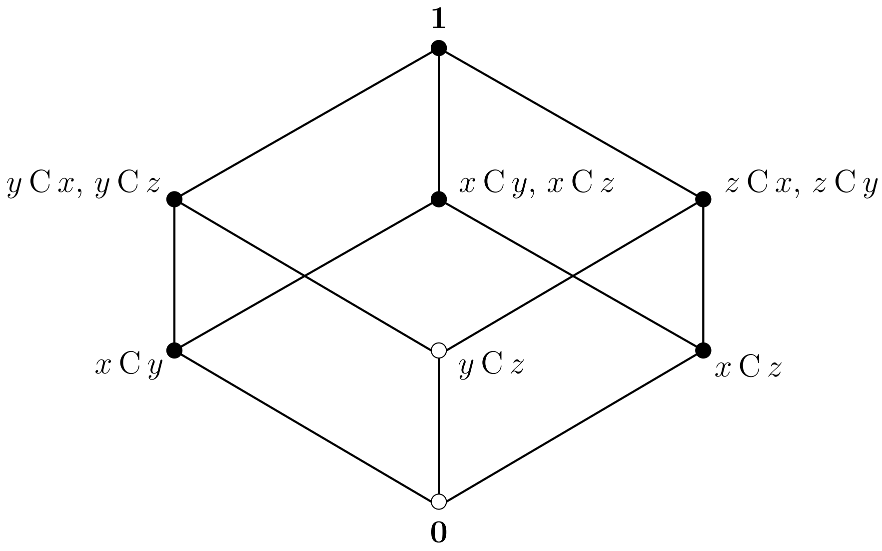


Figure 6.2: Hasse diagram of lattice $\mathbf{2^3}$ of compatibility assumptions.

least one "False'' in the respective columns of Table 6.3, it is already a counterexample. Only if a pair has the same outcome "True'' there is a chance that associativity equation holds for the respective weaker commutation assumption. The Beran's numbers of these operations, the six associative operations not included, are highlighted in purple.

In Table 6.3 the remaining 30 cases (24 operations) are highlighted in purple. These operations need further checks, their Beran's numbers are

For $x \,\mathrm{C}\, y$ $7, 16, 17, 18, 23, 28, 32, 38, 65, 71, 80, 81, 87$

For $y \,\mathrm{C}\, z$ $6, 16, 23, 33, 34, 38, 44, 48, 49, 54, 64, 81, 86$

For $x \,\mathrm{C}\, z$ $16, 55, 70, 81$

Fortunately, we can reduce the amount of checks to more than half of them due to the presence of dual operations and operations in which the arguments are interchanged. We listed them in Table 6.4 by their Beran's numbers. If we find a counterexample for one operation, for instance the one whose Beran's number is in the first column, then it will also be possible to find counterexamples for the operations with Beran's number listed on the same line. The same holds for argumentation that the operation fulfils the associativity identity under some constraints. In the last case attention has to be taken to the correct commuting arguments.

$x * y$	$y * x$	$(x' * y')'$	$(y' * x')'$
6	7	86	87
16	16	81	81
17	33	32	48
18	34	28	44
23	38	23	38

$x * y$	$y * x$	$(x' * y')'$	$(y' * x')'$
49	65	64	80
54	71	54	71
55	70	55	70

Table 6.4: The duals and expressions with interchanged arguments, by their Beran's numbers.

We checked the remaining seven operations in the Dilworth's lattice D_{16} (Figure 6.1). Only to find a counterexample for the operation with Beran's numbers 23 and 54 we used the lattice L_{22} (Figure 6.3)[2]. We could easily find counterexamples, not only for the two commutators, but also for most of the other operations with partial positive results in Table 6.3. The results are showed in Table 6.5, together with our choice of arguments.

$a * b$	commuting elements	choice for x	choice for y	choice for z	result for $(x * y) * z$	result for $x * (y * z)$	in lattice[3]
6	$y \, \mathrm{C} \, z$	a'	c'	d	0	b	D_{16}
16	$x \, \mathrm{C} \, y$	d	c	a	1	c	D_{16}
16	$x \, \mathrm{C} \, z$	d	a	c	1	c	D_{16}
16	$y \, \mathrm{C} \, z$	a	c	d	c	1	D_{16}
17	$x \, \mathrm{C} \, y$	d	c'	a	0	d	D_{16}
23	$x \, \mathrm{C} \, y$	a	b'	d	g	a	L_{22}
23	$y \, \mathrm{C} \, z$	a	c'	d	f'	a	D_{16}
49	$y \, \mathrm{C} \, z$	a	c	d	h	0	D_{16}
54	$y \, \mathrm{C} \, z$	a'	c'	d	d'	g'	D_{16}
55	$x \, \mathrm{C} \, z$	a	h	b	g	c	D_{16}

Table 6.5: Counterexamples for operations of Table 6.3, if only two arguments commute.

The Sasaki projection is not listed in Table 6.5, we can prove the associativity of the Sasaki projection in case that the first and second argument commute.

Theorem 6.3.1 (Gagola III, Gabriëls, Navara [18])
Let L be an orthomodular lattice and let $*$ be the Sasaki projection, the operation with the Beran's number 18. If x, y ,$z \in L$ such that x and y commute then[4]

$$x * (y * z) = (x * y) * z \, .$$

[2]The elements a and a' are displayed twice because, as can be seen in the Greechie diagram, the figure closes at a, a'. In this way the representation gets well–arranged.

[3]All counterexamples could be found in the lattice L_{22}, we tried to use only the "smaller" lattice D_{16} which was not always possible.

[4]B. D'Hooghe and J. Pykacz [10] proved the associativity of the Sasaki projection, under the assumption that one argument commutes with the other two. We prove that $x \, \mathrm{C} \, y$ is sufficient for the associativity of B18.

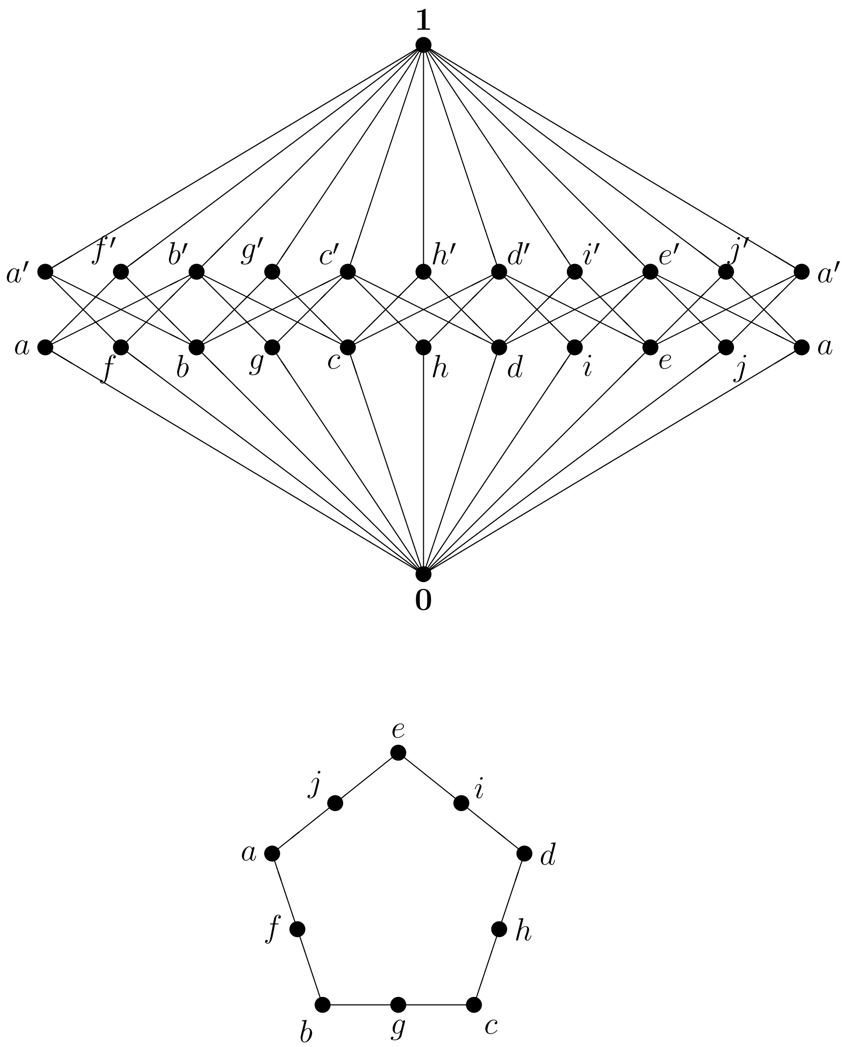


Figure 6.3: Hasse and Greechie diagrams of L_{22}.

Proof:
By Lemma 3.2.2 (*ii*) the commutation relation $x \,\mathrm{C}\, y$ is equivalent to $x * y = x \wedge y$, hence

$$\begin{aligned}(x * y) * z &= (x * y) \wedge \big((x * y)' \vee z\big)\\ &= (x * y) \wedge \big((x \wedge y)' \vee z\big)\\ &= (x * y) \wedge (x' \vee y' \vee z)\\ &= x \wedge (x' \vee y) \wedge (x' \vee y' \vee z)\,.\end{aligned}$$

and

$$\begin{aligned}x * (y * z) &= x * \big(y \wedge (y' \vee z)\big)\\ &= x \wedge \Big(x' \vee \big(y \wedge (y' \vee z)\big)\Big)\end{aligned}$$

Because $y' \leqslant y' \vee z$, it follows by Lemma 3.2.1 that $y' \,\mathrm{C}\, (y' \vee z)$ and by Proposition 3.2.3

also that $y' \,\mathrm{C}\, x'$. Likewise, y commutes with both x' and $y' \vee z$. Thus, by the Foulis–Holland Theorem,

$$x' \vee \big(y \wedge (y' \vee z)\big) = (x' \vee y) \wedge (x' \vee y' \vee z)\,.$$

It then follows that

$$\begin{aligned} x * (y * z) &= x \wedge \big(x' \vee \big(y \wedge (y' \vee z)\big)\big) \\ &= x \wedge (x' \vee y) \wedge (x' \vee y' \vee z)\,. \end{aligned}$$

q.e.d.

Corollary 6.3.2
Let $\star$ be the dual of the Sasaki projection, the operation with Beran's number 28, and $x \,\mathrm{C}\, y$, then

$$x \star (y \star z) = (x \star y) \star z\,.$$

By Lemma 3.2.2 (*iii*) there is an equivalence between $x \,\mathrm{C}\, y$ and $x \star y = x \vee y$. The proof is thus similar to the proof of Theorem 6.3.1 by duality.

Corollary 6.3.3
Let $\circ$ be the operation with Beran's number 34 or the operation with Beran's number 44, the skew meet or skew join respectively. Let $y \,\mathrm{C}\, z$, then

$$x \circ (y \circ z) = (x \circ y) \circ z\,.$$

H. Kröger [38] proved the associativity of both skew operations, but under stronger conditions, he required the commutation of one variable with the other two. We can prove by interchanging the variables in Theorem 6.3.1 that $y \,\mathrm{C}\, z$ is a sufficient condition.

6.3.2 Associativity for one argument commuting with the two others

The next question which rises at this point is: are there operations fulfilling the associativity identity if one argument commutes with the other two? More explicitly, we look for the fulfillment of the associativity identity under the conditions:

$$x \,\mathrm{C}\, y \text{ and } x \,\mathrm{C}\, z \qquad (\mathrm{C}x)$$
$$y \,\mathrm{C}\, x \text{ and } y \,\mathrm{C}\, z \qquad (\mathrm{C}y)$$
$$z \,\mathrm{C}\, x \text{ and } z \,\mathrm{C}\, y \qquad (\mathrm{C}z)$$

N. Megill and M. Pavičić proved this for the operations with the Beran's numbers in {12, 18, 28, 34, 44, 82} and B. D'Hooghe and J. Pykacz proved it for the Beran's numbers $\{50, 60, 66, 76\}$ (Proposition 6.1.3). Again the program of M. Hyčko [33] yields the results, they are listed in Table 6.6. "True" means the equation $(x * y) * z = x * (y * z)$ holds for the operation $*$, "False" if it does not. The table contains 42 orthomodular operations from Table 6.2, which are associative in Boolean algebras but failed to be associative in general orthomodular lattices.

$*$	$y \, C \, x$, $y \, C \, z$ (Cy)	$z \, C \, x$, $z \, C \, y$ (Cz)	$x \, C \, y$, $x \, C \, z$ (Cx)
6	False	True	True
7	False	True	True
8	True	False	False
9	True	False	False
12	True	True	True
16	True	True	True
17	True	True	True
18	True	True	True
23	False	True	True
24	False	False	True
25	False	False	True
28	True	True	True
32	True	True	True
33	True	True	True
34	True	True	True
38	False	True	True
40	False	True	False
41	False	True	False
44	True	True	True
48	True	True	True
49	True	True	True
50	True	True	True
54	False	True	True
55	False	True	True
56	False	True	False
57	False	True	False
60	True	True	True
64	True	True	True
65	True	True	True
66	True	True	True
70	False	True	True
71	False	True	True
72	False	False	True
73	False	False	True
76	True	True	True
80	True	True	True

*	$y \mathrm{C} x$, $y \mathrm{C} z$ ($\mathrm{C}y$)	$z \mathrm{C} x$, $z \mathrm{C} y$ ($\mathrm{C}z$)	$x \mathrm{C} y$, $x \mathrm{C} z$ ($\mathrm{C}x$)
81	True	True	True
82	True	True	True
86	False	True	True
87	False	True	True
88	True	False	False
89	True	False	False

Table 6.6: Validity of the associativity identity for operations, if one variable commutes with the other two.

All the computations are done in the free orthomodular lattice on three generators, one commutes with the two others. This lattice is finite and the computations can be done with the program of M. Hyčko [33]. Due to the properties of free algebras on non–free generators, see Section 4, the results (both positive and negative) extend to all orthomodular lattices.

6.4 Conclusions

All 48 operations from Table 6.2 fulfil the associativity identity,

$$(x * y) * z = x * (y * z)$$

under different commuting relation conditions.

Six operations are associative without any need of the commutation relation between the arguments (Theorem 6.2.8). They correspond to the $\mathbf{0}$ in the Boolean lattice $\mathbf{2^3}$ of Figure 6.2.

Four operations fulfil the equation, if one variable commutes with a particular one. The condition $x \mathrm{C} y$ is sufficient for the Sasaki projection and its dual. The conditions ($\mathrm{C}x$) or ($\mathrm{C}y$) are thus also fulfilled for them, the condition ($\mathrm{C}z$) is proved by Table 6.6. For the skew meet and skew join, the sufficient condition is $y \mathrm{C} z$. The conditions ($\mathrm{C}y$) or ($\mathrm{C}z$) are thus also fulfilled, the condition ($\mathrm{C}x$) is proven by Table 6.6.

The other operations need at least two pairs of commuting elements: For twenty orthomodular lattice operations, it does not matter which argument commutes with the two others. Among them the Sasaki projection, its dual and the two skew operations, see before.

The orthomodular operations with Beran's numbers 24, 25, 27 and 73 fulfil the associativity identity only in case that ($\mathrm{C}x$). The operations with Beran's numbers 8, 9, 88 and 89 fulfil the associativity identity only in case that ($\mathrm{C}y$) and the operations with Beran's numbers 40, 41, 56 and 57 fulfil it only when ($\mathrm{C}y$).

The ten remaining operations fulfil the associativity identity in two cases, ($\mathrm{C}x$) or ($\mathrm{C}y$).

Chapter 7

Monotonicity of binary operations

The absence of the distributivity law in orthomodular lattices is a handicap to bring complex expressions in simpler normal forms. Commuting elements seem to be indispensable. However, is it possible to find two operations out of the 96 binary orthomodular lattice ones, which are distributive? Is it possible to find operations $*$ and $\circ$ for which

$$(x_1 * y) \circ (x_2 * y) \circ \ldots \circ (x_n * y) = (x_1 \circ \ldots \circ x_n) * y$$

holds? Which properties do they need to have?

We need to find two operations which satisfy the distributive law, one of them (“ $\circ$”) has to be commutative and associative as well. As a result of Section 6, for the latter conditions only the lattice join or meet are possible candidates, as they fulfil the commutative as well as the associative law. There are other commutative and associative operations, the constants **0** and **1**, but they do not depend on both arguments. Such operations are of no use in finding normal forms.

By the duality principle, it is not relevant for which, join or meet, we consider the last equation. We decided to find the operations which distribute over the meet:

$$(x_1 * y) \wedge (x_2 * y) \wedge \ldots \wedge (x_n * y) = (x_1 \wedge \ldots \wedge x_n) * y\,.$$

In both cases operations which come into question have to be monotonic. That is, for which of the 96 binary orthomodular lattice operations $*$ does

$$x \leqslant y \quad \Rightarrow \quad x * z \leqslant y * z \tag{7.1}$$

hold?

The right–hand side of implication (7.1) can be expressed as an equation

$$(x * z) \wedge (y * z) = x * z \quad \forall\, z$$

or

$$(x * z) \vee (y * z) = y * z\,.$$

Implication (7.1) is equivalent to any of the following conditions which have the form of equations:

1. Substitution $y := x \vee w$ gives
$$(x * z) \wedge \big((x \vee w) * z\big) = x * z\,. \tag{7.2}$$
2. Substitution $x := y \wedge v$ gives
$$\big((y \wedge v) * z\big) \vee (y * z) = y * z\,. \tag{7.3}$$

Since the handling of both variables is the same, we handle the monotonicity of the first variable and disregard the second variable at first. We also concentrate on the non–decreasing operations, because the non–increasing ones can be characterized analogously by taking the orthocomplements.

7.1 Sufficient conditions

Lattice operations are non–decreasing in both arguments. The negation can produce an operation which violates non–decreasingness property. Thus implication (7.1) is satisfied by all binary orthomodular lattice operations $*$ which can be expressed in a form not negating the first argument. E.g., the operation

$$x * y = (x \wedge y) \vee (x \wedge y')$$

is non–decreasing in the first argument. On the contrary, the operation

$$x * y = (x \wedge y) \vee (x' \wedge y')$$

or

$$x * y = (x \wedge y) \vee (x \vee y)'$$

need not be non–decreasing in the first argument because the argument x is negated. The operation

$$x * y = (x \wedge y) \vee (x' \wedge y)'$$

is defined using the orthocomplement applied to x, but it has an equivalent form

$$x * y = (x \wedge y) \vee (x \vee y')$$

which shows that it is non–decreasing in the first argument.

There are 17 binary orthomodular lattice operations satisfying this condition:

Proposition 7.1.1
The binary orthomodular lattice operations with Beran's numbers 1, 2, 3, 6, 22, 34, 38, 39, 44, 51, 54, 58, 61, 86, 92, 93, and 96 are non–decreasing in the first argument.

7.2 Necessary conditions

After this, we asked what are the necessary conditions. Therefore, we proceed similarly to the Sieve of Eratosthenes for prime numbers, and sort the non–monotonic operations out of the list of all 96 binary orthomodular lattice operations, only the monotonic ones will be kept.

The following techniques can be used: techniques in

- the Boolean part of operations,
- the MO_2 part of operations,
- the free orthomodular lattice with three (non–free) generators,
- the free orthomodular lattice with two free generators,
- the Kalmbach embedding.

It will turn out that the Kalmbach embedding already is sufficient to identify all non–monotone operations.

7.2.1 The use of free algebras – general approach

Equation (7.2), resp. (7.3), holds if and only if it is satisfied in the free orthomodular lattice with three free generators x, z, w, respectively y, z, v. However, the free orthomodular lattice with three free generators has a very complex structure (see [37, p. 229]). Its strange properties are studied in [25]. The direct use of the free orthomodular lattice with three free generators is thus impossible. Nevertheless, it is possible to derive necessary conditions with additional assumptions, which make the computation possible. The use of a free orthomodular lattice with three generators, which are *not free*, leads to *necessary conditions* for monotonicity. The free orthomodular lattice on more than two generators is described in [47]. Two generators are free and the others commute with the two free generators and among each other.

7.2.2 The Boolean part of operations

Here it is assumed that all three generators are compatible. In the form of equations, compatibility of x and y can be written equivalently, $x \underline{\mathrm{com}}\, y = 1$ or $x \overline{\mathrm{com}}\, y = 0$.

Compatibility of x, y and z can be written as

$$(x \underline{\mathrm{com}}\, y) \wedge (x \underline{\mathrm{com}}\, z) \wedge (y \underline{\mathrm{com}}\, z) = 1 \tag{7.4}$$

or, equivalently,

$$(x \overline{\mathrm{com}}\, y) \vee (x \overline{\mathrm{com}}\, z) \vee (y \overline{\mathrm{com}}\, z) = 0\,. \tag{7.5}$$

Note that in orthomodular lattices, unlike orthomodular posets, pairwise compatibility implies compatibility of the whole finite set. I.e., there is a Boolean subalgebra containing all these elements; see [3] for more details.

The free orthomodular lattice with three generators x, y, z satisfying (7.4) (equivalently, (7.5)) is the free Boolean algebra with three generators x, y, z. This algebra is finite and has well–known properties, hence it is easy to verify (7.2) and (7.3).

The candidates are the 16 binary Boolean operations. Each of them corresponds to six orthomodular lattice operations, see [43] and Section 5.3.

Among binary Boolean operations, the following seven violate (7.1) (the MO_2 part in the graphical symbols of the Boolean part in Table 7.1 is omitted):

Beran's no.	expression	graphical notation
4	$x' \wedge y$	x ○○●○ y,
5	$x' \wedge y'$	x ●○○○ y,
8	$(x \wedge y) \vee (x' \wedge y')$	x ●○○● y,
9	$(x \wedge y') \vee (x' \wedge y)$	x ○●●○ y,
75	x'	x ●○●○ y,
94	$x' \vee y$	x ●○●● y,
95	$x' \vee y'$	x ●●●○ y.

Table 7.1: Boolean operations not non–decreasing in the first argument.

If a binary Boolean operation violates (7.1), so do all six corresponding orthomodular lattice operations, independently of their MO_2 part. This principle proves that at least $6 \times 7 = 42$ binary orthomodular lattice operations violate (7.1).

The remaining 54 operations correspond to the nine Boolean operations listed in Table 7.2.

Beran's no.	expression	graphical notation
1	0	x ○○○○ y,
2	$x \wedge y$	x ○●●● y,
3	$x \wedge y'$	x ●●○● y,
22	x	x ○●○● y,
39	y	x ○○●● y,
58	y'	x ●●○○ y,
92	$x \vee y$	x ○○○● y,
93	$x \vee y'$	x ○●○○ y,
96	1	x ●●●● y.

Table 7.2: Boolean operations non–decreasing in the first argument.

In Table 7.3 the Beran's numbers of all orthomodular lattice operations are listed. The not non–decreasing operations are highlighted in purple. We excluded those operations because their Boolean counterparts are non–decreasing. There are other operations which are non–decreasing but not yet marked in this table; they are subject to further tests. The operations which fulfil the sufficient condition are written in red.

$x * y$	$x * y$	$x * y$	$x * y$	$x * y$	$x * y$
1	17	33	49	65	81
2	18	34	50	66	82
3	19	35	51	67	83
4	20	36	52	68	84
5	21	37	53	69	85
6	22	38	54	70	86
7	23	39	55	71	87
8	24	40	56	72	88
9	25	41	57	73	89
10	26	42	58	74	90
11	27	43	59	75	91
12	28	44	60	76	92
13	29	45	61	77	93
14	30	46	62	78	94
15	31	47	63	79	95
16	32	48	64	80	96

Table 7.3: Beran's numbers of binary operations, for which the Boolean counterpart is not non–decreasing in the first argument.

7.2.3 The MO_2 part of operations

In MO_2, two different non–central elements are compatible, if and only if one of them is the orthocomplement of the other. The lattice MO_2 can be considered as the free orthomodular lattice with two (non–free) generators a, b which satisfy

$$a \wedge b = a \wedge b' = a' \wedge b = a' \wedge b' = 0$$

and

$$a \vee b = a \vee b' = a' \vee b = a' \vee b' = 1$$

by duality. To choose $a, b, c \in MO_2$ so that $a \leqslant b$ and c does not commute with at least one of a and b, there are essentially only two possibilities: (see Figure 5.2)

$$a = 0\,, \quad b = x\,, \quad c = y\,, \tag{7.6}$$

or

$$a = x\,, \quad b = 1\,, \quad c = y\,. \tag{7.7}$$

All other possibilities are equivalent up to automorphisms of MO_2. It is thus necessary to test the inequalities

$$0 * y \leqslant x * y\,, \tag{7.8}$$

and

$$x * y \leqslant 1 * y \tag{7.9}$$

for the generators a, b of MO_2. Moreover, the two inequalities are dual, thus testing one of them for all orthomodular lattice operations gives answer also to the other. Even this simple tool excludes many orthomodular lattice operations as being non–monotonic.

$0 * y$	$x * y$	$x * y$	$x * y$	$x * y$	$x * y$	$x * y$
1	1	17	33	49	65	81
1	2	18	34	50	66	82
1	3	19	35	51	67	83
39	4	20	36	52	68	84
58	5	21	37	53	69	85
1	6	22	38	54	70	86
39	7	23	39	55	71	87
58	8	24	40	56	72	88
39	9	25	41	57	73	89
58	10	26	42	58	74	90
96	11	27	43	59	75	91
39	12	28	44	60	76	92
58	13	29	45	61	77	93
96	14	30	46	62	78	94
96	15	31	47	63	79	95
96	16	32	48	64	80	96

Table 7.4: Beran's numbers of the operations which do not fulfil $0 * y \leqslant x * y$.

$x * y$	$x * y$	$x * y$	$x * y$	$x * y$	$x * y$	$1 * y$
1	17	33	49	65	81	1
2	18	34	50	66	82	39
3	19	35	51	67	83	58
4	20	36	52	68	84	1
5	21	37	53	69	85	1
6	22	38	54	70	86	96
7	23	39	55	71	87	39
8	24	40	56	72	88	39
9	25	41	57	73	89	58
10	26	42	58	74	90	58
11	27	43	59	75	91	1
12	28	44	60	76	92	96
13	29	45	61	77	93	96
14	30	46	62	78	94	39
15	31	47	63	79	95	58
16	32	48	64	80	96	96

Table 7.5: Beran's numbers of the operations which do not fulfil $x * y \leqslant 1 * y$.

After combining both, $0 * y \leqslant x * y \leqslant 1 * y$, almost all not non–decreasing operations are sorted out of the list of all binary orthomodular lattice operations.

7.2.4 The free orthomodular lattice with three (non-free) generators

In this section, we make again use of the technique described in Section 5.2.1, on calculations in the free orthomodular lattice $\mathrm{F}(a, b, c)$ with three generators a, b, c, where c commutes with both a and b. Weaker compatibility conditions do not lead to a finite free orthomodular lattice. Only in this case an algorithm for checking the inequalities can be used.

By substituting a by $(x \wedge z)$, b by z, and c by y in (7.1), we obtain:

$$(x \wedge z) * y \leqslant z * y\,. \tag{7.10}$$

This does not reduce the number of inequalities to check, but in this expression the arguments a, b, and c are replaced by the generators of $\mathrm{F}(x, y, z)$ and the computer program of M. Hyčko can be easily used to verify (7.10). If this inequality holds for some orthomodular lattice operation $*$, its monotonicity is undecided. However, if (7.10) is violated in this special case, then $*$ is not non–decreasing in the first argument.

7.2.5 The free orthomodular lattice generated by two free generators

The free orthomodular lattice with three non–free generators discussed in the latter section is a product of two factors which are isomorphic to the product of its intervals $[0, \kappa]$, and $[0, \kappa']$. In the former, κ acts as 1, in the latter, κ acts as 0. Thus we may reduce the number of calculations by considering the two simpler cases separately: First, we choose $\kappa = 1$, then inequality (7.10) reduces to (7.9). Dually, we choose $\kappa = 0$, then inequality (7.10) reduces to (7.8). The violation of (7.9) or (7.8) is sufficient to prove that (7.1) is violated, too. All necessary computations can be performed in the free orthomodular lattice with two free generators x, y. The use of a computer program makes it a routine check.

We checked all 96 binary operations on the inequality (7.9), see Table 7.5, and found 63 operations which do not fulfil (7.9) and which violate (7.1) as well, in Table 7.5 they are highlighted in purple. Among them those 42 operations already found in the section about the Boolean part of operations (Section 7.2.2).

By checking (7.8) on the remaining 33 operations, it was possible to exclude 15 other operations which violate (7.1), see Table 7.4, where those operations not fulfilling $0 * y \leqslant x * y$ are highlighted in purple.

7.2.6 Kalmbach embedding

A further tool which we used to reduce the amount of work is the Kalmbach embedding introduced in [36] and advanced in [23, 42], see an overview in [55].

Kalmbach embedding starts from any given bounded lattice (with or without an orthocomplementation) and embeds it into a bigger orthomodular lattice. The original lattice operations, $\vee$ join and $\wedge$ meet, are preserved (the original orthocomplementation, if any, is not preserved).

Example

We start from the bounded lattice $P = \{0, x, y, z, 1\}$ called the *pentagon*, see Figure 7.1. It is composed of two maximal chains, $(0, z, 1)$ and $(0, x, y, 1)$, which intersect only at the bounds, 0 and 1.

According to the Kalmbach embedding, each chain generates a Boolean algebra of the corresponding size. The intersection of these Boolean algebras is determined by the intersection of the generating chains, here it is the two–element Boolean algebra $\{0, 1\}$. The chain $(0, z, 1)$ generates the Boolean algebra $(0, z, z', 1) \cong \mathbf{2^2}$, the chain $(0, x, y, 1)$ generates the Boolean algebra $\{0, x, y, x', y', t, t', 1\} \cong \mathbf{2^3}$ with three atoms x, y', and $t = y \wedge x'$.

The technique of *pasting* equips the union of the two Boolean algebras with orthomodular lattice operations (orthocomplementation and lattice operations) inherited from the two Boolean algebras and completed in a unique way. Pasting's particular instance is a part of the Kalmbach embedding, see [37, 48]. We obtain the orthomodular lattice $L = \{0, x, y, x', y', t, t', z, z', 1\}$ whose Greechie and Hasse diagram is depicted in Figure 7.2. The elements of L corresponding to those of P are circumcircled in the Hasse diagram. Alternatively, it can be described as the *horizontal sum* of the two Boolean algebras, $\mathbf{2^2}$ and $\mathbf{2^3}$, see [37, 48].

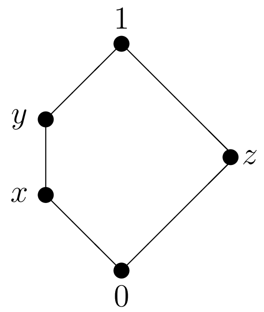


Figure 7.1: The pentagon (P).

The pentagon in Figure 7.1 represents the smallest bounded lattice where $x < y$ and the argument z is incomparable to x, y. Figure 7.2 represents the smallest *orthomodular* lattice with these properties. Since it is finite and small, it allows to test (7.1) easily and obtain necessary conditions for monotonicity of $*$. This particular case appeared crucial for the exclusion of some orthomodular lattice operations.

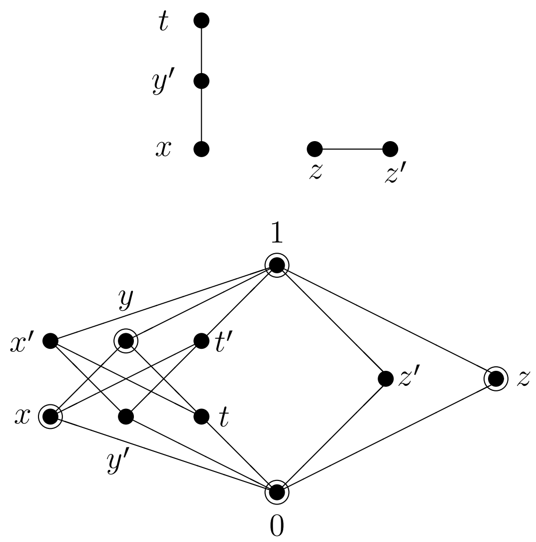


Figure 7.2: Greechie and Hasse diagram of $\mathbf{2^2} \oplus \mathbf{2^3}$.

The orthomodular lattice $L := \mathbf{2^2} \oplus \mathbf{2^3}$ of Figure 7.2 can be also used to test inequality (7.8) (if we take 0 and $x \in L$ instead of x and y in (7.1)) and inequality (7.11) (if we take y and $1 \in L$ instead of x and y in (7.1)). Thus three tests of necessary conditions can be performed in a unique form: in L, we map the chain $(0, x, y, 1)$ to $(0 * z, x * z, y * z, 1 * z)$ and ask if this quadruplet (in this order) is a non–decreasing sequence. This reformulation appeared to be very efficient in finding necessary conditions for monotonicity of orthomodular lattice operations.

Operation	$0 * z$		$x * z$	$y * z$	$1 * z$
1	0		0	0	0
2	0		0	0	z
3	0		0	0	z'
4	z	$>$	0	0	0
5	z'	$>$	0	0	0
6	0		0	0	1
7	z	$>$	0	0	z
8	z'	$>$	0	0	z
9	z	$>$	0	0	z'
10	z'	$>$	0	0	z'
11	1	$>$	0	0	0
12	z	$>$	0	0	1
13	z'	$>$	0	0	1
14	1	$>$	0	0	z
15	1	$>$	0	0	z'
16	1	$>$	0	0	1

Operation	$0 * z$		$x * z$	$y * z$		$1 * z$
17	0		x	y	$>$	0
18	0		x	y	$\nleq$	z
19	0		x	y	$\nleq$	z'
20	z	$\nleq$	x	y	$>$	0
21	z'	$\nleq$	x	y	$>$	0
22	0		x	y		1
23	z	$\nleq$	x	y	$\nleq$	z
24	z'	$\nleq$	x	y	$\nleq$	z
25	z	$\nleq$	x	y	$\nleq$	z'
26	z'	$\nleq$	x	y	$\nleq$	z'
27	1	$>$	x	y	$>$	0
28	z	$\nleq$	x	y		1
29	z'	$\nleq$	x	y		1
30	1	$>$	x	y	$\nleq$	z
31	1	$>$	x	y	$\nleq$	z'
32	1	$>$	x	y		1
33	0		z	z	$>$	0
34	0		z	z		z
35	0		z	z	$\nleq$	z'
36	z		z	z	$>$	0
37	z'	$\nleq$	z	z	$>$	0
38	0		z	z		1
39	z		z	z		z
40	z'	$\nleq$	z	z		z
41	z		z	z	$\nleq$	z'
42	z'	$\nleq$	z	z	$\nleq$	z'
43	1	$>$	z	z	$>$	0
44	z		z	z		1
45	z'	$\nleq$	z	z		1
46	1	$>$	z	z		z
47	1	$>$	z	z	$\nleq$	z'
48	1	$>$	z	z		1
49	0		z'	z'	$>$	0
50	0		z'	z'	$\nleq$	z
51	0		z'	z'		z'
52	z	$\nleq$	z'	z'	$>$	0
53	z'		z'	z'	$>$	0
54	0		z'	z'		1
55	z	$\nleq$	z'	z'	$\nleq$	z
56	z'		z'	z'	$\nleq$	z
57	z	$\nleq$	z'	z'		z'
58	z'		z'	z'		z'

Operation	$0 * z$		$x * z$	$y * z$		$1 * z$
59	1	$>$	z'	z'	$>$	0
60	z	$\nleqslant$	z'	z'		1
61	z'		z'	z'		1
62	1	$>$	z'	z'	$\nleqslant$	z
63	1	$>$	z'	z'		z'
64	1	$>$	z'	z'		1
65	0		x'	y'	$>$	0
66	0		x'	y'	$\nleqslant$	z
67	0		x'	y'	$\nleqslant$	z'
68	z	$\nleqslant$	x'	y'	$>$	0
69	z'	$\nleqslant$	x'	y'	$>$	0
70	0		x'	y'		1
71	z	$\nleqslant$	x'	y'	$\nleqslant$	z
72	z'	$\nleqslant$	x'	y'	$\nleqslant$	z
73	z	$\nleqslant$	x'	y'	$\nleqslant$	z'
74	z'	$\nleqslant$	x'	y'	$\nleqslant$	z'
75	1	$>$	x'	y'	$>$	0
76	z	$\nleqslant$	x'	y'		1
77	z'	$\nleqslant$	x'	y'		1
78	1	$>$	x'	y'	$\nleqslant$	z
79	1	$>$	x'	y'	$\nleqslant$	z'
80	1	$>$	x'	y'	$\nleqslant$	z'
81	0		1	1	$>$	0
82	0		1	1	$>$	z
83	0		1	1	$>$	z'
84	z		1	1	$>$	0
85	z'		1	1	$>$	0
86	0		1	1		1
87	z		1	1	$>$	z
88	z'		1	1	$>$	z
89	z		1	1	$>$	z'
90	z'		1	1	$>$	z'
91	1		1	1	$>$	0
92	z		1	1		1
93	z'		1	1		1
94	1		1	1	$>$	z
95	1		1	1	$>$	z'
96	1		1	1		1

Table 7.6: Binary operations, not fulfilling $0 * z \leqslant x * z \leqslant y * z \leqslant 1 * z$.

In Table 7.6 all binary orthomodular lattice operations *not* fulfilling one of the inequations $0 * z \leqslant x * z \leqslant y * z \leqslant 1 * z$ are highlighted, we wrote "$>$" if the inequation was reversed or “$\nleqslant$” if the results where incomparable. We did not write explicitly “$\leqslant$” for fulfilled inequations. In the first column, all the operations fulfilling the inequation chain are left blank.

We see that the example based on the Kalmbach embedding gives us the necessary condition for monotonicity of the first argument.

7.3 Summary of results

Using the necessary conditions, we proved that 79 binary orthomodular lattice operations are not non–decreasing in the first argument. The remaining 17 operations fulfilled the sufficient condition as well as the necessary conditions. They are listed in Proposition 7.1.1. These operations are exactly those which are non–decreasing in the first argument.

The Kalmbach embedding displays the sufficient condition for monotonicity.

7.4 Monotonicity in the second variable

The same method is appropriate to find all binary orthomodular lattice operations which are monotone in the second variable. Concerning the sufficient conditions, the same argument as before, all operations not containing a negation in the second variable are non–decreasing. These are the seventeen operations with Beran’s numbers: 1, 2, 4, 7, 18, 22, 23, 28, 39, 68, 71, 75, 78, 87, 92, 94, and 96. Trivially, they are the same as for the monotonicity in the first variable by interchanging the first and the second argument, e.g. in the graphical notation x ⁘ $y = y$ ⁘ x. Six of them are non–decreasing in both variables; they are listed in Table 7.7. They correspond to the six associative operations, which we found in Section 6.

Beran’s no.	expression	graphical notation
1	0	x ⁘ y,
2	$x \wedge y$	x ⁘ y,
22	x	x ⁘ y,
39	y	x ⁘ y,
92	$x \vee y$	x ⁘ y,
96	1	x ⁘ y.

Table 7.7: Operations non–decreasing in both variables.

In Figure (7.3) we coloured the non–decreasing operations in the illustration of $\mathrm{F}(x, y)$, the operations non–decreasing in the first argument only are coloured in blue, those non–decreasing only in the second variable in green. Those operations non–decreasing in both variables are coloured in red.

It is remarkable that there is a certain pattern in the grouping of the operations,

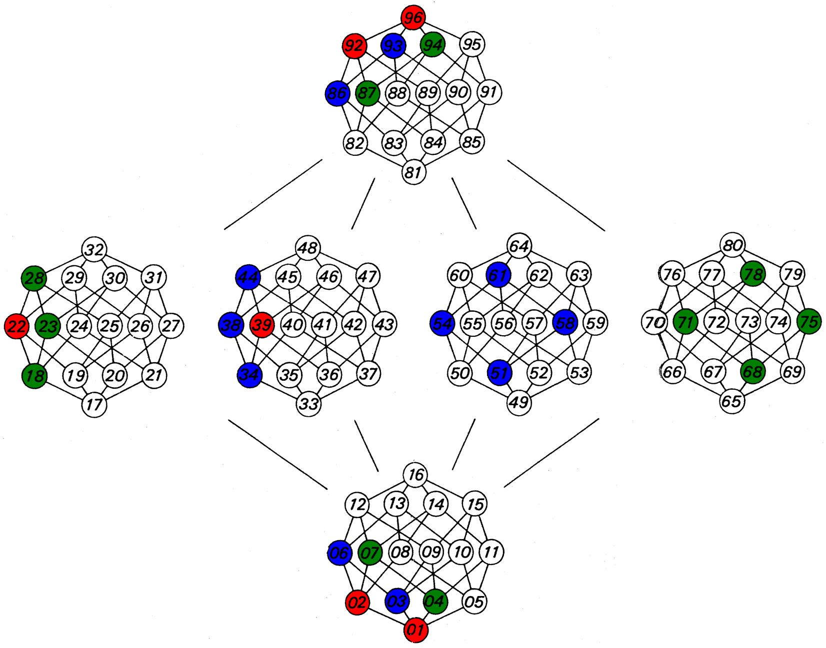


Figure 7.3: Results for non–decreasing binary operations.

except x [symbol] $y = x$ for monotonicity in the first argument and x [symbol] $y = y$ for monotonicity in the second argument. They appear in patterns of four, see Figure 7.3, this figure shows also a symmetry along the horizontal axis.

7.5 Further results

We also reversed the implication (7.1) and traced the following question:

For which of the 96 binary operations $*$ the following implication holds:

$$a \leqslant b \quad \Rightarrow \quad a * c \geqslant b * c \quad \forall\, c \tag{7.11}$$

This question can be treated with exactly the same method as above. Then, implication (7.11) holds for the operations with Beran's numbers 1, 4, 5, 11, 36, 39, 43, 46, 53, 58, 59, 63, 75, 91, 94, 95, and 96 in the first argument, and as many in the second argument, 1, 3, 5, 10, 19, 22, 26, 29, 58, 69 74, 75, 79, 90, 93, 95, and 96. Six of them are non–increasing in both arguments; they are listed in Table 7.8. They are exactly the complements of the operations found in Table 7.7.

In Figure 7.4 we highlighted the Beran's numbers of all monotone orthomodular lattice operations fulfilling implication (7.1) or (7.11) in the first or second variable, there are 46 in total and if $x * y$ is one of them, so also $x * y'$, $x' * y$, $x' * y'$, $y * x$, $y * x'$, $y' * x$, and $y' * x'$, this explains the symmetry along the vertical and horizontal axes.

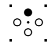
Beran's no.	expression	graphical notation
1	0	x  y
5	$x' \wedge y'$	x  y
58	y'	x  y
75	x'	x  y
95	$x' \vee y'$	x  y
96	1	x  y.

Table 7.8: Operations non–increasing in both arguments.

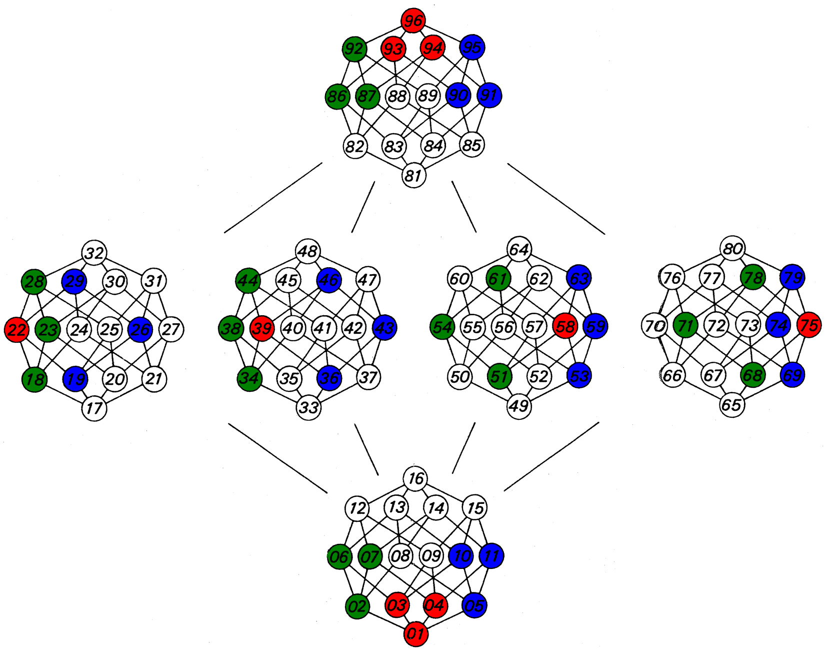


Figure 7.4: Results for monotonicity of binary operations.

7.6 Conclusions

Our goal was to find a possible way of reducing the complexity of some orthomodular lattice operations and write them in a unique normal form.

We analysed the monotonicity of the 96 binary orthomodular lattice operations, we found 17 monotonic operations for each argument, among them the so–called *skew join* (B44):

$$x \mathbin{\dot\vee} y = (x \wedge y') \vee y = x$$  y

and the *skew meet* (B34):

$$x \mathbin{\dot\wedge} y = (x \vee y') \wedge y = x$$  y .

The distributivity of skew operations, if one argument commutes with the two other arguments, was proved in [3]. However, it also does not enable normal forms in orthomodular lattices. The problem is that the skew operations are not associative. Therefore they do not admit to combine many terms to a simple expression with the freedom of the choice of the order of the arguments.

The standard way of finding a normal form, known from Boolean algebras, is not applicable in orthomodular lattices, even if we reduce our requirements.

Chapter 8

Non-associative operations on orthomodular lattices

In the literature three levels of associativity are distinguished. An algebra is called *power–associative* if the subalgebra generated by any single element is associative. It is called an *alternative algebra*, if the subalgebra generated by any two of its elements is associative. If the subalgebra generated by any three elements is associative, then the algebra is called an *associative algebra* [1].

In power–associative algebras the order in which an element is multiplied by itself, e.g. $(xx)x = x(xx)$ does not matter. Power–associative algebras are weaker than alternative or associative algebras, i.e. associative and alternative algebras are power–associative algebras. We will not continue on power–associative algebras, nor on associative algebras (the latter are well–known) but we pay attention to alternative algebras.

8.1 Alternative Algebras

An alternative algebra M is a non–empty set M together with an operation " $\cdot$ ", called the multiplication. The multiplication need not to be associative, but it has to be *alternative*, this means that the following equations hold

$$x \cdot (x \cdot y) = (x \cdot x) \cdot y \qquad \text{left identity and} \tag{L}$$
$$(y \cdot x) \cdot x = y \cdot (x \cdot x) \qquad \text{right identity} \tag{R}$$

for all $x, y \in M$.

From the left (L) and right (R) identities the flexible identity follows

$$x \cdot (y \cdot x) = (x \cdot y) \cdot x \tag{F}$$

Often it is written "xy" instead of "$x \cdot y$".

The Moufang identities

A *loop* is a quasigroup with an identity element. In loops, the identities (L), (R) and (F) are equivalent. The best known loop is the *Moufang loop*, this is a loop in which all the elements x, y and z satisfy the *Moufang identities*. They are:

$$\big((xy)x\big)z = x\big(y(xz)\big) \qquad \text{left Moufang identity,} \qquad \text{(M1)}$$
$$\big((yx)z\big)x = y\big(x(zx)\big) \qquad \text{right Moufang identity,} \qquad \text{(M2)}$$
$$(xy)(zx) = \big(x(yz)\big)x \qquad \text{central Moufang identity, and} \qquad \text{(M3)}$$
$$(xy)(zx) = x\big((yz)x\big)\,. \qquad \text{(M4)}$$

Two examples of alternative algebras are the algebra of the octonions and the Moufang loops.

8.2 Associativity and parentheses

The distribution of the parentheses is closely attached to the concept of associativity. D. Tamari made research on the way of putting parentheses in algebraic expressions. In this section we give a short overview of his work on parentheses and this section is mainly based on his paper [56].

The traditional definition of the associativity law is

$$A_2 : x(yz) = (xy)z\,.$$

In general, associativity can be written as

$$A_n(P,Q) : P(x_0,\dots,x_n) = Q(x_0,\dots,x_n)$$

with $2 \leqslant n \in \mathbb{N}$. The expressions P and Q are two different ways of (correct) parentheses setting of the expression $x_0 \cdots x_n$ of length $n+1$. They are equal if the associativity law holds.

The associativity problem can thus be expressed as:

$$A_n(P,Q) : P(x_0,\dots,x_n) = s, \quad Q(x_0,\dots,x_n) = t \quad \Rightarrow \quad s = t \quad (n \geqslant 2)\,,$$

The case where $n = 1$ can also be included by

$$A_1 : xy = s, \quad xy = t \quad \Rightarrow \quad s = t$$

A word of length $n+1$ has n pairs of parentheses, we usually write only $n-1$, e.g. one possibility for putting parentheses in the word $uvwx$, $n = 3$, could be $(((uv)w)x)$ – three pairs, usually we write $((uv)w)x$ – two pairs.

For a word of length $n+1$ we have

$$C(n) = \frac{1}{n}\binom{2n}{n-1}$$

possibilities of parentheses. The $C(n)$ are called the *Catalan numbers*, they grow asymptotically with n as

$$C(n) \sim \frac{4^n}{\sqrt{2^3\pi}}.$$

To prove associativity, we have

$$\binom{C(n)}{2} = \frac{1}{2}C(n)\big(C(n)-1\big)$$

associative laws A_n to check. In our example there are $C(3) = 5$ possibilities to put parentheses, thus 10 associative laws to check, namely:

$$\begin{aligned}
A_3:&\quad \big((uv)w\big)x = a, && \big(u(vw)\big)x = b && \Rightarrow && a = b,\\
A_3:&\quad \big((uv)w\big)x = a, && (uv)(wx) = c && \Rightarrow && a = c,\\
A_3:&\quad \big((uv)w\big)x = a, && u\big((vw)x\big) = d && \Rightarrow && a = d,\\
A_3:&\quad \big((uv)w\big)x = a, && u\big(v(wx)\big) = e && \Rightarrow && a = e,\\
A_3:&\quad \big(u(vw)\big)x = b, && (uv)(wx) = c && \Rightarrow && b = c,\\
A_3:&\quad \big(u(vw)\big)x = b, && u\big((vw)x\big) = d && \Rightarrow && b = d,\\
A_3:&\quad \big(u(vw)\big)x = b, && u\big(v(wx)\big) = e && \Rightarrow && b = e,\\
A_3:&\quad (uv)(wx) = c, && u\big((vw)x\big) = d && \Rightarrow && c = d,\\
A_3:&\quad (uv)(wx) = c, && u\big(v(wx)\big) = e && \Rightarrow && c = e,\\
A_3:&\quad u\big(v(wx)\big) = e, && \big((uv)w\big)x = a && \Rightarrow && e = a.
\end{aligned}$$

D. Tamari found a way to define a partial order on the set of all possible ways of putting parentheses on words of length $n+1$. He grouped the objects in pairs (for binary operations). Two groupings are comparable if the second can be obtained from the first by successive rightward application of the associative law, $(ab)c \to a(bc)$. In our example: $u\big((vw)x\big)$ can be obtained from $\big(u(vw)\big)x$ by rightward sliding the parentheses, thus $\big(u(vw)\big)x \leqslant u\big((vw)x\big)$. In this manner we get for words of length four the non–modular lattice N_5 (pentagon), see Figure 8.1.

D. Tamari's [56] results were rather negative. He concluded that the associativity problem is not resolvable. The task becomes infinite due to the large number of equations to check, even in finite monoids. As an illustration, see Figure 8.2, of a five variable expression, with already 14 ways to put parentheses and 91 associative laws to check!

8.3 Weaker forms of associativity in orthomodular lattices

In Section 6 we proved that only six of the 96 binary orthomodular lattice operations are associative. In this section we examine some weaker forms of associativity in orthomodular lattices. We study all binary operations on orthomodular lattices which form an alternative algebra; their specification is the following:

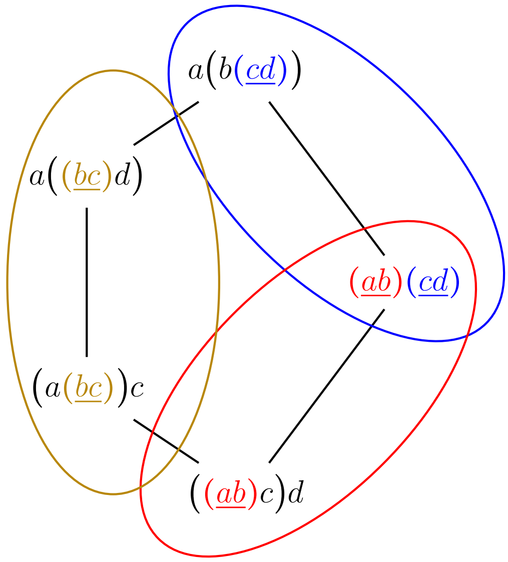


Figure 8.1: A scheme of the five ways of putting parentheses in a four-variable expression.

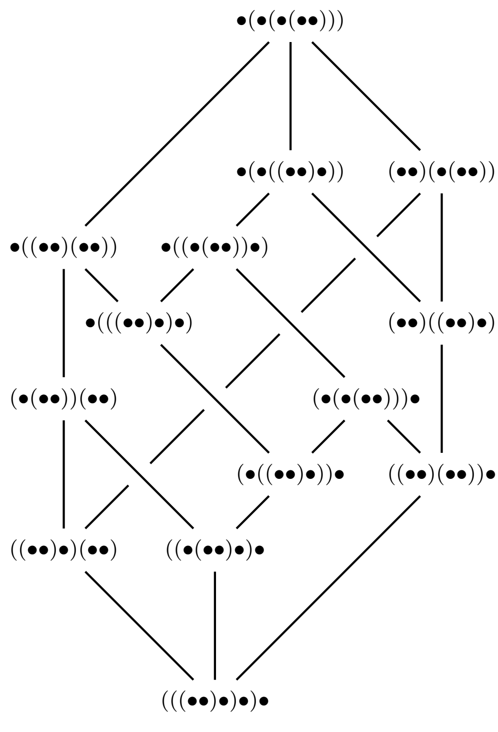


Figure 8.2: The Tamari lattice made by 14 ways of putting parentheses on a five-variable expression.

Theorem 8.3.1
Let $*$ be one of the 96 binary operations on orthomodular lattices. The operation $*$ satisfies all three of the identities (L), (R), (F) in orthomodular lattices if and only if its Beran's number is in the set {1, 2, 16, 18, 22, 23, 28, 34, 38, 39, 44, 81, 92, 96}. All other operations satisfy at most one of the identities (L), (R), (F).

Proof:
As the identities deal with only two variables, the program [32] is suited to find those expressions fulfilling (L), (R), (F). The results were already shown in Table 6.3. The columns (6.2), (6.4) and (6.6) correspond to the equations (L), (R) and (F) respectively. "`True`" means that the equation holds, the results can be checked in the appropriate columns.

q.e.d.

$*$	$x * y$	$x * x * y$, $x * y * y$	$x * y * x$
16	$(x \wedge y) \vee (x \wedge y') \vee (x' \wedge y) \vee (x' \wedge y')$	$\mathbf{1}$	$\mathbf{1}$
18	$x \wedge (x' \vee y)$	$x \wedge (x' \vee y)$	$x \wedge (x' \vee y)$
23	$(x' \vee y) \wedge \big(x \vee (x' \wedge y)\big)$	$(x' \vee y) \wedge \big(x \vee (x' \wedge y)\big)$	x
28	$x \vee (x' \wedge y)$	$x \vee (x' \wedge y)$	$x \vee (x' \wedge y)$
34	$y \wedge (y' \vee x)$	$y \wedge (y' \vee x)$	$x \wedge (x' \vee y)$
38	$(x \vee y') \wedge \big(y \vee (y' \wedge x)\big)$	$(x \vee y') \wedge \big(y \vee (y' \wedge x)\big)$	x
44	$y \vee (y' \wedge x)$	$y \vee (y' \wedge x)$	$x \vee (x' \wedge y)$
81	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y) \wedge (x' \vee y')$	$\mathbf{0}$	$\mathbf{0}$

Table 8.1: Nonassociative operations satisfying (L), (R) and (F) with their results.

In Table 8.1 the operations fulfilling all three (L), (R) and (F) are listed. Observing the expressions we remark that there are three couples which differ only by the order of their arguments (see also Table 6.4), namely:

18 and 34; 28 and 44; 23 and 38.

Further, there are three couples of dual operations,

18 and 28; 34 and 44; 16 and 81.

The two operations, 23 and 38, are self–dual.

The question treated in this section is which of the eight operations of Table 8.1 satisfy weaker laws of associativity similar to the Moufang identities (M1 to M4) In particular, we study the following six *constellations*:

$$x * x * y * z \tag{xxyz}$$
$$x * y * x * z \tag{xyxz}$$
$$x * y * z * x \tag{xyzx}$$
$$y * x * x * z \tag{yxxz}$$
$$z * x * y * x \tag{zxyx}$$
$$z * y * x * x \tag{zyxx}$$

Remark 8.3.2
There are nine different formations of three variables with one double, we only need to examine six of them.

If we take three different elements x, y and z of an algebra, then there are $3! = 6$ possible constellations:

$$\begin{array}{ll} x, y, z\,; & x, z, y\,; \\ y, z, x\,; & z, y, x\,; \\ z, x, y\,; & y, x, z\,. \end{array}$$

Let the variables y and z have the same role, i.e. they are arbitrary elements and are interchangeable, their order is irrelevant. Thus the above constellations reduce to three:

$$x, y, z\,, \qquad y, z, x\,, \qquad z, x, y\,.$$

The variable x is the one we want to appear double i.e. x plays a different role as y and z. Its place is important, there are nine different positions to put the variable x twice in the last three constellations:

$$\begin{array}{lll} x, x, y, z\,; & y, x, x, z\,; & z, x, x, y\,; \\ x, y, x, z\,; & y, x, z, x\,; & z, x, y, x\,; \\ x, y, z, x\,; & y, z, x, x\,; & z, y, x, x\,. \end{array}$$

Again, due to the arbitrarily chosen variables y and z, we can reduce these to six different orders of arguments, namely

$$\begin{array}{ll} x, x, y, z\,; & y, x, x, z\,; \\ x, y, x, z\,; & y, x, z, x\,; \\ x, y, z, x\,; & y, z, x, x\,. \end{array}$$

In Section 8.2, we figured out already that each constellation of length four admits five different ways of putting parentheses. To prove associativity we have to check which of the above constellations give the same result for all possible ways of putting parentheses, e.g.

$$\big((x * x) * y\big) * z = \tag{P1}$$
$$\big(x * (x * y)\big) * z = \tag{P2}$$
$$(x * x) * (y * z) = \tag{P3}$$
$$x * \big((x * y) * z\big) = \tag{P4}$$
$$x * \big(x * (y * z)\big). \tag{P5}$$

We shall use the symbols (Pn) to refer to the ways of putting parentheses in other constellations, e.g., constellation (xyxz) with parentheses (P1) reads

$$((x * y) * x) * z\,, \text{ etc.}$$

The equality of the expressions obtained by all five ways of putting parentheses is encountered rather rarely. Many positive results are obtained if an extra assumption that two of the variables commute is added. If we assume that two pairs of variables commute (and the third does not), then we may apply the program [33] to answer the question. If all three variables mutually commute, all computations can be easily made in a Boolean algebra.

Remark 8.3.3
We have eight operations listed in Table 8.1 along with six basic constellations. Taking into account the duality and reversion of arguments, results for some operations (although they are not identical) can be derived from others, thus it suffices to investigate three operations, see Table 8.2.

no.	expression	name	analogue
18	$a \wedge (a' \vee b)$	Sasaki projection	$28, 34, 44$
23	$\big(a \vee (a' \wedge b)\big) \wedge (a' \vee b)$	swapped projection	38
16	$(a \wedge b) \vee (a \wedge b') \vee (a' \wedge b) \vee (a' \wedge b')$	lower commutator	81

Table 8.2: The three typical operations and the Beran's numbers of their analogues.

In our paper [18] we proposed the name *swapped projection* for the operation with Beran's number 23, it acts as the *right* projection on commuting elements and as the *left* projection on generators of MO_2. For each of these choices, there are five ways of putting parentheses, (P1) to (P5). The associativity problem: "are they all equal?". We have 6 constellations, 3 Beran's expressions, i.e. B16, B18 and B23, thus 18 quintuples of expressions which have to be compared in the five ways of putting parentheses.

8.4 Tools

Different techniques will facilitate our tasks. We will use the already mentioned and used Foulis–Holland Theorem 3.2.4. In cases that at least two compatibility assumptions hold, it can be solved by purely technical tools of [33].

Further we will rely on theorems about commuting elements, Lemmas 3.2.1 and 3.2.2 and Proposition 3.2.3.

8.5 Examples

To find counterexamples we use the lattices MO_2, Figure 5.2, the free generated lattice on two free generators $\mathrm{F}(a, b)$, Figure 5.1 and Figure 5.4. We use also the Dilworth lattice D_{16}, Figure 6.1, and the lattice L_{22}, Figure 6.3.

8.6 Identities generalizing associativity

8.6.1 Results obtained without using commutation

We start with results which do not assume commutation. They are all based on properties of the Sasaki projection, summarized in Lemma 8.6.1. In order to prove the main result, namely Theorem 8.6.2, we will need the results of Theorem 6.3.1.

Lemma 8.6.1
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 18 (Sasaki projection). It has the following properties:

(S1) If $y \leqslant z$ then $x * y \leqslant x * z$,
(S2) If $x \leqslant y$ then $x' \vee (y * z) = x' \vee z$,
(S3) If $x \leqslant z$ then $x' \vee (y * z) = x' \vee y$,
(S4) If $y \leqslant z$ then $(x * y) * z = x * (y * z) = x * y$,
(S5) If $x \leqslant z$ then $(x * y) * z = x * (y * z) = x * y$,
(S6) If $z \leqslant y$ then $x * (y * z) = (x * y) * z = x * z$.

Proof:

(S1) The Sasaki projection is expressed by using monotonic lattice operations $\vee$ and $\wedge$ without the orthocomplement being applied to any formulas containing y, as the second argument, see Section 7.

(S2) In the special case $x = y$, computation in two variables can be done in the free orthomodular lattice with 2 free generators, program [33] suffices to prove that

$$y' \vee (y * z) = y' \vee z .$$

This allows to prove the general result if $x' \vee y' = x'$:

$$\begin{aligned} x' \vee (y * z) &= x' \vee \underbrace{y' \vee (y * z)}_{= y' \vee z} \\ &= \underbrace{x' \vee y'}_{= x'} \vee z \\ &= x' \vee z . \end{aligned}$$

(S3) In the special case $x = z$, computation in two variables using the program [33] verifies

$$z' \vee (y * z) = z' \vee y .$$

This allows to prove the general result if $x' \vee z' = x'$:

$$\begin{aligned} x' \vee (y * z) &= x' \vee \underbrace{z' \vee (y * z)}_{=z' \vee y} \\ &= \underbrace{x' \vee z'}_{=x'} \vee y \\ &= x' \vee y\,. \end{aligned}$$

(S4) Let $y \leqslant z$, then

$$\begin{aligned} (x * y) * z &= x \wedge (x' \vee y) \wedge \big((x \wedge (x' \vee y))' \vee z\big) \\ &= x \wedge (x' \vee y) \wedge \big(x' \vee (x \wedge y') \vee z\big) \\ &\overset{(*)}{=} x \wedge (x' \vee y) = x * y\,, \end{aligned}$$

where the step $(*)$ follows from the inequalities

$$x' \vee y \leqslant x' \vee z \leqslant x' \vee (x \wedge y') \vee z\,.$$

The second part follows easily from the fact that $y \leqslant z$ implies $y * z = y \wedge z = y$,

$$x * (y * z) = x * y\,.$$

(S5) Because $x * y \leqslant x \leqslant z$, we have $(x * y)' \vee z = \mathbf{1}$ and

$$(x * y) * z = (x * y) \wedge \underbrace{\big((x * y)' \vee z\big)}_{=\mathbf{1}} = x * y\,.$$

Similarly, applying (S3),

$$x * (y * z) = x \wedge \underbrace{\big(x' \vee (y * z)\big)}_{=x' \vee y} = x \wedge (x' \vee y) = x * y\,.$$

(S6) Calculation gives:

$$\begin{aligned} (x * y) * z &= (x * y) \wedge \big((x * y)' \vee z\big) \\ &= (x * y) \wedge (x' \vee (x \wedge y') \vee z) \\ &= (x * y) \wedge \big((x \wedge y') \vee (x' \vee z)\big) \\ &= x \wedge (x' \vee y) \wedge \big((x' \wedge y)' \vee (x' \vee z)\big) \end{aligned}$$

Applying the Foulis–Holland Theorem (Theorem 3.2.4) to it, we get an intermediate result

$$\begin{aligned} (x * y) * z &= \big((x * y) \wedge (x' \vee y)'\big) \vee \big((x * y) \wedge (x' \vee z)\big) \\ &= (x * y) \wedge (x' \vee z) \\ &= x \wedge \underbrace{(x' \vee y) \wedge (x' \vee z)}_{x' \vee z} && \text{(monotonicity of } \vee) \\ &= x * z \end{aligned}$$

Because $z \leqslant y$ it follows $z \, \mathrm{C} \, y$ and thus $y * z = z$,

$$(x * y) * z = x * z = x * (y * z) .$$

q.e.d.

Theorem 8.6.2
Let L be an orthomodular lattice and let $*$ be an operation with Beran's number in $\{18, 28\}$. Then

$$\begin{aligned}(x * y * x) * z &= (x * y) * (x * z) \\ \big(z * (x * y)\big) * x &= z * (x * y * x) \\ \big((x * y) * z\big) * x &= (x * y) * (z * x)\end{aligned}$$

for any $x, y, z \in L$.

Proof:
Suppose $x * y = x \wedge (x' \vee y)$ (Beran's number 18). By Theorem 8.3.1, for any $x, y \in L$ the word $x * y * x$ is equal to $x * y$ regardless of the order of operations.

As $(x * y) \, \mathrm{C} \, x$, we may apply Theorem 6.3.1 with the substitutions[1] $x := x * y$, $y := x$ and obtain

$$\big((x * y) * x\big) * z = (x * y) * (x * z) .$$

As $x * y \leqslant x$, we may use (S4) with the substitutions: $x := z$, $y := x * y$, $z := x$

$$\big(z * (x * y)\big) * x = z * \big((x * y) * x\big) = z * (x * y) .$$

Similarly, (S5) with the substitutions: $x := x * y$, $y := z$, $z := x$ gives:

$$\big((x * y) * z\big) * x = (x * y) * (z * x) = (x * y) * z .$$

The second case of B28 is then dual.

q.e.d.

Corollary 8.6.3
Let L be an orthomodular lattice and let $\circledast$ be an operation with Beran's number in $\{34, 44\}$. Then

$$\begin{aligned}z \circledast (x \circledast y \circledast x) &= (z \circledast x) \circledast (y \circledast x) \\ x \circledast \big((y \circledast x) \circledast z\big) &= (x \circledast y \circledast x) \circledast z \\ x \circledast \big(z \circledast (y \circledast x)\big) &= (x \circledast z) \circledast (y \circledast x)\end{aligned}$$

for any $x, y, z \in L$.

[1] All substitutions should be done at the same time, without iterative substitutions, i.e., (x, y) is replaced with $(x * y, x)$, which satisfies the assumption $(x * y) \, \mathrm{C} \, x$. The same meaning is assumed in all substitutions in the sequel.

Proof:
Let $*, \circledast$ be the operations with Beran's numbers $18, 34$, respectively. As $a * b = b \circledast a$, we obtain

$$z \circledast (x \circledast y \circledast x) = (x * y * x) * z = (x * y) * (x * z) = (z \circledast x) \circledast (y \circledast x)$$

and similarly for other equalities from Theorem 8.6.2.

q.e.d.

8.6.2 Results obtained by using commutation

Here we summarise results that assume at least one pair of variables commuting.

Theorem 8.6.4
Let L be an orthomodular lattice and let $*$ be an operation with the Beran's number in $\{18, 28\}$. If $x, y, z \in L$ are such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms appear:

$$\begin{aligned} &x * x * y * z\,, \qquad x * y * x * z\,,\\ &x * y * z * x\,, \qquad y * x * x * z\,,\\ &y * x * z * x\,, \qquad x * y * z * z\,. \end{aligned}$$

Thus all Moufang identities (M1) to (M4) hold.

Remark 8.6.5
The latter two constellations, $y * x * z * x$ and $x * y * z * z$, are the same as $z * x * y * x$ (zxyx) and $z * y * x * x$ (zyxx), respectively; we interchanged variables only to use the same assumption $x \mathrm{C}\, y$ in all cases.

Proof:
Let $*$ be the Sasaki projection (B18). Identity (L) attains the form

$$x * (x * y) = (x * x) * y = x * y = x \wedge y\,.$$

This allows to write

$$\big(x * (x * y)\big) * z = \big((x * x) * y\big) * z = (x * y) * z = (x \wedge y) * z$$

Likewise, by replacing y with $y * z$, it follows that

$$x * \big(x * (y * z)\big) = (x * x) * (y * z) = x * (y * z)\,.$$

These expressions are equal due to Theorem 6.3.1, which also allows to derive equality with the final remaining way of putting the parentheses:

$$x * \big((x * y) * z\big) = x * \big(x * (y * z)\big)\,.$$

Now, by applying Theorem 6.3.1 to the relations $x \,\mathrm{C}\, y$, $(x * y) \,\mathrm{C}\, x$ and $x \,\mathrm{C}\, (y * x)$, it follows that

$$x * \big(y * (x * z)\big) = (x * y) * (x * z) = \big((x * y) * x\big) * z = \big(x * (y * x)\big) * z$$
$$= x * \big((y * x) * z\big) = (x \wedge y) * z\,.$$

Similarly, Theorem 6.3.1 with $x \,\mathrm{C}\, y$ proves that

$$x * (y * z) * x = \big((x * y) * z\big) * x \quad \text{and} \quad x * \big(y * (z * x)\big) = (x * y) * (z * x)\,.$$

According to (S5) with substitutions $x := x * y$, $y := z$, $z := x$, the latter two expressions equal $(x * y) * z = (x \wedge y) * z$. Next, Theorem 6.3.1 with $x \,\mathrm{C}\, y$ proves that

$$(y * x * x) * z = y * (x * x * z) = (y * x) * (x * z) = (x \wedge y) * z\,.$$

Then, by using Theorem 6.3.1 with $x \,\mathrm{C}\, y$, it follows that

$$y * (x * z * x) = (y * x) * (z * x) \quad \text{and} \quad \big((y * x) * z\big) * x = \big(y * (x * z)\big) * x\,.$$

According to (S5) with substitutions $x := y * x$, $y := z$, $z := x$, the latter two expressions equal $(y * x) * z = (x \wedge y) * z$. Finally, Theorem 6.3.1 with $x \,\mathrm{C}\, y$ and (R) proves

$$x * (y * z * z) = (x * y) * (z * z) = \big((x * y) * z\big) * z = \big(x * (y * z)\big) * z = (x \wedge y) * z\,.$$

Likewise, the case of the operation with Beran's number 28 is dual.

q.e.d.

Similarly to Corollary 8.6.3, we may prove the following:

Corollary 8.6.6
Let L be an orthomodular lattice and let $\circledast$ be an operation with the Beran's expression in $\{34, 44\}$. If $x, y, z \in L$ such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms appear:

$$z \circledast y \circledast x \circledast x\,, \qquad z \circledast x \circledast y \circledast x\,,$$
$$x \circledast z \circledast y \circledast x\,, \qquad z \circledast x \circledast x \circledast y\,,$$
$$x \circledast z \circledast x \circledast y\,, \qquad z \circledast z \circledast y \circledast x\,.$$

If y and z are switched and $x \,\mathrm{C}\, z$, then all Moufang identities (M1) to (M4) hold.

Theorem 8.6.7
Let L be an orthomodular lattice and let $*$ be an operation with the Beran's number in $\{16, 81\}$. If $x, y, z \in L$ such that x commutes with y or z then each of the expressions (xxyz), (xyxz), (xyzx), (yxxz), (zxyx), (zyxx) has a unique output regardless of the order in which the terms appear.

Proof:
For the lower commutator ($* = \underline{\mathrm{com}}$, Beran's number 16), the following rules suffice to evaluate all cases showing that they are equal to **1**:

1. The commutators are commutative, $x * y = y * x$.
2. For two commuting elements, the lower commutator results in $\mathbf{1}$.
3. The element $\mathbf{1}$ is absorbing, $\mathbf{1} * x = \mathbf{1}$.
4. The commutator commutes with both its arguments, $x \,\mathrm{C}\, (x * y)$, $y \,\mathrm{C}\, (x * y)$.

As a consequence of the above, we obtain

$$x * x * y \quad = \quad y * x * x \quad = \quad x * y * x \quad = \quad \mathbf{1}\,.$$

With the additional assumption $x \,\mathrm{C}\, z$ it follows from Proposition 3.2.3 that

$$x \,\mathrm{C}\, \big((x * y) * z\big)$$

and therefore

$$x * \big((x * y) * z\big) = \mathbf{1}$$

The same can be applied for the roles of y, z interchanged.

The case of the upper commutator ($\overline{\mathrm{com}}$, Beran's expression 81) is then dual.

q.e.d.

Theorem 8.6.8
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 23. If $x, y, z \in L$ such that x and z commute then each of the following expressions has a unique output regardless of the order in which the terms appear:

$$z * x * y * x$$
$$x * y * z * x$$

The Moufang identities (M3) and (M4) hold, and if $x \mathrm{C} y$ instead of $x \mathrm{C} z$ then also (M2).

Proof:
The operation with Beran's number 23 acts as the right projection on commuting elements, i.e. (1) $z * x = x$. From Table 8.1 it is (2) $x * y * x = x$, so we can simplify following expressions:

$$\begin{aligned}
(z * x) * (y * x) &\overset{(1)}{=} x * (y * x) &\overset{(2)}{=} x\,,\\
\big((z * x) * y\big) * x &\overset{(1)}{=} (x * y) * x &\overset{(2)}{=} x\,,\\
z * (x * y * x) &\overset{(2)}{=} z * x &\overset{(1)}{=} x\,.
\end{aligned}$$

To prove the case of $(z * (x * y)) * x$: although x need not commute with y, the element x always commutes with $x * y$. According to the assumption, x commutes with z, thus also with $z * (x * y)$. As $*$ acts as the right projection on commuting elements,

$$\big(z * (x * y)\big) * x = x\,. \tag{3}$$

The second constellation uses analogous arguments:

$$\begin{array}{lcccc} x * (y * z) * x & \overset{(2)}{=} & x\,, & & \\ x * \big(y * (z * x)\big) & \overset{(1)}{=} & x * (y * x) & \overset{(2)}{=} & x\,, \\ (x * y) * (z * x) & \overset{(1)}{=} & (x * y) * x & \overset{(2)}{=} & x\,, \\ \big((x * y) * z\big) * x & = & x\,, & & \end{array}$$

where the latter equality follows from $x \,\mathrm{C}\, (x * y) * z$.

q.e.d.

Similarly to Corollary 8.6.3, we may prove the following:

Corollary 8.6.9
Let L be an orthomodular lattice and let $\circledast$ be the operation with Beran's number 38. If $x, y, z \in L$ are such that x and z commute then each of the following expressions has a unique output regardless of the order in which the terms appear:

$$x \circledast y \circledast x \circledast z$$
$$x \circledast z \circledast y \circledast x$$

The Moufang identity (M1) holds and if $x \,\mathrm{C}\, y$ instead of $x \,\mathrm{C}\, z$ then also (M3) and (M4) hold.

The following Theorem is proven by Stephen M. Gagola III[2]:

Theorem 8.6.10
Let x, y and z be elements of an orthomodular lattice L, $*$ be the Sasaki projection and assume $y \,\mathrm{C}\, z$, then

$$\big((z * x) * y\big) * x = (z * x) * (y * x) = (z * x) * y\,.$$

Proof:
Some considerations in advance: Let L be an orthomodular lattice, the chain $s, x, p' \in L$, with $s \leq x \leq p'$ generates a Boolean subalgebra of L (because all the elements commute) in which

$$x \wedge (s \vee p) = s\,.$$

More generally, we can replace

s by a finite supremum $\bigvee_{s \in S} s$ of elements $s \leq x$ and

p by a finite supremum $\bigvee_{p \in P} p$ of elements $p \perp x$,

[2]Personal communication, April 2015

where $S, P \subseteq L$. We obtain

$$x \wedge \Big(\bigvee_{s \in S} s \vee \bigvee_{p \in P} p \Big) = \bigvee_{s \in S} s \,,$$

thus only the "small" elements, $s \in S$, determine the result. Finally, we admit also finitely many elements $c \in C \subseteq L$ which *commute with* x, i.e. each $c = (c \wedge x) \vee (c \wedge x')$ splits into a "small" element $c \wedge x \leq x$ and an orthogonal element $(c \wedge x') \perp x$, which are subject to the above rules,

$$\begin{aligned} x \wedge \Big(\bigvee_{s \in S} s \vee \bigvee_{p \in P} p \vee \bigvee_{c \in C} c \Big) &= x \wedge \Big(\bigvee_{s \in S} s \vee \bigvee_{c \in C} (c \wedge x) \vee \bigvee_{p \in P} p \vee \bigvee_{c \in C} (c \wedge x^{\perp}) \Big) \\ &= \bigvee_{s \in S} s \vee \bigvee_{c \in C} (c \wedge x) \,. \end{aligned} \tag{8.1}$$

It may happen that not all $c \in C$ commute with x. If C admits a partition to classes such that the supremum of each class commutes with x, we may apply the above procedure to these classes.

$$\begin{aligned} (z * x) * y &= (z * x) \wedge \big((z * x)' \vee y \big) \\ &= z \wedge (z' \vee x) \wedge \big(z' \vee (z \wedge x') \vee y \big) \\ &= (z' \vee x) \wedge z \wedge \Big(\underbrace{(z \wedge x')}_{\leqslant z} \vee \underbrace{z'}_{\perp z} \vee \underbrace{y}_{\mathrm{C} z} \Big) \\ &\overset{(8.1)}{=} (z' \vee x) \wedge \big((z' \vee x)' \vee (z \wedge y) \big) \\ &= (z' \vee x) * (z \wedge y) \end{aligned}$$

Then, inserting this result in (P1):

$$\big((z * x) * y \big) * x = \big((z' \vee x) * (z \wedge y) \big) * x \,,$$

If we substitute in the previous result $(z * x) * y = (z' \vee x) * (z \wedge y)$ the arguments z by $z' \vee x$, x by $z \wedge y$ and y by x then we get:

$$\begin{aligned} \big((z * x) * y \big) * x &= \big((z \wedge x') \vee (z \wedge y) \big) * \underbrace{\big((z' \vee x) \wedge x \big)}_{= x} \\ &= \big((z \wedge x') \vee (z \wedge y) \big) \wedge \Big(\big((z' \vee x) \wedge \underbrace{(z' \vee y')}_{\geqslant z'} \big) \vee \underbrace{x}_{\leqslant z' \vee x} \Big) \\ &= \big((z \wedge x') \vee (z \wedge y) \big) \wedge (z' \vee x) \\ &= (z * x) * y \,. \end{aligned}$$

Further for (P3):

$$
\begin{aligned}
(z * x) * (y * x) &= (z * x) \wedge \big((z * x)' \vee (y * x)\big) \\
&= \big(z \wedge (z' \vee x)\big) \wedge \Big(\big(z' \vee (z \wedge x')\big) \vee \big(y \wedge (y' \vee x)\big)\Big) \\
&= (z' \vee x) \wedge z \wedge \Big(\underbrace{(z \wedge x')}_{\leqslant z} \vee \underbrace{z' \vee \big(y \wedge (y' \vee x)\big)}_{\mathrm{C} z}\Big) \\
&\overset{(8.1)}{=} (z' \vee x) \wedge \Big((z \wedge x') \vee \big(z \wedge (z' \vee (y * x))\big)\Big) \\
&= (z' \vee x) \wedge \Big((z \wedge x') \vee \big(z * (y * x)\big)\Big) \\
&= (z' \vee x) * \big(z * (y * x)\big) \\
&\overset{(\text{Th. } 6.3.1)}{=} (z' \vee x) * \big((z * y) * x\big) \\
&\overset{(\text{Lemma } 3.2.2)}{=} (z' \vee x) * \big((z \wedge y) * x\big) \\
&= (z' \vee x) \wedge \Big((z \wedge x') \vee \big((z \wedge y) \wedge (z' \vee y' \vee x)\big)\Big) \\
&= (z' \vee x) \wedge \Big(\underbrace{(z \wedge x')}_{=:a} \vee \big(\underbrace{(z \wedge y)}_{=:b} \wedge (\underbrace{z' \vee y'}_{=b'} \vee \underbrace{z' \vee x}_{=a'})\big)\Big)
\end{aligned}
$$

Note: $a \vee \big(b \wedge (a' \vee b')\big) = a \vee b$, and thus

$$
\begin{aligned}
(z * x) * (y * x) &= (z' \vee x) \wedge (a \vee b) \\
&= (z' \vee x) \wedge \big((z \wedge x') \vee (z \wedge y)\big) \\
&= (z * x) * y\,.
\end{aligned}
$$

q.e.d.

8.6.3 Summary of results

Beran's no.	fulfils Moufang identity	condition	proven by
18 and 28	(M1), (M2), (M3), (M4)	$x \,\mathrm{C}\, y$	Theorem 8.6.4
34 and 44	(M1), (M2), (M3), (M4)	$x \,\mathrm{C}\, z$	Corollary 8.6.6
23	(M3), (M4)	$x \,\mathrm{C}\, z$	Theorem 8.6.8
23	(M2)	$x \,\mathrm{C}\, y$	Theorem 8.6.8
38	(M1)	$x \,\mathrm{C}\, z$	Corollary 8.6.9
38	(M3), (M4)	$x \,\mathrm{C}\, y$	Corollary 8.6.9

Table 8.3: Operations fulfilling Moufang identities.

Further results are summarised in Tables 8.4, 8.17 and 8.25. For each constellation, they show the common value of the quintuple of different ways of putting parentheses (if any) and *minimal* commutation assumptions under which they are equal. Value "`none`" means that there is no common value (in general) under the *maximal* commutation assumptions. For positive results, the column "Argument" refers to a theorem or to the applicable program (referenced by [33]). For negative results, it is a reference to the corresponding paragraph below presenting more details.

We made graphical representations, similar to the Tamari lattice of the results, Figures 8.30, 8.31 and 8.33.

The swapped projection

Beran's expression number 23 : $a * b = (a' \vee b) \wedge (a \vee (a' \wedge b))$

Constellation	Compatibility assumptions	Common value	Argument
$x * x * y * z$	$x \, \mathrm{C} \, z$	`none`	23.1
	$(\mathrm{C}x)$	$y * z$	Program [33]
	$(\mathrm{C}y)$	`none`	23.1
	$(\mathrm{C}z)$	z	Program [33]
$x * y * x * z$	$y \, \mathrm{C} \, z$	`none`	23.2
	$(\mathrm{C}x)$	`none`	23.2
	$(\mathrm{C}y)$	$x * z$	Program [33]
	$(\mathrm{C}z)$	z	Program [33]
$x * y * z * x$	$x \, \mathrm{C} \, z$	x	Theorem 8.6.8
	$(\mathrm{C}y)$	`none`	23.3
$y * x * x * z$	$y \, \mathrm{C} \, z$	`none`	23.4
	$(\mathrm{C}x)$	`none`	23.4
	$(\mathrm{C}y)$	$x * z$	Program [33]
	$(\mathrm{C}z)$	z	Program [33]
$z * x * y * x$	$x \, \mathrm{C} \, z$	x	Theorem 8.6.8
	$(\mathrm{C}y)$	`none`	23.5
$z * y * x * x$	$x \, \mathrm{C} \, z$	`none`	23.6
	$(\mathrm{C}x)$	x	Program [33]
	$(\mathrm{C}y)$	`none`	23.6
	$(\mathrm{C}z)$	$y * x$	Program [33]

Table 8.4: Results for the swapped projection.

Remark 8.6.11
For the swapped projection, as well as for the Sasaki projection, the following holds: $x \, \mathrm{C} \, (x * y)$, but not $y \, \mathrm{C} \, (x * y)$. Note that both projections are idempotent.

Argument 23.1

In general, without assuming commuting elements, we have following calculations:

$$\begin{aligned}
\text{P1}: &\ \big((x*x)*y\big)*z &&= &&(x*y)*z\\
\text{P2}: &\ \big(x*(x*y)\big)*z &&\overset{\text{(L)}}{=} &&(x*y)*z\\
\text{P3}: &\ (x*x)*(y*z) &&= &&x*(y*z)\\
\text{P4}: &\ x*\big((x*y)*z\big) && &&\text{no simplification possible}\\
\text{P5}: &\ x*\big(x*(y*z)\big) &&\overset{\text{(L)}}{=} &&x*(y*z)
\end{aligned}$$

The cases $x\,\mathrm{C}\,y$, $x\,\mathrm{C}\,z$, $y\,\mathrm{C}\,z$ and $(\mathrm{C}y)$ can be simplified further:

	$x\,\mathrm{C}\,y$	$x\,\mathrm{C}\,z$	$y\,\mathrm{C}\,z$	$(\mathrm{C}y)$
P1:	$y*z$	$(x*y)*z$	$(x*y)*z$ (◊2)	z
P2:	$y*z$	$(x*y)*z$	$(x*y)*z$ (◊2)	z
P3:	$x*(y*z)$	$x*(y*z)$	$x*z$	$x*z$
P4:	$x*(y*z)$	$(x*y)*z$ (◊1)	$x*\big((x*y)*z\big)$ (◊3)	$x*z$
P5:	$x*(y*z)$	$x*(y*z)$	$x*z$	$x*z$

Table 8.5: Summary of results for Argument 23.1.

(◊1) We assume $x\,\mathrm{C}\,z$, by Remark 8.6.11 also $x\,\mathrm{C}\,(x*y)$ holds. By Proposition 3.2.3 it follows that $x\,\mathrm{C}\,\big((x*y)*z\big)$. The swapped projection acts as the right projection if both arguments commute, it follows $\text{P4} = x*\big((x*y)*z\big) = (x*y)*z$.

(◊2) An example of $\text{P1} = (x*y)*z \neq x*z = \text{P3}$ can be found in L_{22} when choosing $x := a'$, $y := c$ and $z := d$, see Table 8.6.

(◊3) The condition $y\,\mathrm{C}\,z$ is not sufficient to fulfil the associativity equation $(x*y)*z = x*(y*z)$, see Table 6.5. Therefore $(x*y)*z \neq x*(y*z) = x*z$, thus $\text{P4} \neq \text{P3}$, P5.

Equality of P4, P1 and P2 would be true if $x*\big((x*y)*z\big) = (x*y)*z$, a counterexample to this can be found in MO_2 (Figure 5.2) by choosing $x := x$, $y := \mathbf{0}$ and $z := y$, see Table 8.6.

	$y\,\mathrm{C}\,z$ in L_{22} (◊2)	$y\,\mathrm{C}\,z$ in MO_2 (◊3)
$x*x*y*z$	$x := a'$, $y := c$, $z := d$	$x := x$, $y := \mathbf{0}$, $z := y$
P1	$\big((a'*a')*c\big)*d = f$	$\big((x*x)*\mathbf{0}\big)*y = y$
P2	$\big(a'*(a'*c)\big)*d = f$	$\big(x*(x*\mathbf{0})\big)*y = y$
P3	$(a'*a')*(c*d) = j$	$(x*x)*(\mathbf{0}*y) = x$
P4	$a'*\big((a'*c)*d\big) = f$	$x*\big((x*\mathbf{0})*y\big) = x$
P5	$a'*\big(a'*(c*d)\big) = j$	$x*\big(x*(\mathbf{0}*y)\big) = x$

Table 8.6: Counterexamples for Argument 23.1.

Argument 23.2

In general we can simplify only P1 and P2:

$$\text{P1}:\ \big((x*y)*x\big)*z \overset{(F)}{=} x*z$$
$$\text{P2}:\ \big(x*(y*x)\big)*z \overset{(F)}{=} x*z$$

The cases $x\,\mathrm{C}\,y$, $x\,\mathrm{C}\,z$, $y\,\mathrm{C}\,z$ and $(\mathrm{C}x)$ can be simplified further:

	$x\,\mathrm{C}\,y$	$x\,\mathrm{C}\,z$	$y\,\mathrm{C}\,z$	$(\mathrm{C}x)$
P1:	$x*z$	z	$x*z$	z
P2:	$x*z$	z	$x*z$	z
P3:	$y*(x*z)$	$(x*y)*z$ (◊3)	$(x*y)*(x*z)$ (◊4)	$y*z$
P4:	$x*z$	$x*\big((y*x)*z\big)$ (◊2)	$x*\big((y*x)*z\big)$ (◊4)	z
P5:	$y*(x*z)$ (◊1)	$x*(y*z)$ (◊3)	$x*\big(y*(x*z)\big)$ (◊4)	$y*z$

Table 8.7: Summary of results for Argument 23.2.

(◊1) We assume $x\,\mathrm{C}\,y$, by Remark 8.6.11 also $x\,\mathrm{C}\,(x*z)$ holds. By Proposition 3.2.3 it follows that $x\,\mathrm{C}\,\big(y*(x*z)\big)$. The swapped projection acts as the right projection if both arguments commute, thus $\text{P5} = x*\big(y*(x*z)\big) = y*(x*z)$.

Further, by accepting a more relaxed condition, also $x\,\mathrm{C}\,z$ is assumed, it can be proven that $y*(x*z) \neq x*z$, see the last column of Table 8.7 $(\mathrm{C}x)$.

(◊2) The counterexamples in Table 8.8, for $x\,\mathrm{C}\,z$ prove that P4 ≠ P1, P2 and P4 ≠ P3, P5.

(◊3) The condition $x\mathrm{C}z$ is not sufficient to fulfil the associativity equation, see Table 6.3, in general $(x*y)*z \neq x*(y*z)$, thus P3 ≠ P5.

(◊4) See counterexamples in Table 8.8: where P3, P4 and P5 have different results moreover they differ also from those of P1 and P2.

	$x\,\mathrm{C}\,z$ in L_{22} (◊2)	$y\,\mathrm{C}\,z$ in L_{22} (◊4)	$y\,\mathrm{C}\,z$ in L_{22} (◊4)
$x*y*x*z$	$x:=e,\ y:=b'\ z:=d'$	$x:=b,\ y:=e,\ z:=d'$	$x:=a,\ y:=c',\ z:=d'$
P1	$\big((e*b')*e\big)*d'=d'$	$\big((b*e)*b\big)*d'=g'$	$\big((a*c')*a\big)*d'=j'$
P2	$\big(e*(b'*e)\big)*d'=d'$	$\big(b*(e*b)\big)*d'=g'$	$\big(a*(c'*a)\big)*d'=j'$
P3	$(e*b')*(e*d')=j'$	$(b*e)*(b*d')=g'$	$(a*c')*(a*d')=f'$
P4	$e*\big((b'*e)*d'\big)=e$	$b*\big((e*b)*d'\big)=g'$	$a*\big((c'*a)*d'\big)=b'$
P5	$e*\big(b'*(e*d')\big)=j'$	$b*\big(e*(b*d')\big)=b$	$a*\big(c'*(a*d')\big)=f'$

Table 8.8: Counterexamples for Argument 23.2.

Argument 23.3

In general, without commuting elements we can simplify P2 and P4:

$$\text{P2}: \ \big(x * (y * z)\big) * x \overset{(F)}{=} x$$
$$\text{P4}: \ x * \big((y * z) * x\big) \overset{(F)}{=} x$$

For the cases $x \,\mathrm{C}\, y$, $y \,\mathrm{C}\, z$ and $(\mathrm{C}y)$ we can calculate further:

	$x \,\mathrm{C}\, y$		$y \,\mathrm{C}\, z$		$(\mathrm{C}y)$
P1:	$(y * z) * x$	(◊1)	$\big((x * y) * z\big) * x$	(◊2)	$z * x$
P2:	x		x		x
P3:	$y * (z * x)$	(◊1)	$(x * y) * (z * x)$	(◊2)	$z * x$
P4:	x		x		x
P5:	$x * \big(y * (z * x)\big)$	(◊1)	$x * \big(y * (z * x)\big)$	(◊3)	x

Table 8.9: Summary of results for Argument 23.3.

(◊1) The condition $x \,\mathrm{C}\, y$ is not sufficient for $(y * z) * x = y * (z * x)$, see Table 6.5. Moreover a counterexample proving P1 ≠ P3, P5 is found in Table 8.10.

A more relaxed condition, $y \,\mathrm{C}\, z$ is additionally assumed, is summarised in the last column of Table 8.9 $(\mathrm{C}y)$ ((◊4) and contains a counterexample, thus P5 ≠ P1, P3.

(◊2) Table 8.10 shows a counterexample to prove that the result of P3 can differ from the results of any other possibilities of putting parentheses.

(◊3) If $y \,\mathrm{C}\, x$ is additionally assumed, thus a more strict assumption, i.e. $(\mathrm{C}y)$ but not $x \mathrm{C} z$, then the last column of Table 8.9 contains a counterexample and P1 $= z * x \neq x =$ P5.

We wanted to prove that P5 ≠ P2 = P4, in other words $x * \big(y * (z * x)\big) \neq x$, with $y \,\mathrm{C}\, z$. But, although this is rather unlikely, exactly this result is obtained in all orthomodular lattices we considered. At this point we do not have a good explanation for this.

The cases $(\mathrm{C}x)$ and $(\mathrm{C}z)$ are covered by the sufficient condition $x \,\mathrm{C}\, z$.

	$x \,\mathrm{C}\, y$ in L_{22} (◊1)	$y \,\mathrm{C}\, z$ in L_{22} (◊2)
$x * y * z * x$	$x := a,\ y := b'\ z := i$	$x := d,\ y := b'\ z := a'$
P1:	$\big((a * b') * i\big) * a = a$	$\big((d * b') * a'\big) * d = d$
P2:	$\big(a * (b' * i)\big) * a = a$	$\big(d * (b' * a')\big) * d = d$
P3:	$(a * b') * (i * a) = b'$	$(d * b') * (a' * d) = h'$
P4:	$a * \big((b' * i) * a\big) = a$	$d * \big((b' * a') * d\big) = d$
P5:	$a * \big(b' * (i * a)\big) = b'$	$d * \big(b' * (a' * d)\big) = d$

Table 8.10: Counterexamples for Argument 23.3.

Argument 23.4

The general case, no elements commute, the following calculations can be done:

$$
\begin{aligned}
\text{P1}:&\ \big((y*x)*x\big)*z &\overset{(R)}{=}&\ (y*x)*z\\
\text{P2}:&\ \big(y*(x*x)\big)*z &\overset{(R)}{=}&\ (y*x)*z\\
\text{P3}:&\ (y*x)*(x*z) && \text{no simplification possible}\\
\text{P4}:&\ y*\big((x*x)*z\big) &\overset{(L)}{=}&\ y*(x*z)\\
\text{P5}:&\ y*\big(x*(x*z)\big) &\overset{(L)}{=}&\ y*(x*z)
\end{aligned}
$$

For the cases $x\,\mathrm{C}\,y$, $x\,\mathrm{C}\,z$, $y\,\mathrm{C}\,z$ and $(\mathrm{C}x)$ we can calculate further:

	$x\,\mathrm{C}\,y$	$x\,\mathrm{C}\,z$	$y\,\mathrm{C}\,z$	$(\mathrm{C}x)$
P1:	$x*z$	$(y*x)*z$	$(y*x)*z$	z
P2:	$x*z$	$(y*x)*z$	$(y*x)*z$	z
P3:	$x*z$	$(y*x)*z$	$(y*x)*(x*z)$ (◊1)	z
P4:	$y*(x*z)$	$y*z$	$y*(x*z)$	$y*z$
P5:	$y*(x*z)$	$y*z$	$y*(x*z)$	$y*z$

Table 8.11: Summary of results for Argument 23.4.

(◊1) See the counterexamples in Table 8.12 where the five expressions give three different results, i.e. P3 ≠ P1 = P2 ≠ P4 = P5 ≠ P3.

	$y\,\mathrm{C}\,z$ in L_{22}(◊1)
$y*x*x*z$	$x:=a,\ y:=c'\ z:=d'$
P1:	$\big((c'*a)*a\big)*d' = b'$
P2:	$\big(c'*(a*a)\big)*d' = b'$
P3:	$(c'*a)*(a*d') = g$
P4:	$c'*\big((a*a)*d'\big) = c'$
P5:	$c'*\big(a*(a*d')\big) = c'$

Table 8.12: Counterexample for Argument 23.4.

Argument 23.5

In general we can simplify P4 and P5 in the following way:

$$
\begin{aligned}
\text{P4}:&\ z*\big((x*y)*x\big) &\overset{(F)}{=}&\ z*x\\
\text{P5}:&\ z*\big(x*(y*x)\big) &\overset{(F)}{=}&\ z*x
\end{aligned}
$$

For the cases $x\,\mathrm{C}\,y$, $y\,\mathrm{C}\,z$ and $(\mathrm{C}y)$ we can calculate further:

	$x \,\mathrm{C}\, y$		$y \,\mathrm{C}\, z$		$(\mathrm{C}y)$
P1:	$\big((z * x) * y\big) * x$	($\lozenge$1)	$\big((z * x) * y\big) * x$	($\lozenge$3)	x
P2:	$(z * y) * x$	($\lozenge$2)	$\big(z * (x * y)\big) * x$	($\lozenge$3)	x
P3:	$z * x$		$(z * x) * (y * x)$	($\lozenge$4)	$z * x$
P4:	$z * x$		$z * x$		$z * x$
P5:	$z * x$		$z * x$		$z * x$

Table 8.13: Summary of results for Argument 23.5.

($\lozenge$1) A more relaxed condition, also $y \,\mathrm{C}\, z$ is supposed, is summarised in the last column of Table 8.13 ($\mathrm{C}y$) and proves the inequality of P1 with P3, P4 and P5.

($\lozenge$2) The condition $x \,\mathrm{C}\, y$ is not sufficient for the associativity equation $(z * x) * y = z * (x * y)$. In Table 8.14 a counterexample proves that P2 has an other result as all other ways of putting the parentheses.

($\lozenge$3) A counterexample that $\big((z * x) * y\big) * x = \big(z * (x * y)\big) * x$ does not hold can be found in Table 8.14.

A stronger condition, $y \,\mathrm{C}\, z$ is also supposed, is summarised in the last column of Table 8.13 and gives an example of P1 $\neq$ P3 = P4 = P5.

($\lozenge$4) The last column from Table 8.13, as well as the counterexample from Table 8.14, shows that P3 can have an other result as P2, P4 and P5 is shown in Table 8.14.

The cases ($\mathrm{C}x$) and ($\mathrm{C}z$) are covered by the sufficient condition $x \,\mathrm{C}\, z$.

	$x \,\mathrm{C}\, y$ in L_{22} ($\lozenge$2)	$y \,\mathrm{C}\, z$ in MO_2 ($\lozenge$3), ($\lozenge$4)
$z * x * y * x$	$x := c,\ y := d\ z := a'$	$x := x,\ y := y,\ z := y'$
P1:	$\big((a' * c) * d\big) * c = f$	$\big((y' * x) * y\big) * x = y$
P2:	$\big(a' * (c * d)\big) * c = j$	$\big(y' * (x * y)\big) * x = y'$
P3:	$(a' * c) * (d * c) = f$	$(y' * x) * (y * x) = y$
P4:	$a' * \big((c * d) * c\big) = f$	$y' * \big((x * y) * x\big) = y'$
P5:	$a' * \big(c * (d * c)\big) = f$	$y' * \big(x * (y * x)\big) = y'$

Table 8.14: Counterexamples for Argument 23.5.

Argument 23.6

In general we have following calculations:

$$\begin{aligned}
\text{P1}: &\ \big((z * y) * x\big) * x \overset{(R)}{=} (z * y) * x \\
\text{P2}: &\ \big(z * (y * x)\big) * x \quad \text{no simplification possible} \\
\text{P3}: &\ (z * y) * (x * x) = (z * y) * x \\
\text{P4}: &\ z * \big((y * x) * x\big) \overset{(R)}{=} z * (y * x) \\
\text{P5}: &\ z * \big(y * (x * x)\big) \overset{(R)}{=} z * (y * x)
\end{aligned}$$

For the cases $x \,\mathrm{C}\, y$, $x \,\mathrm{C}\, z$, $y \,\mathrm{C}\, z$ and $(\mathrm{C}y)$, we can calculate further:

	$x \,\mathrm{C}\, y$	$x \,\mathrm{C}\, z$	$y \,\mathrm{C}\, z$	$(\mathrm{C}y)$
P1:	$(z * y) * x$ (◊1)	$(z * y) * x$ (◊2)	$y * x$	x
P2:	$z * x$	$\big(z * (y * x)\big) * x$ (◊3)	$\big(z * (y * x)\big) * x$ (◊4)	$z * x$
P3:	$(z * y) * x$	$(z * y) * x$	$y * x$	x
P4:	$z * x$	$z * (y * x)$ (◊3)	$z * (y * x)$	$z * x$
P5:	$z * x$	$z * (y * x)$	$z * (y * x)$	$z * x$

Table 8.15: Summary of results for Argument 23.6.

(◊1) A counterexample to $(z * y) * x \neq z * x$ can be found in MO_2 by choosing $x := x$, $y := y$ and $z = \mathbf{0}$:

$$\begin{aligned}
\text{P1, P3}: &\ (\mathbf{0} * y) * x = y \\
\text{P2, P4, P5}: &\ \mathbf{0} * x = x
\end{aligned}$$

(◊2) The condition $x \,\mathrm{C}\, z$ is not sufficient to fulfil the associativity equation $(z * y) * x = z * (y * x)$, see Table 6.5. Thus P1 = P3 ≠ P4 = P5.

(◊3) The counterexample in Table 8.16 proves that P2 can have other results as all other possibilities of putting parentheses.

(◊4) The last culumn of Table 8.15 shows that P2 can have other results as P1 and P3 moreover Table 8.16 includes a counterexample.

At this point we do not have a good explanation that $z * (y * x) \neq z * (y * x) * x$, when $y \,\mathrm{C}\, z$ (to prove that P2 ≠ P4 = P5). But exactly this result is obtained in all orthomodular lattices we considered. At this point we do not have a good explanation for this.

	$x \,\mathrm{C}\, z$ in L_{22} ($\Diamond 3$)	$y \,\mathrm{C}\, z$ in L_{22} ($\Diamond 4$)
$z * y * x * x$	$x := a',\ y := d,\ z := b'$	$x := d',\ y := b,\ z := a$
P1:	$\big((b' * d) * a'\big) * a' = c'$	$\big((a * b) * d'\big) * d' = g'$
P2:	$\big(b' * (d * a')\big) * a' = a'$	$\big(a * (b * d')\big) * d' = f'$
P3:	$(b' * d) * (a' * a') = c'$	$(a * b) * (d' * d') = g'$
P4:	$b' * \big((d * a') * a'\big) = b'$	$a * \big((b * d') * d'\big) = f'$
P5:	$b' * \big(d * (a' * a')\big) = b'$	$a * \big(b * (d' * d')\big) = f'$

Table 8.16: Counterexamples for Argument 23.6.

The Sasaki Projection

Beran's expression number 18 : $a * b = a \wedge (a' \vee b)$

Constellation	Compatibility assumptions	Common value	Argument
$x * x * y * z$	$x \,\mathrm{C}\, y$	$(x \wedge y) * z$	Theorem 8.6.4
	$x \,\mathrm{C}\, z$ or $y \,\mathrm{C}\, z$	`none`	18.1
	$(\mathrm{C}z)$	$(x * y) \wedge z$	Program [33]
$x * y * x * z$	$x \,\mathrm{C}\, y$	$(x \wedge y) * z$	Theorem 8.6.4
	$x \,\mathrm{C}\, z$ or $y \,\mathrm{C}\, z$	`none`	18.2
	$(\mathrm{C}z)$	$(x * y) \wedge z$	Program [33]
$x * y * z * x$	$x \,\mathrm{C}\, y$	$(x \wedge y) * z$	Theorem 8.6.4
	$x \,\mathrm{C}\, z$ or $y \,\mathrm{C}\, z$	`none`	18.3
	$(\mathrm{C}z)$	$(x * y) \wedge z$	Program [33]
$y * x * x * z$	$x \,\mathrm{C}\, y$	$(x \wedge y) * z$	Theorem 8.6.4
	$x \,\mathrm{C}\, z$ or $y \,\mathrm{C}\, z$	`none`	18.4
	$(\mathrm{C}z)$	$(y * x) \wedge z$	Program [33]
$z * x * y * x$	$x \,\mathrm{C}\, y$ or $y \,\mathrm{C}\, z$	`none`	18.5
	$x \,\mathrm{C}\, z$	$(x \wedge z) * y$	Theorem 8.6.4
	$(\mathrm{C}y)$	$(z * x) \wedge y$	Program [33]
$z * y * x * x$	$x \,\mathrm{C}\, y$	`none`	18.6
	$x \,\mathrm{C}\, z$	`none`	18.6
	$y \,\mathrm{C}\, z$	$(y \wedge z) * x$	Theorem 8.6.4
	$(\mathrm{C}x)$	$(z * y) \wedge x$	Program [33]

Table 8.17: Results for the Sasaki projection.

Remark 8.6.12

1. One of many properties of the Sasaki projection is the following:

$$x \wedge y \leqslant x \wedge (x' \vee y) \leqslant x$$

see [36] page 156.

2. For the Sasaki projection, one non-sufficient condition for fulfilling the associativity identity is $z < x$ as can been calculated in the orthomodular lattice $\mathbf{2^2} \oplus \mathbf{2^3}$ (Figure 7.2) by substituting (x, y, z) by (t', z, x):

$$t' * (z * x) = t' * z = t' \quad \neq \quad (t' * z) * x = t' * x = x\,.$$

Argument 18.1

In general, without commuting elements, we have following calculations:

$$\begin{aligned}
\text{P1}: \quad & \big((x * x) * y\big) * z && = && (x * y) * z \\
\text{P2}: \quad & \big(x * (x * y)\big) * z && \overset{\text{(L)}}{=} && \big((x * x) * y\big) * z \quad = \quad (x * y) * z \\
\text{P3}: \quad & (x * x) * (y * z) && = && x * (y * z) \\
\text{P4}: \quad & x * \big((x * y) * z\big) && \overset{(\Diamond 1)}{=} && (x * y) * z \\
\text{P5}: \quad & x * \big(x * (y * z)\big) && \overset{\text{(L)}}{=} && x * (y * z)
\end{aligned}$$

($\Diamond$1) By Remark 8.6.11 we know that $x \,\mathrm{C}\, (x * y)$ and by Theorem 6.3.1 it follows:

$$\text{P4}: \quad x * \big((x * y) * z\big) \quad = \quad \big(x * (x * y)\big) * z \quad : \text{P2}$$

thus

$$\text{P4}: \quad x * \big((x * y) * z\big) \quad = \quad (x * y) * z\,.$$

From Section 6.3.1 it follows that for $x \,\mathrm{C}\, z$ or $y \,\mathrm{C}\, z$ the equation $(x * y) * z = x * (y * z)$ does not necessarily hold.

The cases $(\mathrm{C}x)$ and $(\mathrm{C}y)$ are covered by the cases $x \,\mathrm{C}\, y$ and $(\mathrm{C}x)$.

Argument 18.2

In general the following holds:

$$\begin{aligned}
\text{P1}: \quad & \big((x * y) * x\big) * z && \overset{\text{(F)}}{=} && (x * y) * z \\
\text{P2}: \quad & \big(x * (y * x)\big) * z && \overset{\text{(F)}}{=} && (x * y) * z \\
\text{P3}: \quad & (x * y) * (x * z) && \overset{\text{(Th. 8.6.2)}}{=} && (x * y * x) * z \overset{\text{(F)}}{=} (x * y) * z \\
\text{P4}: \quad & x * \big((y * x) * z\big) && && \text{no simplification possible} \\
\text{P5}: \quad & x * \big(y * (x * z)\big) && && \text{no simplification possible}
\end{aligned}$$

For the cases $z \leqslant x$ or $x \leqslant z$ and the case $y \leqslant z$ we can calculate further:

	$z \leqslant x$		$x \leqslant z$		$y \leqslant z$	
P1:	$(x * y) * z$		$x * y$		$y * x$	
P2:	$(x * y) * z$		$x * y$		$y * x$	
P3:	$(x * y) * z$		$x * y$		$y * x$	
P4:	$x * (y * z)$	(◊1)	$x * y$	(◊1)	$y * x$	(◊3)
P5:	$x * (y * z)$	(◊2)	$x * y$	(◊2)	$y * x$	(◊4)

Table 8.18: Summary of results for Argument 18.2.

(◊1) If $x \,\mathrm{C}\, z$, then we can find a counterexample that P4 has a different result as all other ways of putting the parentheses, see Table 8.19.

a) In case $z \leqslant x$ we apply Lemma 8.6.1 (S6):

$$\text{P4}: \quad x * \big((y * x) * z\big) \quad = \quad x * (y * z) \quad = \quad \text{P5}.$$

b) In case $x \leqslant z$ we apply Lemma 8.6.1 (S5):

$$\text{P4}: \quad x * \big((y * x) * z\big) \quad = \quad x * (y * x) \quad = \quad x * y.$$

(◊2) For $x \,\mathrm{C}\, z$ we can calculate further

$$\text{P5}: \quad x * \big(y * (x * z)\big) \overset{\text{(Lemma 3.2.2)}}{=} x * \big(y * (x \wedge z)\big)$$

a) If we assume $z \leqslant x$ in addition, then

$$\text{P5}: \quad x * \big(y * (x * z)\big) \quad = \quad x * \big(y * (x \wedge z)\big) \quad = \quad x * (y * z).$$

b) In the case that $x \leqslant z$

$$\text{P5}: \quad x * \big(y * (x \wedge z)\big) \quad = \quad x * (y * x) \quad = \quad x * y = \text{P1, P2, P3}.$$

(◊3) If $y \,\mathrm{C}\, z$, but $y \nleqslant z$, then we can find a counterexample that P4 has a different result as all other ways of putting the parentheses, see Table 8.19.

If we assume $y \leqslant z$, then by Lemma 8.6.1 (S5) and Table 8.1:

$$\text{P4} = x * \big((y * x) * z\big) = x * (y * x) = x * y,$$

but for $z \leqslant y$ nothing can be said.

(◊4) For parentheses P5 and $y \,\mathrm{C}\, z$:

a) With the additional assumption that $y \leqslant z$, by Lemma 8.6.1 (S5) it follows

$$y * (x * z) = (y * x) * z = y * x.$$

In this case P5 = P1, P2, P3.

b) But the contrary assumption $z < y$ is not a sufficient condition for $y * (x * z) = (y * x) * z$, see Remark 8.6.12.

The cases (Cx) and (Cy) are covered by the condition $x \,\mathrm{C}\, y$ of Theorem 8.6.4.

	$x \,\mathrm{C}\, z$ in MO_2 (◊1)	$y \,\mathrm{C}\, z$ in MO_2 (◊3)
$x * y * x * z$	$x := a,\ y := b\ z := a'$	$x := b,\ y := a,\ z := a'$
P1:	$\big((a * b) * a\big) * a' = \mathbf{0}$	$\big((b * a) * b\big) * a' = b$
P2:	$\big(a * (b * a)\big) * a' = \mathbf{0}$	$\big(b * (a * b)\big) * a' = b$
P3:	$(a * b) * (a * a') = \mathbf{0}$	$(b * a) * (b * a') = b$
P4:	$a * \big((b * a) * a'\big) = a$	$b * \big((a * b) * a'\big) = \mathbf{0}$
P5:	$a * \big(b * (a * a')\big) = \mathbf{0}$	$b * \big(a * (b * a')\big) = b$

Table 8.19: Counterexamples for Argument 18.2.

Argument 18.3

In general we have

$$
\begin{aligned}
&\text{P1}: && \big((x * y) * z\big) * x && \overset{\text{(Th. 8.6.2)}}{=} && \text{P3} \\
&\text{P2}: && \big(x * (y * z)\big) * x && \overset{\text{(F)}}{=} && x * (y * z) \\
&\text{P3}: && (x * y) * (z * x) && \overset{\text{(Th. 8.6.2)}}{=} && \text{P1} \\
&\text{P4}: && x * \big((y * z) * x\big) && \overset{\text{(F)}}{=} && x * (y * z) \\
&\text{P5}: && x * \big(y * (z * x)\big) && && \text{no simplification possible}
\end{aligned}
$$

For the cases $z \leqslant x$ or $z \leqslant x$ and $y \,\mathrm{C}\, z$ we can calculate further:

	$z \leqslant x$		$x \leqslant z$		$y \,\mathrm{C}\, z$	
P1:	$(x * y) * z$		$x * y$		$=$ P3	(◊3)
P2:	$x * (y * z)$		$x * y$		$x * (y \wedge z)$	(◊2)
P3:	$(x * y) * z$	(◊1)	$x * y$	(◊1)	$=$ P1	(◊3)
P4:	$x * (y * z)$		$x * y$		$x * (y \wedge z)$	
P5:	$x * (y * z)$	(◊1)	$x * y$	(◊1)	$x * (y \wedge z)$	

Table 8.20: Summary of results for Argument 18.3.

(◊1) For $x \,\mathrm{C}\, z$ we have

$$
\begin{aligned}
&\text{P1}: && (x * y) * (z \wedge x) \\
&\text{P3}: && (x * y) * (z \wedge x) \\
&\text{P5}: && x * \big(y * (z \wedge x)\big)
\end{aligned}
$$

a) For $z \leqslant x$,

$$
\begin{aligned}
&\text{P3}: && (x * y) * (z * x) &&=&& (x * y) * (z \wedge x) &&=&& (x * y) * z \\
&\text{P5}: && x * \big(y * (z * x)\big) &&=&& x * \big(y * (z \wedge x)\big) &&=&& x * (y * z)
\end{aligned}
$$

From Section 6.3.1 follows that even if $x \mathrm{C} z$ the equation $(x * y) * z = x * (y * z)$ does not necessarily hold.

b) For $x \leqslant z$,

$$\begin{aligned} \text{P3}: \quad (x * y) * (z \wedge x) &= (x * y) * x \overset{(F)}{=} x * y \\ \text{P5}: \quad x * \big(y * (z \wedge x)\big) &= x * (y * x) \overset{(F)}{=} x * y \end{aligned}$$

which is equal to P2 and P4, all results are then equal.

(◊2) If $y \mathrel{C} z$ we can calculate further:

$$\begin{aligned} \text{P2}: \quad \big(x * (y * z)\big) * x &= \big(x * (y \wedge z)\big) * x \overset{(F)}{=} x * (y \wedge z) \\ \text{P4}: \quad x * \big((y * z) * x\big) &= x * \big((y \wedge z) * x\big) \overset{(F)}{=} x * (y \wedge z) \\ \text{P5}: \quad x * \big(y * (z * x)\big) &\overset{(\text{Th. } 6.3.1)}{=} x * \big((y * z) * x\big) \overset{(F)}{=} x * (y \wedge z) \end{aligned}$$

(◊3) A counterexample proving that P1 and P3 can have different results as the other possibilities of putting the parentheses can be found on MO_2 by choosing $x := x$, $y := y$ and $z := y'$

$$\begin{aligned} \text{P1}, \text{P3}: \quad (x * y) * (y' * x) &= x \\ \text{P2}: \quad \big(x * (y * y')\big) * x &= \mathbf{0} = \text{P4, P5} \end{aligned}$$

The cases (Cx) and (Cy) are covered by the condition $x \mathrel{C} y$ of Theorem 8.6.4.

Argument 18.4

In general we have

$$\begin{aligned} \text{P1}: \quad \big((y * x) * x\big) * z &\overset{(R)}{=} (y * x) * z \\ \text{P2}: \quad \big(y * (x * x)\big) * z &\overset{(R)}{=} (y * x) * z \\ \text{P3}: \quad (y * x) * (x * z) &\overset{(\Diamond)}{=} y * (x * z) \\ \text{P4}: \quad y * \big((x * x) * z\big) &\overset{(L)}{=} y * (x * z) \\ \text{P5}: \quad y * \big(x * (x * z)\big) &\overset{(L)}{=} y * (x * z) \end{aligned}$$

$$(\Diamond) \quad \text{P3}: \quad (y * x) * \underbrace{(x * z)}_{\leqslant x} \overset{(S6)}{=} y * (x * z) = \text{P4}, \text{P5} \qquad \text{(by Remark 8.6.12)}$$

The Sasaki projection is not associative in general, by Section 6 it is thus $(y * x) * z \neq y * (x * z)$.

The cases (Cx) and (Cy) are covered by the condition $x \mathrel{C} y$ and (Cz).

Argument 18.5

In general we have

$$\begin{array}{llll} \text{P1}: & \big((z*x)*y\big)*x & & \text{no simplification possible} \\ \text{P2}: & \big(z*(x*y)\big)*x & \overset{\text{(Th. 8.6.2)}}{=} & z*(x*y*x) \overset{\text{(F)}}{=} z*(x*y) \\ \text{P3}: & (z*x)*(x*y) & & \text{no simplification possible} \\ \text{P4}: & z*\big((x*y)*x\big) & \overset{\text{(F)}}{=} & z*(x*y) \\ \text{P5}: & z*\big(x*(y*x)\big) & \overset{\text{(F)}}{=} & z*(x*y) \end{array}$$

For the cases $x \,\mathrm{C}\, y$ or $x \leqslant y$ or $y \leqslant x$ or $y \,\mathrm{C}\, z$ we can calculate further:

	$x \,\mathrm{C}\, y$		$x \leqslant y$ or $y \leqslant x$		$y \,\mathrm{C}\, z$	
P1:	$\big((z*x)*y\big)*x$	(◊1)	$z*(x\wedge y)$	(◊1)	$(z*x)*y$	(◊2)
P2:	$z*(x\wedge y)$		$z*(x\wedge y)$		$z*(x*y)$	
P3:	$z*(x\wedge y)$	(◊1)	$z*(x\wedge y)$	(◊1)	$(z*x)*y$	(◊2)
P4:	$z*(x\wedge y)$		$z*(x\wedge y)$		$z*(x*y)$	
P5:	$z*(x\wedge y)$		$z*(x\wedge y)$		$z*(x*y)$	

Table 8.21: Summary of results for Argument 18.5.

(◊1) For $x \,\mathrm{C}\, y$, the counterexample in Table 8.22 shows that P1 is not equal to any of the expressions P2 to P5. Further by $y\wedge x \leqslant x$:

$$\text{P3}: \quad (z*x)*(y*x) = (z*x)*(y\wedge x) \overset{\text{(S6)}}{=} z*(x\wedge y)$$

a) If we additionally assume that $x \leqslant y$ then we have:

$$\begin{array}{lllll} \text{P1}: & \big((z*x)*y\big)*x & \overset{\text{(S4)}}{=} & (z*x)*x & = z*x \\ \text{P3}: & (z*x)*(y\wedge x) & = & (z*x)*x & = z*x \end{array}$$

which is in this case equal to the results of P2, P4 and P5.

b) By assuming $y \leqslant x$:

$$\begin{array}{llllll} \text{P1}: & \big((z*x)*y\big)*x & \overset{\text{(S6)}}{=} & (z*x)*(y*x) & = \text{P3} & \\ \text{P3}: & (z*x)*(y\wedge x) & = & (z*x)*y & \overset{\text{(S6)}}{=} z*(x*y) & = z*y\,. \end{array}$$

Thus also equal to the results of P2, P4 and P5.

(◊2) The case $y \,\mathrm{C}\, z$ is solved by Theorem 8.6.10, it proves

$$\text{P1} = \big((z*x)*y\big)*x = (z*x)*(y*x) = (z*x)*y = \text{P3}\,.$$

From Section 6.3.1 it follows that even if $y \,\mathrm{C}\, z$ the equation $(z*x)*y = z*(x*y)$ does not necessarily hold, see also the counterexample in Table 8.22.

The cases (Cx) and (Cz) are covered by the condition $x \,\mathrm{C}\, z$ of Theorem 8.6.4, where y and z are interchanged.

	$x \,\mathrm{C}\, y$ in MO_2 (◊1)	$y \,\mathrm{C}\, z$ in MO_2 (◊2)
$z * x * y * x$	$x := a,\ y := a'\ z := b$	$x := b,\ y := a\ z := a'$
P1:	$\big((b * a) * a'\big) * a = b$	$\big((a' * b) * a\big) * b = \mathbf{0}$
P2:	$\big(b * (a * a')\big) * a = \mathbf{0}$	$\big(a' * (b * a)\big) * b = a'$
P3:	$(b * a) * (a' * a) = \mathbf{0}$	$(a' * b) * (a * b) = \mathbf{0}$
P4:	$b * \big((a * a') * a\big) = \mathbf{0}$	$a' * \big((b * a) * b\big) = a'$
P5:	$b * \big(a * (a' * a)\big) = \mathbf{0}$	$a' * \big(b * (a * b)\big) = a'$

Table 8.22: Counterexamples for Argument 18.5.

Argument 18.6

In general we have

$$\begin{array}{lll} \mathrm{P1}: & \big((z * y) * x\big) * x & \overset{(\mathrm{R})}{=} & (z * y) * x \\ \mathrm{P2}: & \big(z * (y * x)\big) * x & & \text{no simplification possible} \\ \mathrm{P3}: & (z * y) * (x * x) & = & (z * y) * x \\ \mathrm{P4}: & z * \big((y * x) * x\big) & \overset{(\mathrm{R})}{=} & z * (y * x) \\ \mathrm{P5}: & z * \big(y * (x * x)\big) & = & z * (y * x) \end{array}$$

For the cases $x \,\mathrm{C}\, y$ or $x \leqslant y$ or $y \leqslant x$ or the case $x \,\mathrm{C}\, z$ we can calculate further:

	$x \,\mathrm{C}\, y$		$x \leqslant y$ or $y \leqslant x$		$x \,\mathrm{C}\, z$	
P1:	$(z * y) * x$		$z * (y \wedge x)$	(◊1)	$(z * y) * x$	(◊2)
P2:	$z * (y \wedge x)$	(◊1)	$z * (y \wedge x)$	(◊1)	$\big(z * (y * x)\big) * x$	(◊2)
P3:	$(z * y) * x$		$z * (y \wedge x)$		$(z * y) * x$	
P4:	$z * (y \wedge x)$		$z * (y \wedge x)$		$z * (y * x)$	(◊2)
P5:	$z * (y \wedge x)$		$z * (y \wedge x)$		$z * (y * x)$	

Table 8.23: Summary of results for Argument 18.6.

(◊1) The assumption $x \,\mathrm{C}\, y$ is not sufficient for associativity of the Sasaki projection, by Section 6. An example that $\mathrm{P2} = \big(z * (y \wedge x)\big) * x \neq (z * y) * x = \mathrm{P1}, \mathrm{P3}$ can be found in Table 8.24.
On the other hand, while $y \wedge x \leqslant x$ it is

$$\mathrm{P2}: \quad \big(z * (y * x)\big) * x \;=\; \big(z * (y \wedge x)\big) * x \;\overset{(\mathrm{S4})}{=}\; z * \big((y \wedge x) * x\big) \;=\; \mathrm{P4}, \mathrm{P5}$$

a) If $x \leqslant y$ then

$$\mathrm{P4}, \mathrm{P5}: \quad z * (y * x) \;=\; z * (y \wedge x) \;=\; z * x$$

By Lemma 8.6.1 (S6), $z * (y * x) = (z * y) * x$. All results are then equal.

b) If $x \geqslant y$ then, by Theoem 8.6.1 (S4):

$$\text{P2}: \quad \big(z * (y \wedge x)\big) * x \quad = \quad (z * y) * x \quad = \quad z * (y * x).$$

Also in this case all results are equal to $z * (y \wedge x) = z * y$.

($\lozenge$2) In case $x \,\mathrm{C}\, z$ then P2 $\neq$ P4, P5, see the counterexample in Table 8.24. In the same table is also an example proving that P2 $\neq$ P1, P3.

The cases (Cy) and (Cz), are covered by $y \,\mathrm{C}\, z$ and by Theoren 6.3.1 it is easy to prove that in this case all ways to put parentheses have equal results.

	$x \,\mathrm{C}\, y$ in L_{22} ($\lozenge$1)	$x \,\mathrm{C}\, z$ in MO_2 ($\lozenge$2)	$x \,\mathrm{C}\, z$ in L_{22} ($\lozenge$2)
$z * y * x * x$	$x := b,\ y := c,\ z := e$	$x := x,\ y := y,\ z := x'$	$x := a,\ y := i',\ z := b'$
P1:	$\big((e * c) * b\big) * b = e$	$\big((x' * y) * x\big) * x = \mathbf{0}$	$\big((b' * i') * a\big) * a = a$
P2:	$\big(e * (c * b)\big) * b = \mathbf{0}$	$\big(x' * (y * x)\big) * x = \mathbf{0}$	$\big(b' * (i' * a)\big) * a = g$
P3:	$(e * c) * (b * b) = e$	$(x' * y) * (x * x) = \mathbf{0}$	$(b' * i') * (a * a) = a$
P4:	$e * \big((c * b) * b\big) = \mathbf{0}$	$x' * \big((y * x) * x\big) = x'$	$b' * \big((i' * a) * a\big) = g$
P5:	$e * \big(c * (b * b)\big) = \mathbf{0}$	$x' * \big(y * (x * x)\big) = x'$	$b' * \big(i' * (a * a)\big) = g$

Table 8.24: Counterexamples for Argument 18.6.

Lower commutator

Beran's expression number 16 : $a * b = (a \wedge b) \vee (a' \wedge b) \vee (a \wedge b') \vee (a' \wedge b')$

Constellation	Compatibility assumptions	Common value	Argument
$x * x * y * z$	$x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$	$\mathbf{1}$	Theorem 8.6.7
	$y \,\mathrm{C}\, z$	none	16.1
$x * y * x * z$	$x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$	$\mathbf{1}$	Theorem 8.6.7
	$y \,\mathrm{C}\, z$	none	16.2
$x * y * z * x$	$x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$	$\mathbf{1}$	Theorem 8.6.7
	$y \,\mathrm{C}\, z$	none	16.3
$y * x * x * z$	$x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$	$\mathbf{1}$	Theorem 8.6.7
	$y \,\mathrm{C}\, z$	none	16.4
$z * x * y * x$	$x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$	$\mathbf{1}$	Theorem 8.6.7
	$y \,\mathrm{C}\, z$	none	16.5
$z * y * x * x$	$x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$	$\mathbf{1}$	Theorem 8.6.7
	$y \,\mathrm{C}\, z$	none	16.6

Table 8.25: Results for the lower commutator.

Remark 8.6.13
Due to the commutativity of the commutators, the results of (xxyz) (Argument 16.1) and (zyxx) (Argument 16.6) and those of (xyxz) (Argument 16.2) and (zxyx) (Argument 16.5) will be similar. We will see that their results are symmetric in terms of the way of putting parentheses, see Figure 8.33, Arguments 16.5 and 16.6.

Argument 16.1

$y \, \mathrm{C} \, z$	Lower commutator in D_{16}
$x * x * y * z$	$x := a,\ y := c\ z := d$
P1:	$\big((a*a)*c\big)*d = \mathbf{1}$
P2:	$\big(a*(a*c)\big)*d = \mathbf{1}$
P3:	$(a*a)*(c*d) = \mathbf{1}$
P4:	$a*\big((a*c)*d\big) = b$
P5:	$a*\big(a*(c*d)\big) = \mathbf{1}$

Table 8.26: Counterexample for Argument 16.1.

$$\begin{aligned} \mathrm{P1}: \ \big((x*x)*y\big)*z &= (1*y)*z = \mathbf{1} \\ \mathrm{P2}: \ \big(x*(x*y)\big)*z &= \mathbf{1} \\ \mathrm{P3}: \ (x*x)*(y*z) &= \mathbf{1} \\ \mathrm{P5}: \ x*\big(x*(y*z)\big) &= \mathbf{1} \end{aligned}$$

Argument 16.2

$y \, \mathrm{C} \, z$	Lower commutator in L_{22}
$x * y * x * z$	$x := c,\ y := e,\ z := a$
P1:	$\big((c*e)*c\big)*a = \mathbf{1}$
P2:	$\big(c*(e*c)\big)*a = \mathbf{1}$
P3:	$(c*e)*(c*a) = c$
P4:	$c*\big((e*c)*a\big) = d$
P5:	$c*\big(e*(c*a)\big) = b$

Table 8.27: Counterexamples for Argument 16.2.

$$\begin{aligned} \mathrm{P1}: \ \big((x*y)*x\big)*z &\overset{(F)}{=} \mathbf{1} \\ \mathrm{P2}: \ \big(x*(y*x)\big)*z &\overset{(F)}{=} \mathbf{1} \end{aligned}$$

Argument 16.3

$y \, \mathrm{C} \, z$	Lower commutator in D_{16}	Lower commutator in L_{22}
$x * y * z * x$	$x := a,\ y := c\ z := d$	$x := c,\ y := e,\ z := a$
P1:	$\big((a * c) * d\big) * a = b$	$\big((c * e) * a\big) * c = \mathbf{1}$
P2:	$\big(a * (c * d)\big) * a = \mathbf{1}$	$\big(c * (e * a)\big) * c = \mathbf{1}$
P3:	$(a * c) * (d * a) = \mathbf{1}$	$(c * e) * (a * c) = c$
P4:	$a * \big((c * d) * a\big) = \mathbf{1}$	$c * \big((e * a) * c\big) = \mathbf{1}$
P5:	$a * \big(c * (d * a)\big) = \mathbf{1}$	$c * \big(e * (a * c)\big) = b$

Table 8.28: Counterexamples for Argument 16.3.

$$\text{P2}: \quad \big(x * (y * z)\big) * x \overset{\text{(F)}}{=} \mathbf{1}$$
$$\text{P4}: \quad x * \big((y * z) * x\big) \overset{\text{(F)}}{=} \mathbf{1}$$

Argument 16.4

$y \, \mathrm{C} \, z$	Lower commutator in L_{22}
$y * x * x * z$	$x := a,\ y := c\ z := d$
P1:	$\big((c * a) * a\big) * d = 1$
P2:	$\big(c * (a * a)\big) * d = 1$
P3:	$(c * a) * (a * d) = a$
P4:	$c * \big((a * a) * d\big) = 1$
P5:	$c * \big(a * (a * d)\big) = 1$

Table 8.29: Counterexample for Argument 16.4.

$$\text{P1}: \quad \big((y * x) * x\big) * z \overset{\text{(R)}}{=} \mathbf{1}$$
$$\text{P2}: \quad \big(y * (x * x)\big) * z = \mathbf{1}$$
$$\text{P4}: \quad y * \big((x * x) * z\big) \overset{\text{(L)}}{=} \mathbf{1}$$
$$\text{P5}: \quad y * \big(x * (x * z)\big) \overset{\text{(L)}}{=} \mathbf{1}$$

Argument 16.5

The commutators are commutative, the results of Argument 16.5 (`zxyx`) are symmetric to the results of Argument 16.2 (`xyxz`) (Remark 8.6.13), see Figure 8.33.

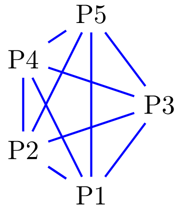
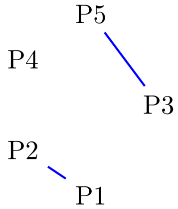
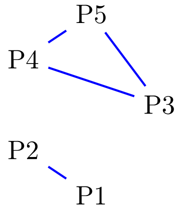
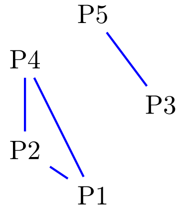
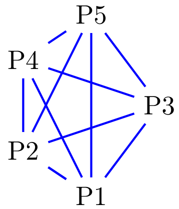
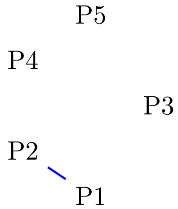
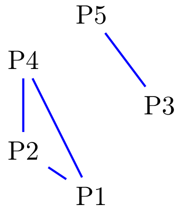
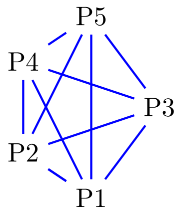
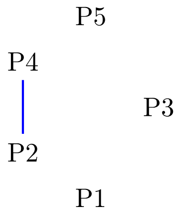
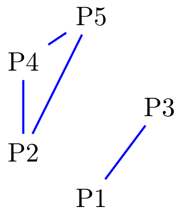
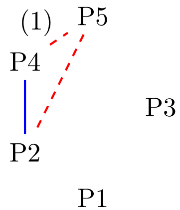
Argument 16.6

Also the results of Argument 16.6 (`zyxx`) are symmetric to the results of Argument 16.1 (`xxyz`) (Remark 8.6.13), see Figure 8.33.

On the next pages we made graphical representations of the different constellations, and the kind of fulfilling the associativity–like equations for the three representative operations, the swapped projection, the Sasaki projection and the lower commutator.

The vertices represent the way of putting parentheses. If two or more vertices are connected by an edge then they have equal results, if edges are not connected then these kinds of putting the parentheses do not have equal results. For associative operations, the representation would be a complete graph. The second column summarises the conditions for which the constellation fulfils all associativity equations.

The tables represent the minimal conditions, e.g. if $x\,\mathrm{C}\,y$ is sufficient for fulfilling some equations, then the case $(\mathrm{C}x)$ will not be mentioned.

Constellation	**Beran's number 23**			
xxyz	All associativity equations are fulfilled if (Cx) or (Cz) 	In general, no commuting elements assumed, or if $y \,\mathrm{C}\, z$ 	$x \,\mathrm{C}\, y$ or (Cy) 	$x \,\mathrm{C}\, z$ 
xyxz	All associativity equations are fulfilled if (Cy) or (Cz) 	In general, no commuting elements assumed, also if $x \,\mathrm{C}\, z$ or $y \,\mathrm{C}\, z$ 	$x \,\mathrm{C}\, y$ or (Cx) 	
xyzx	All associativity equations are fulfilled if $x \,\mathrm{C}\, z$ 	In general, no commuting elements assumed, also if $x \,\mathrm{C}\, y$ 	(Cy) 	$y \,\mathrm{C}\, z$  (1) we do not know

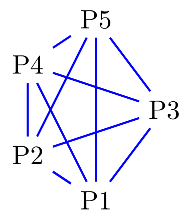
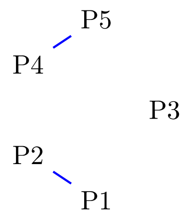
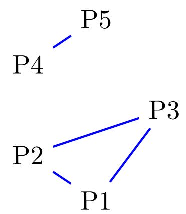
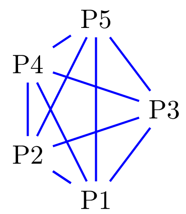
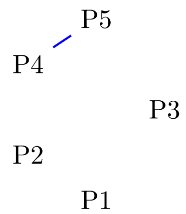
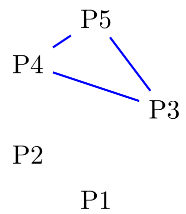
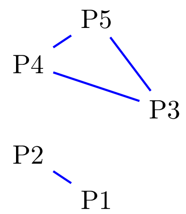
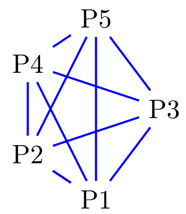
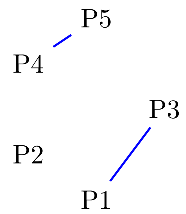
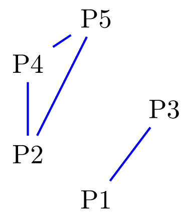
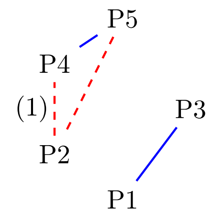
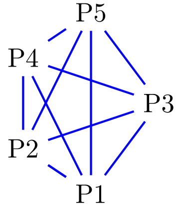
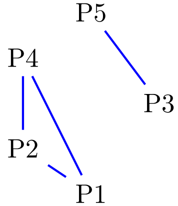
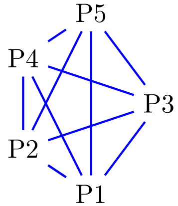
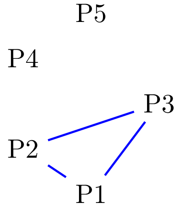
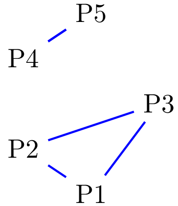
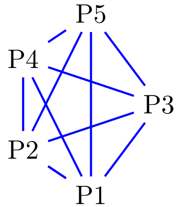
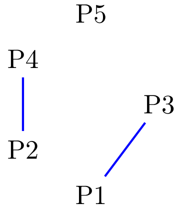
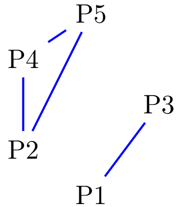
Constellation	Beran's number 23			
yxxz	All associativity equations are fulfilled if (Cy) or (Cz) 	In general, no commuting elements assumed, also if y C z 	x C y or x C z or (Cx) 	
zxyx	All associativity equations are fulfilled if x C z 	In general, no commuting elements assumed, also if y C z 	x C y 	(Cy) 
zyxx	All associativity equations are fulfilled if (Cx) or (Cz) 	In general, no commuting elements assumed also if x C z 	x C y or (Cy) 	y C z  (1) we do not know

Table 8.30: Graphical overview of all Arguments of the swapped projection.

Constellation	**Beran's number 18**		
`xxyz`	All associativity equations are fulfilled if $x \,\mathrm{C}\, y$, or $(\mathrm{C}z)$, or $x \leqslant z$, or $y \leqslant z$, or $z \leqslant y$ 	In general, no commuting elements assumed or $x \,\mathrm{C}\, z$, or $y \,\mathrm{C}\, z$, or $z \leqslant x$ 	
`xyxz`	All associativity equations are fulfilled if $x \,\mathrm{C}\, y$ or $x \leqslant z$ or $y \leqslant z$ or $(\mathrm{C}z)$ 	In general, no commuting elements assumed, also if $x \,\mathrm{C}\, z$, or $y \,\mathrm{C}\, z$, or $z \leqslant y$ 	$z \leqslant x$ 
`xyzx`	All associativity equations are fulfilled if $x \,\mathrm{C}\, y$, $x \leqslant z$, or $(\mathrm{C}z)$, or $y \leqslant z$, or $z \leqslant y$ 	In general, no commuting elements assumed or $x \,\mathrm{C}\, z$ 	$z \leqslant x$ or $y \,\mathrm{C}\, z$ 

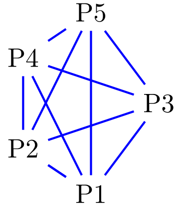
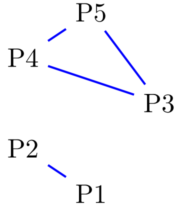
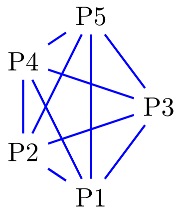
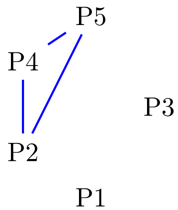
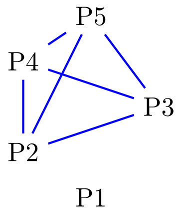
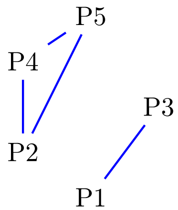
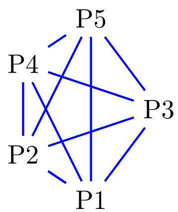
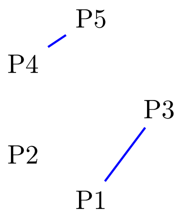
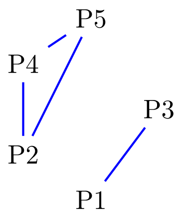
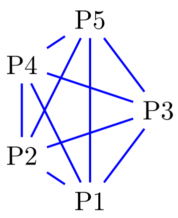
Constellation	Beran's number 18			
yxxz	All associativity equations are fulfilled if $x \,\mathrm{C}\, y$, or $(\mathrm{C}z)$, or $x \leqslant z$, or $z \leqslant x$, or $y \leqslant z$ 	In general, no commuting elements assumed, or $x \,\mathrm{C}\, z$, or $y \,\mathrm{C}\, z$, or $z \leqslant y$ 		
zxyx	All associativity equations are fulfilled if $x \,\mathrm{C}\, z$, or $x \leqslant y$, or $y \leqslant x$, or $(\mathrm{C}y)$, or $z \leqslant y$ 	In general, no commuting elements assumed 	$x \,\mathrm{C}\, y$ 	$y \,\mathrm{C}\, z$ or $y \leqslant z$ 
zyxx	All associativity equations are fulfilled if $y \,\mathrm{C}\, z$, or $x \leqslant y$, or $y \leqslant x$, or $(\mathrm{C}x)$, or $z \leqslant x$ 	In general, no commuting elements assumed, also if $x \,\mathrm{C}\, z$ or $x \leqslant z$ 	$x \,\mathrm{C}\, y$ 	

Table 8.31: Graphical overview of all Arguments of the Sasaki projection.

Beran's number 16



All associativity equations are fulfilled if $x \,\mathrm{C}\, y$ or $x \,\mathrm{C}\, z$
for all constellations `xxyz`, `xyxz`, `xyzx`, `yxxz`, `zxyx` and `zyxx`

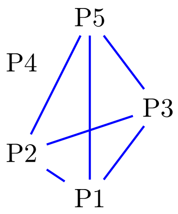
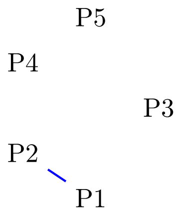
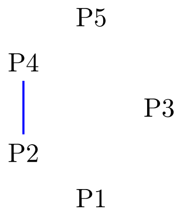
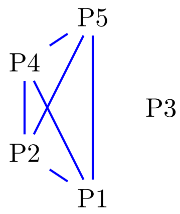
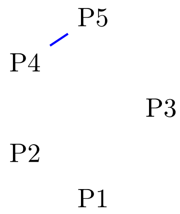
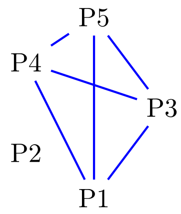
xxyz	xyxz	xyzx	yxxz	zxyx	zyxx
$y \,\mathrm{C}\, z$	$y \,\mathrm{C}\, z$	$y \,\mathrm{C}\, z$	$y \,\mathrm{C}\, z$	$y \,\mathrm{C}\, z$	$y \,\mathrm{C}\, z$
					

Table 8.33: Graphical overview of all Arguments of the lower commutator

8.7 Other weaker laws of associativity involving the orthocomplement

In this section we look at other weaker laws of associativity that involve the complement and other non–decreasing operations for which they are satisfied. Introducing the orthocomplementation, we may modify the previously studied identities in which some variable appear repeatedly. In particular, identities (L), (R), (F) can be modified to the following:

$$x * (x' * y) = (x * x') * y \tag{L'}$$
$$(y * x) * x' = y * (x * x') \tag{R'}$$
$$(x * y) * x' = x * (y * x') \tag{F'}$$

We first ask which nonassociative operations in orthomodular lattices satisfy these identities. These are also listed in Table 6.3. The equations (L'), (R') and (F') correspond to the columns (6.3), (6.5) and (6.7) respectively. We then ask which of them give a unique output for the following set of constellations regardless of the order in which the terms appear:

$$x * x' * y * z \tag{xx'yz}$$
$$x * y * x' * z \tag{xyx'z}$$
$$x * y * z * x' \tag{xyzx'}$$
$$y * x * x' * z \tag{yxx'z}$$
$$z * x * y * x' \tag{zxyx'}$$
$$z * y * x * x' \tag{zyxx'}$$

Theorem 8.7.1
Let L be an orthomodular lattice and let $*$ be an operation with the Beran's numbers in $\{16, 81\}$. If $x, y, z \in L$ such that x commutes with either y or z then each of the expressions (xx'yz), (xyx'z), (xyzx'), (yxx'z), (zxyx'), (zyxx') has a unique output regardless of the order in which the terms appear.

Proof:
The proof progresses analogously to the proof of Theorem 8.6.7. *q.e.d.*

The same questions can be posed also for other operations, but the chance of getting reasonably strong positive results is weak. We select only the following results.

Theorem 8.7.2
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 54, $x * y = (x \vee y) \wedge \big(y' \vee (x \wedge y)\big)$. If $x, y, z \in L$ such that x and y commute then each of

the following expressions has a unique output regardless of the order in which the terms appear:

$$x * y * z * x'$$
$$x * z * x' * y$$

Remark 8.7.3
The constellation $x * z * x' * y$ (xzx'y) is the same as $x * y * x' * z$ (xyx'z); we interchanged variables y, z only to use the same assumption $x \mathrm{C} y$ in both cases.

Proof:
Notice that $*$ acts as the left projection on commuting elements. Thus $x * y = x$. Also note that x commutes with $z * x'$. Further, from computation with only two variables, $x * w * x' = x$. These facts admit the following simplifications:

$$\begin{array}{lclcl} (x * y) * (z * x') & \overset{(1')}{=} & x * (z * x') & \overset{(2')}{=} & x \\ ((x * y) * z) * x' & \overset{(1')}{=} & (x * z) * x' & \overset{(2')}{=} & x \\ x * (y * z) * x' & \overset{(2')}{=} & x\,. & & \end{array} \qquad \begin{array}{lll} (1'): & x * y & = x \\ (2'): & x * w * x' & = x, \end{array}$$

For the remaining case of $x * \big(y * (z * x')\big)$, note that since x commutes with both y and $z * x'$, x also commutes with $y * (z * x')$. Hence, since $*$ acts as the left projection on commuting elements,

$$x * \big(y * (z * x')\big) = x\,.$$

Likewise, for the second constellation it follows that

$$\begin{array}{lclcl} (x * z) * (x' * y) & \overset{(1')}{=} & (x * z) * x' & \overset{(2')}{=} & x \\ (x * z * x') * y & \overset{(2')}{=} & x * y & \overset{(1')}{=} & x \\ x * \big(z * (x' * y)\big) & = & x * (z * x') & \overset{(2')}{=} & x\,. \end{array} \qquad \begin{array}{lll} (1'): & x * y & = x \\ (2'): & x * * x' & = x \end{array}$$

For the last case of $x * \big((z * x') * y\big)$, note that since x commutes with both y and $z * x'$, x also commutes with $(z * x') * y$. Hence, since $*$ acts as the left projection on commuting elements,

$$x * \big((z * x') * y\big) = x\,.$$

q.e.d.

Similarly to Corollary 8.6.3, we may prove the following:

Corollary 8.7.4
Let L be an orthomodular lattice and let $\circledast$ be the operation with Beran's number 71, $x \circledast y = (x \vee y) \wedge \big(x' \vee (x \wedge y)\big)$. If $x, y, z \in L$ such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms appear:

$$x \circledast z \circledast y \circledast x'$$
$$y \circledast x \circledast z \circledast x'$$

8.8 Conclusions

We studied six Moufang–like identities, (xxyz), (xyxz), (xyxz), (zxyx), (zyxx) and (xxyz), under the eight operations which fulfil the left (L), the right (R) and thus also the flexible (F) identities. The six associative operations are not considered, they fulfil these identities anyway. Four arguments have five possible ways of putting parenthesises and thus ten associativity equations to check, see Section 8.2. All eight operations summarised in Table 8.1 fulfil at least one of these ten associativity equations under certain conditions.

The swapped projection and the Sasaki projection even fulfil all associativity equations under the conditions summarised in Table 8.34. The upper and lower commutators have for all constellations the same conditions, i.e. for the commutators the conditions $x \mathrm{C} y$ or $x \mathrm{C} z$ are sufficient to fulfil the associativity–like identities for all constellations.

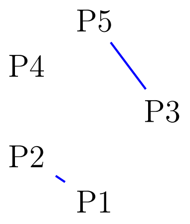
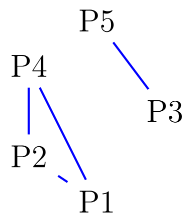
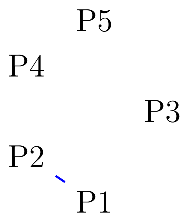
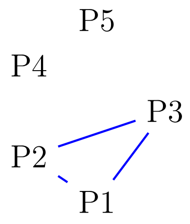
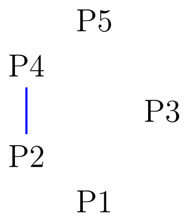
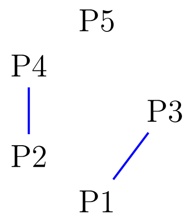
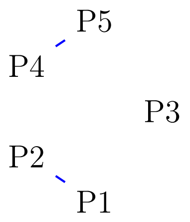
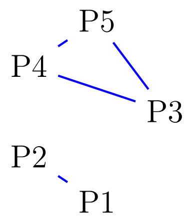
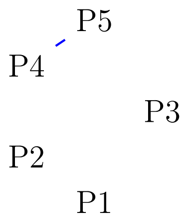
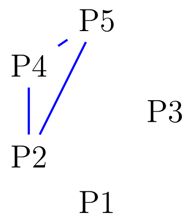
	Constellation	(Cx)	(Cy)	(Cz)	$x \mathrm{C} y$	$x \mathrm{C} z$	$y \mathrm{C} z$
Swapped projection	`xxyz`	$\circledast$		$\circledast$			
	`xyxz`		$\circledast$	$\circledast$			
	`xyzx`	$\circledast$		$\circledast$		$\circledast$	
	`yxxz`		$\circledast$	$\circledast$			
	`zxyx`	$\circledast$		$\circledast$		$\circledast$	
	`zyxx`	$\circledast$		$\circledast$			
Sasaki projection	`xxyz`	$\circledast$	$\circledast$	$\circledast$	$\circledast$		
	`xyxz`	$\circledast$	$\circledast$	$\circledast$	$\circledast$		
	`xyzx`	$\circledast$	$\circledast$	$\circledast$	$\circledast$		
	`yxxz`	$\circledast$	$\circledast$	$\circledast$	$\circledast$		
	`zxyx`	$\circledast$	$\circledast$	$\circledast$		$\circledast$	
	`zyxx`	$\circledast$	$\circledast$	$\circledast$			$\circledast$

Table 8.34: Sufficient conditions for the swapped projection and the Sasaki projection.

In Table 8.34 we summarised the sufficient conditions for the six constellations and for both operations, the swapped projection and the Sasaki projection. The swapped projection possesses the fewest cases (sufficient conditions) in which all associativity equations hold.

The Sasaki projection fulfils all associativity equations if at least one argument commutes with the other two. The Sasaki projection, in some constellations the condition $x \mathrm{C} y$, or $x \mathrm{C} z$, or $y \mathrm{C} z$ do not suffice, in these cases the arguments x and y, x and z, y and z respectively, have additionally to be comparable.

Moreover, in Table 8.35 we compared the graphs of the general cases, no commuting arguments are supposed, we see that the Sasaki projection fulfils more equalities as the swapped projections for the same constellation. The Sasaki projection fulfils exactly one equation more, except the constellation (zyxx), which fulfil exactly the same equations as the swapped projection.

Constellation	Swapped projection	Sasaki projection
`xxyz`		
`xyxz`		
`xyzx`		
`yxxz`		
`zxyx`		

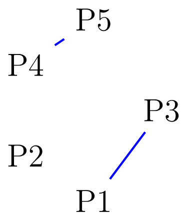
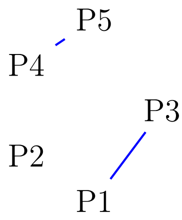
Constellation	Swapped projection	Sasaki projection
zyxx		

Table 8.35: General graphs, no commuting arguments are assumed, for the swapped projection and the Sasaki projection.

Remark 8.8.1
All our counterexamples were found in the single orthomodular lattice L_{22} of Figure 6.3. Although D_{16} is not a sublattice of L_{22}, the counterexamples using it could have also been chosen from L_{22}, too. This experience suggests a conjecture that the free orthomodular lattice with three free generators belongs to the variety generated by L_{22}. (The free orthomodular lattice with two free generators belongs to the variety generated by MO_2; L_{22} does not belong to it.) This conjecture is not true. All the orthomodular lattices used here, MO_2, D_{16}, and L_{22}, admit order–determining sets of two–valued measures (see [51]). It is known that such orthomodular lattices form a variety [41]. The lattice of subspaces of the three–dimensional real Hilbert space is an orthomodular lattice with three generators which does not admit any two–valued measure [19]. Thus the free orthomodular lattice with three free generators belongs to a variety strictly larger than that generated by MO_2, D_{16} and L_{22}.

The variety generated by the free orthomodular lattice on three free generators is the variety of all orthomodular lattices [25]. More precisely: The variety of orthomodular lattices is not generated by any set S of orthomodular lattices with a finite upper bound on the lengths of their chains. This is best seen by Jónsson's Lemma[3], since the subdirectly irreducibles in the variety generated by S will be homomorphic images of subalgebras of ultraproducts of members of S. They will have the same upper bound on the length of their chains, and it is easy to construct subdirectly irreducible orthomodular lattices that have no such finite upper bound (the horizontal sum of two infinite Boolean algebras), this follows from results in [24].

Other weaker laws of associativity If we compare the three columns `xx’y`, `xy’y` and `xyx’` with the other three columns `xxy`, `xyy` und `xyx` of Table 6.3, then the first which strikes is that there are only two nonassociative operations which satisfy the three equations (L’), (R’), and (F’), namely the two commutators, B16 and B81. We remark also that there are less operations which satisfy one or two equations as those which satisfy (L), (R), or (F).

[3]By Jónsson's Lemma, the variety $V(K)$ generated by a finite lattice has only finitely many subvarieties. This led to the conjecture that, conversely, if a lattice variety has only finitely many subvarieties, then it is generated by a finite lattice.

We notice that the analogue of the second part of Theorem 8.3.1 does not hold; some operations fulfil exactly two equations, namely B23, B38, B54 and B71, they are each other's analogues.

Because $x \underline{\mathrm{com}}\, y = x \underline{\mathrm{com}}\, y'$ holds, we get the same results with or without orthocomplements; they are summarized in Theorem 8.7.1 and Table 8.25 where the second x is replaced by x'.

Chapter 9

Conclusions and a glance on future work

Keeping the word problem in mind and having the ambition to contribute to find a way to transform elements of an orthomodular lattice to their normal form, we studied the associativity, monotonicity and alternating property in orthomodular lattices. Principally we study these properties on the 96 orthomodular operations of the free orthomodular lattice on two free generators, $\mathrm{F}(x, y)$.

The construction of $\mathrm{F}(x, y)$, being isomorphic to the direct product of the Boolean algebra $\mathbf{2^4}$ and the orthomodular lattice MO_2, has the effect that each of the sixteen Boolean operations has six orthomodular counterparts. Half of these sixteen Boolean operations are not associative, thus each of their counterparts does not fulfil the associativity identity. Hence 48 orthomodular lattice operations are excluded from the outset.

The computer program by M. Hyčko [33] allows to obtain first negative results in $\mathrm{F}(x, y)$. The remaining results have to be considered further. We examined several orthomodular lattices to find counterexamples, nevertheless, they could be all found in the lattice L_{22}.

Among the 48 possible operations we found six associative operations, the least and greatest element, the left and right projections and the meet and join operations. The least and greatest element are uninteresting for our purpose, as are the left and right projections. The meet and join are associative by definition. The left and right projections are rather unsurprising to be associative, their results only depend on the order of the arguments.

The meet and join are the only operations which are commutative as well (except the zero and the one, but they are insignificant). This means that only the meet and join are candidates over which other operations distribute. Therefore the second operation, for distributivity, has to be non–decreasing.

Further, we studied the chance that orthomodular lattice operations could have some properties related to weaker forms of associativity. Notably alternative associative characteristics. Although in the free orthomodular lattice on two free generators some operations fulfil the three alternative identities, it was in general not difficult to find counterexamples in other orthomodular lattices. Nevertheless we obtained positive results, which are summarised in the appendix.

The presence of commuting elements is crucial, for the associative as well as for distributive laws. We found operations fulfilling the associativity identity under the constraint of the presence of commuting elements. Only four nonassociative operations fulfil the associativity identity if one pair of arguments commutes, the Sasaki projection and its dual fulfil the associativity identity if $x\mathrm{C}y$ and the skew join and meet if $y\mathrm{C}z$. Others need two pairs of commuting arguments. Twelve orthomodular lattice operations fulfil the associativity identity only if all three arguments commute.

Another remarkable matter is that we met the same operations several times; the six associative operations are monotone in both variables and are trivially alternative. The Sasaki projection (B18) and its dual with Beran's number 28, fulfil the three identities (L), (R) and (F), they are monotone in the second argument and they are also idempotent operations. The skew join and its dual, Beran's number 44 and 34 respectively, fulfil the three identities (L), (R) and (F), the skew join is monotone in the first argument, the skew meet is monotone in the second argument and both are idempotent operations, too.

Also the operations with Beran's number 23, the swapped projection, and its dual with Beran's number 38, often appear. They fulfil the three identities (L), (R) and (F). The swapped projection is monotone in the second argument, its dual is monotone in the first argument. Both are idempotent.

We proved new properties of the Sasaki projection, as well as its dual or the skew operations. These give a chance to develop new algebraic methods based on these operations instead of (only) the lattice operations.

A deep research of these last expressions could be promising, particularly in connection with embedding. It is known that nonassociative algebras can be embedded in division rings [49]. Does there exist a similar approach for (orthomodular) lattices?

Our aim to develop useful tools to transform orthomodular expressions to their normal form didn't carry out completely; a result is the awareness that the traditional way of finding canonical forms does not lead to success in orthomodular lattices. The commuting property among arguments plays a crucial role.

Future work

Orthomodular lattices are studied as event structures of quantum logic. The lack of distributivity makes computations in orthomodular lattices difficult and it is an open question whether the word problem is solvable for them. As an alternative to the use of the lattice–theoretical meet and join, other operations were considered; Sasaki operations seem to be the most promising.

Related expressions to the Sasaki projection are its dual, the operation with Beran's number 28:

$$x \vee (x' \wedge y)\,,$$

the skew meet (B34) and skew join (B44) which differ from the Sasaki projection and its dual only by the order of arguments. Orthomodular lattices with one of these operations form an alternative algebra. Interesting is also an other related operation, the Sasaki arrow, see Section 5, in physics this operation is better known as the Sasaki hook. The Sasaki arrow does not form an orthomodular lattice to an alternative algebra but it is the right adjoint of the Sasaki projection.

More fundamental properties of the Sasaki projection will be handled in a future joint paper.

Further, we regret the open problem for which we did not find a satisfactory solution. The Arguments 23.3 and 23.6 (see Section 8.6.3) are related to each other and need a more fundamental study with profound knowledge of the variety generated by the lattice L_{22} and the one generated by $\mathrm{L}(\mathbb{R}^3)$.

Mathematicians are like Frenchmen: whatever you say to them they translate into their own language and forthwith it is something entirely different.

Johann Wolfgang von Goethe (1749 – 1832)

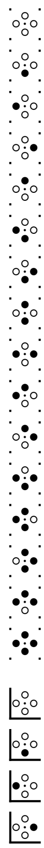
Die Mathematiker sind eine Art Franzosen: redet man zu ihnen, so übersetzen sie es in ihre Sprache, und dann ist es alsobald ganz etwas anderes.

Johann Wolfgang von Goethe (1749 – 1832)

Appendix

Beran's numbers and their expressions

1	$\mathbf{0}$
2	$x \wedge y$
3	$x \wedge y'$
4	$x' \wedge y$
5	$x' \wedge y'$
6	$(x \wedge y) \vee (x \wedge y')$
7	$(x \wedge y) \vee (x' \wedge y)$
8	$(x \wedge y) \vee (x' \wedge y')$
9	$(x \wedge y') \vee (x' \wedge y)$
10	$(x' \wedge y') \vee (x \wedge y')$
11	$(x' \wedge y') \vee (x' \wedge y)$
12	$(x \wedge y) \vee (x \wedge y') \vee (x' \wedge y)$
13	$(x \wedge y) \vee (x \wedge y') \vee (x' \wedge y')$
14	$(x \wedge y) \vee (x' \wedge y) \vee (x' \wedge y')$
15	$(x \wedge y') \vee (x' \wedge y) \vee (x' \wedge y')$
16	$(x \wedge y) \vee (x \wedge y') \vee (x' \wedge y) \vee (x' \wedge y')$
17	$x \wedge (x' \vee y) \wedge (x' \vee y')$
18	$x \wedge (x' \vee y)$
19	$x \wedge (x' \vee y')$
20	$(x' \wedge y) \vee \big(x \wedge (x' \vee y) \wedge (x' \vee y)\big)$



21	$(x' \wedge y') \vee \big(x \wedge (x' \vee y) \wedge (x' \vee y')\big)$	
22	x	
23	$(x' \vee y) \wedge \big(x \vee (x' \wedge y)\big)$	
24	$(x' \vee y) \wedge \big(x \vee (x' \wedge y')\big)$	
25	$(x' \vee y') \wedge \big(x \vee (x' \wedge y)\big)$	
26	$(x' \vee y') \wedge \big(x \vee (x' \wedge y')\big)$	
27	$(x' \vee y') \wedge (x' \vee y) \wedge \big(x \vee (x' \wedge y) \vee (x' \wedge y')\big)$	
28	$x \vee (x' \wedge y)$	
29	$x \vee (x' \wedge y')$	
30	$(x' \vee y) \wedge \big(x \vee (x' \wedge y) \vee (x' \wedge y')\big)$	
31	$(x' \vee y') \wedge \big(x \vee (x' \wedge y) \vee (x' \wedge y')\big)$	
32	$x \vee (x' \wedge y) \vee (x' \wedge y')$	
33	$y \wedge (x \vee y') \wedge (x' \vee y')$	
34	$y \wedge (x \vee y')$	
35	$(x \wedge y') \vee \big(y \wedge (x \vee y') \wedge (x' \vee y')\big)$	
36	$y \wedge (x' \vee y')$	
37	$(x' \wedge y') \vee \big(y \wedge (x \vee y') \wedge (x' \vee y')\big)$	
38	$(x \vee y') \wedge \big(y \vee (x \wedge y')\big)$	
39	y	
40	$(x \vee y') \wedge \big(y \vee (x' \wedge y')\big)$	
41	$(x' \vee y') \wedge \big(y \vee (x \wedge y')\big)$	
42	$(x' \vee y') \wedge (x \vee y') \wedge \big(y \vee (x \wedge y') \vee (x' \wedge y')\big)$	
43	$(x' \vee y') \wedge \big(y \vee (x' \wedge y')\big)$	
44	$y \vee (x \wedge y')$	
45	$(x \vee y') \wedge \big(y \vee (x \wedge y') \vee (x' \wedge y')\big)$	
46	$y \vee (x' \wedge y')$	
47	$(x' \vee y') \wedge \big(y \vee (x \wedge y') \vee (x' \wedge y')\big)$	
48	$y \vee (x \wedge y') \vee (x' \wedge y')$	
49	$y' \wedge (x \vee y) \wedge (x' \vee y)$	
50	$(x \wedge y) \vee \big(y' \wedge (x \vee y) \wedge (x' \vee y)\big)$	
51	$y' \wedge (x \vee y)$	
52	$(x' \wedge y) \vee \big(y' \wedge (x \vee y) \wedge (x' \vee y)\big)$	
53	$y' \wedge (x' \vee y)$	
54	$(x \vee y) \wedge \big(y' \vee (x \wedge y)\big)$	
55	$(x \vee y) \wedge (x' \vee y) \wedge \big(y' \vee (x \wedge y) \vee (x' \wedge y)\big)$	
56	$(x' \vee y) \wedge \big(y' \vee (x \wedge y)\big)$	
57	$(x \vee y) \wedge \big(y' \vee (x' \wedge y)\big)$	
58	y'	
59	$(x' \vee y) \wedge \big(y' \vee (x' \wedge y)\big)$	
60	$(x \vee y) \wedge \big(y' \vee (x \wedge y) \vee (x' \wedge y)\big)$	

61	$y' \vee (x \wedge y)$
62	$(x' \vee y) \wedge \big(y' \vee (x \wedge y) \vee (x' \wedge y)\big)$
63	$y' \vee (x' \wedge y)$
64	$y' \vee (x \wedge y) \vee (x' \wedge y)$
65	$x' \wedge (x \vee y) \wedge (x \vee y')$
66	$(x \wedge y) \vee \big(x' \wedge (x \vee y) \wedge (x \vee y')\big)$
67	$(x \wedge y') \vee \big(x' \wedge (x \vee y) \wedge (x \vee y')\big)$
68	$x' \wedge (x \vee y)$
69	$x' \wedge (x \vee y')$
70	$(x \vee y) \wedge (x \vee y') \wedge \big(x' \vee (x \wedge y) \vee (x \wedge y')\big)$
71	$(x \vee y) \wedge \big(x' \vee (x \wedge y)\big)$
72	$(x \vee y') \wedge \big(x' \vee (x \wedge y)\big)$
73	$(x \vee y) \wedge \big(x' \vee (x \wedge y')\big)$
74	$(x \vee y') \wedge \big(x' \vee (x \wedge y')\big)$
75	x'
76	$(x \vee y) \wedge \big(x' \vee (x \wedge y) \vee (x \wedge y')\big)$
77	$(x \vee y') \wedge \big(x' \vee (x \wedge y) \vee (x \wedge y')\big)$
78	$x' \vee (x \wedge y)$
79	$x' \vee (x \wedge y')$
80	$x' \vee (x \wedge y) \vee (x \wedge y')$
81	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y) \wedge (x' \vee y')$
82	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y)$
83	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y')$
84	$(x \vee y) \wedge (x' \vee y) \wedge (x' \vee y')$
85	$(x \vee y') \wedge (x' \vee y) \wedge (x' \vee y')$
86	$(x \vee y) \wedge (x \vee y')$
87	$(x \vee y) \wedge (x' \vee y)$
88	$(x \vee y') \wedge (x' \vee y)$
89	$(x \vee y) \wedge (x' \vee y')$
90	$(x \vee y') \wedge (x' \vee y')$
91	$(x' \vee y) \wedge (x' \vee y')$
92	$x \vee y$
93	$x \vee y'$
94	$x' \vee y$
95	$x' \vee y'$
96	$\mathbf{1}$

In the sequel, we denote by $\mathrm{B}n(x, y)$, the binary orthomodular operation in x and y with the Beran's number n, $(1 \leqslant n \leqslant 96)$.

Associative and conditionally associative operations

$(x * y) * z = x * (y * z)$		No commuting arguments
$B1(x, y)$	$=$	0
$B2(x, y)$	$=$	$x \wedge y$
$B22(x, y)$	$=$	x
$B39(x, y)$	$=$	y
$B92(x, y)$	$=$	$x \vee y$
$B96(x, y)$	$=$	1

$(x * y) * z = x * (y * z)$		x commutes with y
$B18(x, y)$	$=$	$x \wedge (x' \vee y)$
$B28(x, y)$	$=$	$x \vee (x' \wedge y)$

$(x * y) * z = x * (y * z)$		y commutes with z
$B34(x, y)$	$=$	$y \wedge (x \vee y')$
$B44(x, y)$	$=$	$y \vee (x \wedge y')$

$(x * y) * z = x * (y * z)$		$(x \,\mathrm{C}\, y$ and $x \,\mathrm{C}\, z)$ or $(y \,\mathrm{C}\, x$ and $y \,\mathrm{C}\, z)$ or $(z \,\mathrm{C}\, x$ and $z \,\mathrm{C}\, y)$
$B12(x, y)$	$=$	$(x \wedge y) \vee (x \wedge y') \vee (x' \wedge y)$
$B16(x, y)$	$=$	$(x \wedge y) \vee (x \wedge y') \vee (x' \wedge y) \vee (x' \wedge y')$
$B17(x, y)$	$=$	$x \wedge (x' \vee y) \wedge (x' \vee y')$
$B18(x, y)$	$=$	$x \wedge (x' \vee y)$
$B28(x, y)$	$=$	$x \vee (x' \wedge y)$
$B32(x, y)$	$=$	$x \vee (x' \wedge y) \vee (x' \wedge y')$
$B33(x, y)$	$=$	$y \wedge (x \vee y') \wedge (x' \vee y')$
$B34(x, y)$	$=$	$y \wedge (x \vee y')$
$B44(x, y)$	$=$	$y \vee (x \wedge y')$
$B48(x, y)$	$=$	$y \vee (x \wedge y') \vee (x' \wedge y')$
$B49(x, y)$	$=$	$y' \wedge (x \vee y) \wedge (x' \vee y)$
$B50(x, y)$	$=$	$(x \wedge y) \vee \big(y' \wedge (x \vee y) \wedge (x' \vee y)\big)$
$B60(x, y)$	$=$	$(x \vee y) \wedge \big(y' \vee (x \wedge y) \vee (x' \wedge y)\big)$
$B64(x, y)$	$=$	$y' \vee (x \wedge y) \vee (x' \wedge y)$
$B65(x, y)$	$=$	$x' \wedge (x \vee y) \wedge (x \vee y')$
$B66(x, y)$	$=$	$(x \wedge y) \vee \big(x' \wedge (x \vee y) \wedge (x \vee y')\big)$
$B76(x, y)$	$=$	$(x \vee y) \wedge \big(x' \vee (x \wedge y) \vee (x \wedge y')\big)$

$B80(x,y)$	$=$	$x' \vee (x \wedge y) \vee (x \wedge y')$
$B81(x,y)$	$=$	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y) \wedge (x' \vee y')$
$B82(x,y)$	$=$	$(x \vee y) \wedge (x \vee y') \wedge (x' \vee y)$

$(x * y) * z = x * (y * z)$		$(x \,\mathrm{C}\, y,\ x \,\mathrm{C}\, z)$ or $(z \,\mathrm{C}\, x,\ z \,\mathrm{C}\, y)$
$B6(x,y)$	$=$	$(x \wedge y) \vee (x \wedge y')$
$B7(x,y)$	$=$	$(x \wedge y) \vee (x' \wedge y)$
$B23(x,y)$	$=$	$(x' \vee y) \wedge \big(x \vee (x' \wedge y)\big)$
$B38(x,y)$	$=$	$(x \vee y') \wedge \big(y \vee (x \wedge y')\big)$
$B54(x,y)$	$=$	$(x \vee y) \wedge \big(y' \vee (x \wedge y)\big)$
$B55(x,y)$	$=$	$(x \vee y) \wedge (x' \vee y) \wedge \big(y' \vee (x \wedge y) \vee (x' \wedge y)\big)$
$B71(x,y)$	$=$	$(x \vee y) \wedge \big(x' \vee (x \wedge y)\big)$
$B72(x,y)$	$=$	$(x \vee y') \wedge \big(x' \vee (x \wedge y)\big)$
$B86(x,y)$	$=$	$(x \vee y) \wedge (x \vee y')$
$B87(x,y)$	$=$	$(x \vee y) \wedge (x' \vee y)$

$(x * y) * z = x * (y * z)$		$x \,\mathrm{C}\, y$ and $x \,\mathrm{C}\, z$
$B24(x,y)$	$=$	$(x' \vee y) \wedge \big(x \vee (x' \wedge y')\big)$
$B25(x,y)$	$=$	$(x' \vee y') \wedge \big(x \vee (x' \wedge y)\big)$
$B72(x,y)$	$=$	$(x \vee y') \wedge \big(x' \vee (x \wedge y)\big)$
$B73(x,y)$	$=$	$(x \vee y) \wedge \big(x' \vee (x \wedge y')\big)$

$(x * y) * z = x * (y * z)$		$y \,\mathrm{C}\, x$ and $y \,\mathrm{C}\, z$
$B8(x,y)$	$=$	$(x \wedge y) \vee (x' \wedge y')$
$B9(x,y)$	$=$	$(x \wedge y') \vee (x' \wedge y)$
$B88(x,y)$	$=$	$(x \vee y') \wedge (x' \vee y)$
$B89(x,y)$	$=$	$(x \vee y) \wedge (x' \vee y')$

$(x * y) * z = x * (y * z)$		$z \,\mathrm{C}\, x$ and $z \,\mathrm{C}\, y$
$B40(x,y)$	$=$	$(x \vee y') \wedge \big(y \vee (x' \wedge y')\big)$
$B41(x,y)$	$=$	$(x' \vee y') \wedge \big(y \vee (x \wedge y')\big)$
$B56(x,y)$	$=$	$(x' \vee y) \wedge \big(y' \vee (x \wedge y)\big)$
$B57(x,y)$	$=$	$(x \vee y) \wedge \big(y' \vee (x' \wedge y)\big)$

Monotonicity of orthomodular operations

For which operations do the following monotonicity implications hold?

$x \leqslant y \Rightarrow x * z \leqslant y * z$			$x \leqslant y \Rightarrow z * x \leqslant z * y$		
B1(x,y)	$=$	0	B1(x,y)	$=$	0
B2(x,y)	$=$	$x \wedge y$	B2(x,y)	$=$	$x \wedge y$
B3(x,y)	$=$	$x \wedge y'$	B4(x,y)	$=$	$x' \wedge y$
B6(x,y)	$=$	$(x \wedge y) \vee (x \wedge y')$	B7(x,y)	$=$	$(x \wedge y) \vee (x' \wedge y)$
B22(x,y)	$=$	x	B18(x,y)	$=$	$x \wedge (x' \vee y)$
B34(x,y)	$=$	$y \wedge (x \vee y')$	B22(x,y)	$=$	x
B38(x,y)	$=$	$(x \vee y') \wedge \big(y \vee (x \wedge y')\big)$	B23(x,y)	$=$	$(x' \vee y) \wedge \big(x \vee (x' \wedge y)\big)$
B39(x,y)	$=$	y	B28(x,y)	$=$	$x \vee (x' \wedge y)$
B44(x,y)	$=$	$y \vee (x \wedge y')$	B39(x,y)	$=$	y
B51(x,y)	$=$	$y' \wedge (x \vee y)$	B68(x,y)	$=$	$x' \wedge (x \vee y)$
B54(x,y)	$=$	$(x \vee y) \wedge \big(y' \vee (x \wedge y)\big)$	B71(x,y)	$=$	$(x \vee y) \wedge \big(x' \vee (x \wedge y)\big)$
B58(x,y)	$=$	y'	B75(x,y)	$=$	x'
B61(x,y)	$=$	$y' \vee (x \wedge y)$	B78(x,y)	$=$	$x' \vee (x \wedge y)$
B86(x,y)	$=$	$(x \vee y) \wedge (x \vee y')$	B87(x,y)	$=$	$(x \vee y) \wedge (x' \vee y)$
B92(x,y)	$=$	$x \vee y$	B92(x,y)	$=$	$x \vee y$
B93(x,y)	$=$	$x \vee y'$	B94(x,y)	$=$	$x' \vee y$
B96(x,y)	$=$	1	B96(x,y)	$=$	1

$x \leqslant y \Rightarrow x * z \geqslant y * z$			$x \leqslant y \Rightarrow z * x \geqslant z * y$		
B1(x,y)	$=$	0	B1(x,y)	$=$	0
B4(x,y)	$=$	$x' \wedge y$	B3(x,y)	$=$	$x \wedge y'$
B5(x,y)	$=$	$x' \wedge y'$	B5(x,y)	$=$	$x' \wedge y'$
B11(x,y)	$=$	$(x' \wedge y') \vee (x' \wedge y)$	B10(x,y)	$=$	$(x' \wedge y') \vee (x \wedge y')$
B36(x,y)	$=$	$y \wedge (x' \vee y')$	B19(x,y)	$=$	$x \wedge (x' \vee y')$
B39(x,y)	$=$	y	B22(x,y)	$=$	x
B43(x,y)	$=$	$(x' \vee y') \wedge \big(y \vee (x' \wedge y')\big)$	B26(x,y)	$=$	$(x' \vee y') \wedge \big(x \vee (x' \wedge y')\big)$
B46(x,y)	$=$	$y \vee (x' \wedge y')$	B29(x,y)	$=$	$x \vee (x' \wedge y')$
B53(x,y)	$=$	$y' \wedge (x' \vee y)$	B58(x,y)	$=$	y'
B58(x,y)	$=$	y'	B69(x,y)	$=$	$x' \wedge (x \vee y')$
B59(x,y)	$=$	$(x' \vee y) \wedge \big(y' \vee (x' \wedge y)\big)$	B74(x,y)	$=$	$(x \vee y') \wedge \big(x' \vee (x \wedge y')\big)$
B63(x,y)	$=$	$y' \vee (x' \wedge y)$	B75(x,y)	$=$	x'
B75(x,y)	$=$	x'	B79(x,y)	$=$	$x' \vee (x \wedge y')$
B91(x,y)	$=$	$(x' \vee y) \wedge (x' \vee y')$	B90(x,y)	$=$	$(x \vee y') \wedge (x' \vee y')$
B94(x,y)	$=$	$x' \vee y$	B93(x,y)	$=$	$x \vee y'$
B95(x,y)	$=$	$x' \vee y'$	B95(x,y)	$=$	$x' \vee y'$
B96(x,y)	$=$	1	B96(x,y)	$=$	1

Weak–associative operations on lattices

Theorem
Let L be an orthomodular lattice and let $*$ be an operation with Beran's number in $\{18, 28\}$. Then

$$\begin{array}{lcl} (x * y * x) * z & = & (x * y) * (x * z) \\ \big(z * (x * y)\big) * x & = & z * (x * y * x) \\ \big((x * y) * z\big) * x & = & (x * y) * (z * x) \end{array}$$

for any $x, y, z \in L$.

Theorem
Let L be an orthomodular lattice and let $*$ be an operation with Beran's number in $\{34, 44\}$. Then

$$\begin{array}{lcl} z * (x * y * x) & = & (z * x) * (y * x) \\ x * \big((y * x) * z\big) & = & (x * y * x) * z \\ x * \big(z * (y * x)\big) & = & (x * z) * (y * x) \end{array}$$

for any $x, y, z \in L$.

Theorem
Let L be an orthomodular lattice and let $*$ be an operation with Beran's number in $\{18, 28\}$. If $x, y, z \in L$ such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$\begin{array}{l} x * y * x * z \\ x * y * z * x \\ y * x * z * x \\ x * x * y * z \\ x * y * y * z \\ x * y * z * z \end{array}$$

Theorem
Let L be an orthomodular lattice and let $*$ be an operation with Beran's number in $\{34, 44\}$. If $x, y, z \in L$ such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$\begin{array}{l} z * x * y * x \\ x * z * y * x \\ x * z * x * y \\ z * y * x * x \\ z * y * y * x \\ z * z * y * x \end{array}$$

Theorem
Let L be an orthomodular lattice and let $*$ be an operation with Beran's number in $\{16, 81\}$. If $x, y, z \in L$ such that x commutes with either y or z then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$\begin{array}{l} x * y * x * z \\ x * y * z * x \\ z * x * y * x \\ x * x * y * z \\ y * x * x * z \\ z * y * x * x \end{array}$$

Theorem
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 23. If $x, y, z \in L$ such that x and z commute then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$z * x * y * x$$
$$x * y * z * x$$

Theorem
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 38. If $x, y, z \in L$ such that x and z commute then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$x * y * x * z$$
$$x * z * y * x$$

Theorem
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 54, $x * y = (x \vee y) \wedge \big(y \vee (x \wedge y)\big)$. If $x, y, z \in L$ such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$x * z * x' * y$$
$$x * y * z * x'$$

Theorem
Let L be an orthomodular lattice and let $*$ be the operation with Beran's number 71, $x * y = (x \vee y) \wedge \big(x \vee (x \wedge y)\big)$. If $x, y, z \in L$ such that x and y commute then each of the following expressions has a unique output regardless of the order in which the terms are evaluated:

$$y * x * z * x'$$
$$x * z * y * x'$$

Theorem
Let x, y and z be elements of an orthomodular lattice L, $*$ is the Sasaki projection and assume $y \, \mathrm{C} \, z$, then

$$\big((z * x) * y\big) * x = (z * x) * (y * x) = (z * x) * y \, .$$

Sufficient conditions for associativity equations in Moufang–like constellations

Constellation	operation	(Cx)	(Cy)	(Cz)	$x \, \mathrm{C} \, y$	$x \, \mathrm{C} \, z$	$y \, \mathrm{C} \, z$
xxyz	B23	⊛		⊛			
xxyz	B18	⊛	⊛	⊛	⊛		
xyxz	B23		⊛	⊛			
xyxz	B18	⊛	⊛	⊛	⊛		
xyzx	B23	⊛		⊛		⊛	
xyzx	B18	⊛	⊛	⊛	⊛		
yxxz	B23		⊛	⊛			
yxxz	B18	⊛	⊛	⊛	⊛		
zxyx	B23	⊛		⊛		⊛	
zxyx	B18	⊛	⊛	⊛		⊛	
zyxx	B23	⊛		⊛			
zyxx	B18	⊛	⊛	⊛			⊛

Bibliography

[1] Baez, John C.: The Octonions. *Bulletin of the American Mathematical Society*, **39** (2) (2002), 145–205. Errata in *Bulletin of the American Mathematical Society*, **42** (2) (2005), 213.

[2] Beran, Ladislav: Über gewisse Sätze vom Foulis–Holland–Typ in Boole'schen Zwerchverbänden. *Journal für die reine und angewandte Mathematik*, **297** (1978), 214–220.

[3] Beran, Ladislav: *Orthomodular Lattices. Algebraic Approach.* Academia, Praha, 1984.

[4] Birkhoff, Garrett and von Neumann, John: The logic of quantum mechanics. *Annals of Mathematics*, **37** (4) (1936), 823–843.

[5] Boone, William W.: The word problem. *Proceedings of the National Academy of Sciences*, **44** (10) (1958), 1061–1065.

[6] Bruns, Günter: Free Ortholattices. *Canadian Journal of Mathematics*, **28** (1976), 977–985.

[7] Conway, John H. and Smith, Derek A.: *On Quaternions and Octonions. Their Geometry, Arithmetic, and Symmetry.* CRC Press, London, New-York, 2003.

[8] Dedekind, Richard: Über die drei Moduln erzeugte Dualgruppe. *Mathematische Annalen*, **53** (3) (1900), 371–403.

[9] Dehn, Max: Über unendliche diskontinuierliche Gruppen. *Mathematische Annalen*, **71** (1) (1911), 116–144.

[10] D'Hooghe, Bart and Pykacz, Jaroslaw: On some new operations on orthomodular lattices. *International Journal of Theoretical Physics*, **39** (3) (2000), 641–652.

[11] Egly, Uwe and Tompits, Hans: Gentzen-like methods in quantum logic. In *Proceedings of the Eight International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '99)*, Position Papers. Technical Report, Institute for Programming and Logics, University at Albany - SUNY, 1999.

[12] Foulis, David J.: A note on orthomodular lattices. *Portugaliae Mathematica*, **21** (1) (1962), 65–72.

[13] Freese, Ralph: Free modular lattices. *Transactions of the Mathematical Society*, **261** (1) (1980), 81–90.

[14] Freese, Ralph: Free lattice algorithms. *Order*, **3** (4) (1987), 331–344.

[15] Freese, Ralph: Finitely Presented Lattices: Canonical Forms and the Covering Relation. *Transactions of the American Mathematical Society*, **312** (2) (1989), 841–860.

[16] Gabriëls, Jeannine J.M. and Navara, Mirko: Associativity of operations on orthomodular lattices. *Mathematica Slovaca*, **62** (6) (2012), 1069-1078.

[17] Gabriëls, Jeannine J.M. and Navara, Mirko: Computer proof of monotonicity of operations on orthomodular lattices. *Information Sciences*, **236** (2013), 205–217.

[18] Gagola III, Stephen M., Gabriëls, Jeannine J.M. and Navara, Mirko: Weaker forms of associativity in orthomodular lattices. *Algebra Universalis*, **73** (3–4) (2015), 349–266.

[19] Gleason, Andrew M.: Measures on the closed subspaces of a Hilbert space. *Journal of Mathematics and Mechanics*, **6** (6) (1957), 885–893.

[20] Grätzer, George: *Lattice Theory: Foundation.* Birkhäuser Verlag, Basel, 2010.

[21] Greechie, Richard J.: On generating distributive sublattices of orthomodular lattices. *Proceedings of the American Mathematical Society*, **67** (1) (1977), 17–22.

[22] Greechie, Richard J.: An addendum to "On generating distributive sublattices of orthomodular lattices". *Proceedings of the American Mathematical Society*, **76** (2) (1979), 216–218.

[23] Harding, John: Orthomodular lattices whose McNeille completions are not orthomodular. *Order*, **8** (1) (1991), 93–103.

[24] Harding, John: Completions of othomodular lattices II. *Order*, **10** (3) 1993, 283–294.

[25] Harding, John: The free orthomodular lattice on countably many generators is a subalgebra of the free orthomodular lattice on three generators. *Algebra Universalis*, **48** (2) (2002), 171–182.

[26] Harding, J.: The source of orthomodular law. *The Handbook of Quantum Logic and Quantum Structures*, Elsevier 2007, 553–586.

[27] Haviar, Miroslav, Konôpka, Pavel, Priestley, Hilary A. and Wegener, Claudia B.: Finitely Generated Free Modular Ortholattices I. *International Journal of Theoretical Physics*, **36** (12) (1997), 2639–2660.

[28] Herrmann, Christian: On the word problem for the modular lattice with four free generators. *Mathematische Annalen*, **265** (4) (1983), 513–527.

[29] Herrmann, Christian, Micol, Florence and Roddy Michael S.: On n-distributive Modular Ortholattices. *Algebra Universalis*, **53** (2–3) (2005), 143–147.

[30] Holland Jr., Samuel S.: A Radon-Nikodym theorem in dimension lattices. *Transactions of the American Mathematical Society*, **108** (1) (1963), 66–87.

[31] Hyčko, Marek: Implications and equivalences in orthomodular lattices. *Demonstratio Mathematica* **XXXVIII** (38–4) (2005), 777-792.

[32] Hyčko, Marek and Navara, Mirko: Decidability in orthomodular lattices. *International Journal of Theoretical Physics*, **44** (22) (2005), 2239–2248. DOI: 10.1007/s10773-005-8019-x

[33] Hyčko, Marek: Computations in OML. http://www.mat.savba.sk/~hycko/oml, 2011/06/07

[34] Jordan, Pascual, von Neumann, John and Wigner, Eugene P.: On an Algebraic Generalization of the Quantum Mechanical Formalism. *Annals of Mathematics*, Second Series, **35** (1) (1934), 29–64.

[35] Jordan, Pascual: Halbgruppen von Idempotenten und nichtkommutative Verbände. *Journal für die reine und angewandte Mathematik*, **211** (1962), 136–161.

[36] Kalmbach, Gudrun: Orthomodular Lattices do not satisfy any special lattice equation. *Archiv der Mathematik* **28** (1) (1977), 7–8.

[37] Kalmbach, Gudrun: *Orthomodular Lattices*. Academic Press, London, 1983.

[38] Kröger, Henner: Zwerch-Assoziativität und verbandsähnliche Algebren. *Bayerische Akademie der Wissenschaften, mathematisch-naturwissenschaftliche Klasse*, Sonderdruck 3 aus den Sitzungsberichten 1973, 23–48.

[39] Lipshitz, Leonard M.: The Undecidability of the Word Problems for Projective Geometries and Modular Lattices. *Transactions of the American Mathematical Society*, **193** (1974), 171–180.

[40] MacLaren, M. Donald: Atomic orthocomplemented lattices. *Pacific Journal of Mathematics*, **14** (2) (1964), 597–612.

[41] Mayet, René: Varieties of orthomodular lattices related to states. *Algebra Universalis*, **20** (3) (1985), 368–396.

[42] Mayet, René and Navara, Mirko: Classes of logics representable as kernels of measures. In: *Contributions to General Algebra*, **9**, G. Pilz, (ed.) Teubner, Stuttgart, Wien, 1995, 241–248.

[43] Megill, Norman D. and Pavičić, Mladen: Orthomodular lattices and a quantum algebra. *International Journal of Theoretical Physics*, **40** (8) (2001), 1387–1410.

[44] Megill, Norman D. and Pavičić, Mladen: Equivalencies, identities, symmetric differences, and congruences in orthomodular lattices. *International Journal of Theoretical Physics*, **42** (12) (2003), 2797–2805.

[45] Megill, Norman D. and Pavičić, Mladen: Quantum implication algebras. *International Journal of Theoretical Physics*, **42** (12) (2003), 2807–2822.

[46] Meinander, Andrea: A solution of the uniform word problem for ortholattices. *Mathematical Structures in Computer Science*, **20** (04) (2010), 625–638.

[47] Navara, Mirko: On generating finite orthomodular sublattices. *Tatra Mountains Mathematical Publications*, **10** (1997), 109–117.

[48] Navara, Mirko: Constructions of quantum structures. In: D. Gabbay, D. Lehmann, K. Engesser (eds.), *Handbook of Quantum Logic*, Vol. 1, Elsevier, 2007, 335–366.

[49] Neumann, Bernhard H.: Embedding non-associative rings in division rings. *Proceedings of the London Mathematical Society*, **1** (3) (1951). 241–256.

[50] Novikov, Petr S.: On the algorithmic unsolvability of the word problem in group theory. *Proceedings of the Steklov Institute of Mathematics*, **44** (1955), 3–143. (Russian)

[51] Pták, Pavel and Pulmannová, Sylvia: *Orthomodular Structures as Quantum Logics*. Kluwer Academic Publishers, Dordrecht, 1991.

[52] Roddy, Michael S.: On the word problem for orthocomplemented modular lattices. *Canadian Journal of Mathematics*, **61** (6) (1989), 961–1004.

[53] Schafer, Richard D.: An Introduction to Nonassociative Algebras, Oxford City Press, 2010.

[54] Skolem, Thoralf: Logisch-Kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze nebst einem Theoreme über dichte Mengen. *Skrifter utgit av Videnskapsselskapet i Kristiania, I. Matematisk-naturvidenskabelig klasse*, **4** (1920), 1–36. Kristiania (Oslo): Jacob Dybwad, 1920.

[55] Svozil, Karl: *Quantum logic*. Springer-Verlag, Signapore, 1998.

[56] Tamari, Dov: *Problèmes d'associativité des monoïdes et problèmes des mots pour les groupes*, Séminaire Dubreil. Algèbre et théorie des nombres, tome 16, no 1 (1962-1963), exp. no 7, 1–29.

[57] Whitman, Philip M.: Free Lattices. *Annals of Mathematics*, **42** (1) (1941), 325–330.

[58] Whitman, Philip M.: Free Lattices II. *Annals of Mathematics*, **43** (1) (1942), 104–115.

Author's Publications

Journal papers related to the topic of the thesis

Gabriëls, Jeannine J.M. and Navara, Mirko: Associativity of operations on orthomodular lattices. *Mathematica Slovaca*, **62** 6 (2012), 1069-1078. [authorship 50%, cited once]

Gabriëls, Jeannine J.M. and Navara, Mirko: Computer proof of monotonicity of operations on orthomodular lattices. *Information Sciences*, **236** (2013), 205–217. [authorship 50%]

Gagola III, Stephen M., Gabriëls, Jeannine J.M. and Navara, Mirko: Weaker forms of associativity in orthomodular lattices. *Algebra Universalis*, **73** (2015), 249–266. [authorship 33%]

Gabriëls, Jeannine J.M.: Conditional associativity in orthomodular lattices. Submitted. [authorship 100%]

Gabriëls, Jeannine J.M., Gagola III, Stephen M. and Navara, Mirko: Sasaki projection. Submitted. [authorship 33%]

Papers not related to the thesis

Dästner, Kaeye, Gabriëls, Jeannine, Henrich, Werner, Kausch, Thomas, Opitz, Felix: Sensor Data Fusion on the Frigate F124. Architecture and Principles. International Radar Symposium IRS 2003. Dresden, Germany, October 2003. [authorship 50%]

Conference papers related to the thesis

Gabriëls, Jeannine J.M. and Navara Mirko: Properties of OML operations and the role of computers in proofs. In: M. Kalina, O. Nánásiová, L. Valášková (eds.), Abstr. International Conference Quantum Structures 2011, Kočovce, Slovakia 2011. [Authorship 50 %]

Gabriëls, Jeannine J.M.: Shall we find normal forms on orthomodular lattices? Klement E.P., Mesiar R., Drobná E., Struk P. (eds.), Abstr. Eleventh International Conference on Fuzzy Set Theory and Applications, FSTA 2012, Liptovský Ján, Slovakia, 2012. [Authorship 100 %]

Gabriëls, Jeannine J.M. and Navara, Mirko: Searching normal forms for orthomodular lattices, Abstr. Eleventh Biennal IQSA Meeting Quantum Structures, Cagliari 2012. [authorship 50%]

Gagola III, Stephen M., Gabriëls, Jeannine J.M. and Navara, Mirko: Identities valid in orthomodular lattices. Abstr. Twelfth Biennal IQSA Meeting Quantum Structures, Olomouc 2014. [authorship 33%]

Gabriëls, Jeannine J.M., Gagola III, Stephen M. and Navara, Mirko: New properties of Sasaki projections. Topology, Algebra, and Categories in Logic 2015, Ischia (Italy), 21 – 26 June 2015, *Booklet of abstracts*, 107 – 109. [authorship 33%]

Talks related to the thesis

QS 2011 Presentation at the International Conference Quantum Structures, Kočovce, Slovakia, 16 – 20 May 2011.

FSTA 2012 Presentation at the Eleventh International Conference on Fuzzy Set Theory and Applications, FSTA, Liptovský Ján, Slovakia, 30 January – 3 February 2012.

IQSA 2012 Presentation at the Eleventh Biennal IQSA Meeting Quantum Structures, Cagliari, Italy, 23 – 27 July 2012.

TACL 2015 Presentation at Topology, Algebra, and Categories in Logic 2015, Ischia (Italy), 21 – 26 June 2015.

Index