



university of
 groningen

faculty of science
and engineering

mathematics and applied
mathematics

Underlying mathematical structures in Aristotelian Diagrams

Bachelor's Project Mathematics

August 2018

Student: C. Nijhof

First supervisor: Prof. dr. G.R. Renardel de Lavalette

Second assessor: Prof. dr. J. Top

Abstract

In a recent paper, Lorenz Demey presents an algorithm to compute the maximal Boolean complexity of a family of Aristotelian diagrams. However, the underlying mathematical notions, involving partial orders extended with an involutive negation function, are hardly worked out. The purpose of this thesis is to provide a critical analysis of Demey's paper and related work from a mathematical viewpoint. In a clear and understandable way, the theory of Aristotelian diagrams is connected to the mathematical notions of Hasse diagrams and Boolean algebras. Demey's algorithm is rewritten and applied in several examples of Aristotelian families.

Contents

1	Introduction	4
2	Preliminaries	5
2.1	Modal logic	5
2.2	Aristotelian diagrams	6
2.3	Partial orders	8
2.4	Boolean algebras	9
3	Aristotelian structures	12
4	Maximal Boolean Complexity of an Aristotelian family	15
4.1	Computing the maximal Boolean complexity	15
4.2	Examples of partitions	16
4.3	Algorithm in a mathematical setting	17
4.4	Time complexity of the algorithm	18
4.5	Examples of computing the maximal Boolean complexity	19
5	Conclusion	23

1 Introduction

Logicians make use of several kinds of diagrams for a variety of purposes, such as obtaining new results and communicating their findings more effectively. Using diagrams is a way to visualize properties of logical systems. Roughly seen, the logical diagrams can be divided into two categories. The first is of diagrams that visualize formulas for some given logical system. The second is of diagrams that visualize the *relations* between such formulas. Aristotelian diagrams fall under the latter. These diagrams visualize the relations between a set of formulas from some given logical system. The ongoing investigations concerning Aristotelian diagrams are mostly about determining their properties. Logicians develop algorithms to obtain new properties of these diagrams. Aristotelian diagrams are not only used in logic, but also in fields such as cognitive science [16], linguistics [12], philosophy [15], neuroscience [1], law [13] and computer science: Aristotelian diagrams are used by computer scientists to study various ways of knowledge representation in, e.g. rough set theory [4], possibility theory [9], formal argumentation theory [2] and multiple-criterion decision-making [10]. The most used and therefore most important Aristotelian diagram is the so-called *square of opposition*, which will be shown later on in the thesis. Recently other, more complex diagrams have also been investigated, such as hexagons, cubes, etc. Despite the many researches on Aristotelian diagrams that have been done, the underlying mathematics of these diagrams have not been made explicit yet. Working out these mathematics could make it less complicated to obtain a better understanding of Aristotelian diagrams and its properties.

The overall purpose of this thesis is to provide a critical analysis of Demey's article on maximal Boolean complexity and the related work from a mathematical viewpoint. Therefore, the research question that this thesis will answer is: what are the mathematical structures underlying logical diagrams and how can these be used to compute their properties?

Before we answer this question you will find an extensive introduction to Aristotelian diagrams, including an introduction to several forms of modal logic. Some mathematical notions will be defined, such as Hasse diagrams and Boolean algebras. The connection between these notions and Aristotelian diagrams will be explained and analyzed. After this we can take a closer look at some properties of Aristotelian diagrams, including the maximal Boolean complexity, introduced by Demey in [6].

2 Preliminaries

This section serves two purposes. The first is, in view of the possible situation where some of the readers might not be well accustomed with the logic that is required to understand the mathematics in this thesis, to acquaint general readers with fundamental background of modal logic. Secondly, this section prepares the readers for the mathematics in later chapters through some introductory chapters about Aristotelian and Hasse diagrams.

2.1 Modal logic

In this section the concept of *modal logic* will be introduced. For more information on modal logic, see Priests book on logic [17].

Modal logic studies reasoning that involves the notions 'necessarily' and 'possibly'. We use a box (\Box) for 'it is necessarily the case that' and a diamond (\Diamond) for 'it is possibly the case that'. For example, let p be the formula 'I will wear a red shirt today'. Then $\Box p$ means *it is necessarily the case that I will wear a red shirt today* and $\Diamond p$ means *it is possibly the case that I will wear a red shirt today*. These operators can be expressed in terms of the other by use of negation:

$$\begin{aligned}\Box p &\leftrightarrow \neg \Diamond \neg p \\ \Diamond p &\leftrightarrow \neg \Box \neg p\end{aligned}$$

This shows that it is necessary that I will wear a red shirt today if and only if it is *not* possible that I will *not* wear a red shirt today, which sounds logical. Similarly, the converse says that it is possible that I will wear a red shirt today if and only if it is *not* necessary that I will *not* wear a red shirt today.

The semantics of modal logic can be formulated as follows. We have a set W containing possible worlds. Intuitively, one can think of these possible worlds as places where things may be different from the world we live in. For example, what kind of world would we live in if everyone was two inches taller, or if you would have a different hair color. Of course, our actual world is a possible world also. On such a set of possible worlds we have an *accessibility relation* R . This relation is a binary relation on W . If u and v are two possible worlds in W , then uRv means that in world u , world v is considered possible, i.e. v is *accessible* from u .

A model \mathfrak{M} in modal logic is given by the triple $\langle W, R, V \rangle$, where W is the set of possible worlds, R is a relation on W called the accessibility relation and V is a *valuation*. The valuation V maps every atomic formula to the subset of W containing all worlds in which the formula is true. We call a formula *atomic* if it contains no logical operators (negation, conjunction, etc.), i.e. an atomic formula has no strict subformulas.

We introduce a symbol for semantical validity: \models . When a sentence φ is true in some world w , we note this as $w \models \varphi$. The negated symbol $\not\models$ is used when a formula φ is not true in some world w : $w \not\models \varphi$. With this information, we can give formal definitions of the operators \Box and \Diamond .

Definition 2.1. *Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model in some logical system. Then, for some possible world $w \in W$, some atomic formula p and some formula φ ,*

$$\begin{aligned}w \models p &\quad \text{iff } w \in V(p), \\ \mathfrak{M} \models p &\quad \text{iff } w \models p \text{ for every world } w \in W, \\ w \models \neg \varphi &\quad \text{iff } w \not\models \varphi, \\ w \models (\varphi \wedge \psi) &\quad \text{iff } w \models \varphi \text{ and } w \models \psi, \\ w \models (\varphi \vee \psi) &\quad \text{iff } w \models \varphi \text{ or } w \models \psi, \\ w \models (\varphi \rightarrow \psi) &\quad \text{iff } w \not\models \varphi \text{ or } w \models \psi,\end{aligned}$$

$$\begin{aligned}
w \models \Box \varphi & \quad \text{iff for every world } u \in W, wRu \text{ implies } u \models \varphi, \\
w \models \Diamond \varphi & \quad \text{iff for some world } u \in W, it holds that } wRu \text{ and } u \models \varphi.
\end{aligned}$$

So the formula $\Box p$ being true in world u means that in all accessible worlds from u the formula p is true. The formula $\Diamond p$ being true in world u means that there is at least one world, accessible from u , where the formula p is true. So the operators show whether or not formulas in the accessible worlds are true.

We have that $w \models p$ if p is true in world w , we have $\mathfrak{M} \models p$ if p is true in *every* possible world in the model, and we have one more expression: the left-hand side of the sentence being empty ($\models \varphi$) means that $\mathfrak{M} \models p$ for all models \mathfrak{M} . In that case, the right-hand side is called a *tautology*, i.e. a formula that is always true; in any model and in any world. The symbol is used for logical entailment also, when a formula implies another: the sentence $\varphi \models \psi$ is equivalent to the sentence $\models \varphi \rightarrow \psi$. In the sentence $\not\models \varphi$ the right-hand side is a *contradiction*, i.e. a formula that is always false.

There are several logical systems in modal logic. In [11] an extensive overview is given of these logics. The different systems of modal logic have their own properties of the corresponding accessibility relation. The logic K is the most basic logic, having no extra properties. The logical system KD has a *serial* accessibility relation: from every possible world another world can be accessed. The following axiom belongs to this system: $\Box p \models \Diamond p$. Without the serial property of the accessibility relation, the formula $\Box p$ does not necessarily mean that there exists an accessible world (think about a model with a single world; then every formula starting with \Box is true). Some properties that accessibility relations may have are:

- Serial: for every $u \in W$, there exists a $v \in W$ such that uRv
- Reflexive: uRu for every $u \in W$
- Symmetric: uRv implies vRu for all $u, v \in W$
- Transitive: uRv and vRw together imply uRw for all $u, v, w \in W$
- Euclidean: uRv and uRw together imply vRw for all $u, v, w \in W$ ¹

One can prove that the serial property directly follows from reflexivity: for every $u \in W$, there exists a $v \in W$ such that uRv , namely u itself. Many properties correspond to an axiom in the logic:

- Serial: $\Box p \rightarrow \Diamond p$
- Reflexive: $\Box p \rightarrow p$
- Symmetric: $p \rightarrow \Box \Diamond p$
- Transitive: $\Box p \rightarrow \Box \Box p$
- Euclidean: $\Diamond p \rightarrow \Box \Diamond p$

Combining the different properties and thereby the different axioms give logical systems, e.g. KD (serial), KT (reflexive), KB (reflexive and symmetric), S4 (reflexive and transitive) and S5 (reflexive and Euclidean). For a given logical system S the set of atomic formulas is denoted by \mathcal{L}_S .

2.2 Aristotelian diagrams

In general, Aristotelian diagrams are diagrams that visualize the Aristotelian relations between formulas from some given logical system. We define the Aristotelian relations relative to some logical system S. This system is assumed to have the usual operators (\neg , \wedge , \vee) and a model-theoretic semantics \models_S . Now we define the Aristotelian relations. The formulas $\varphi, \psi \in \mathcal{L}_S$ are said to be

$$\begin{array}{lll}
\text{S-contradictory} & \text{iff} & \models_S \neg(\varphi \wedge \psi) \quad \text{and} \quad \models_S \varphi \vee \psi \\
\text{S-contrary} & \text{iff} & \models_S \neg(\varphi \wedge \psi) \quad \text{and} \quad \not\models_S \varphi \vee \psi
\end{array}$$

¹(note that this also implies wRv)

$$\begin{array}{ll}
\text{S-subcontrary} & \text{iff} \quad \not\models_S \neg(\varphi \wedge \psi) \quad \text{and} \quad \models_S \varphi \vee \psi \\
\text{in S-subalternation} & \text{iff} \quad \models_S \varphi \rightarrow \psi \quad \text{and} \quad \not\models_S \psi \rightarrow \varphi.
\end{array}$$

The definitions of these relations can be read as follows. Every relation has two conditions: the first tells us whether or not the two formulas can be *true* together; the second condition tells us whether or not the formulas can be *false* together. So two formulas are contradictory when they can neither be true together nor be false together, i.e. in any situation, one of them is true and the other is false. Two formulas are contrary when they cannot be true together, but they can be false together, i.e. in any situation, either one of them is true and the other is false, or they are both false. For subcontrary formulas we have the opposite: two formulas are subcontrary when they cannot be false together, i.e. in any situation, either one of them is true and the other is false, or they are both true. The last relation is different. Two formulas being in subalternation means that the first entails the second, but the second does not entail the first.

Generally, Aristotelian diagrams impose three constraints on the formulas visualized: they are contingent (may in some worlds be true and in some worlds be false), pairwise non-equivalent, and they come in contradictory pairs (when a diagram contains a formula φ , it also contains the negation $\neg\varphi$).

An example of an Aristotelian diagram is the well-known *square of opposition* in the modal logic KD. This diagram consists of four formulas: $\Box p$, $\neg\Box p$, $\Diamond p$ and $\neg\Diamond p$. As can be seen in figure 1b, the formulas can be represented as vertices and the Aristotelian relations between them as edges. More examples are shown in figure 2. These are examples of *Jacoby-Sesmat-Blanché (JSB) hexagons* in the two different logical systems KD and KT.

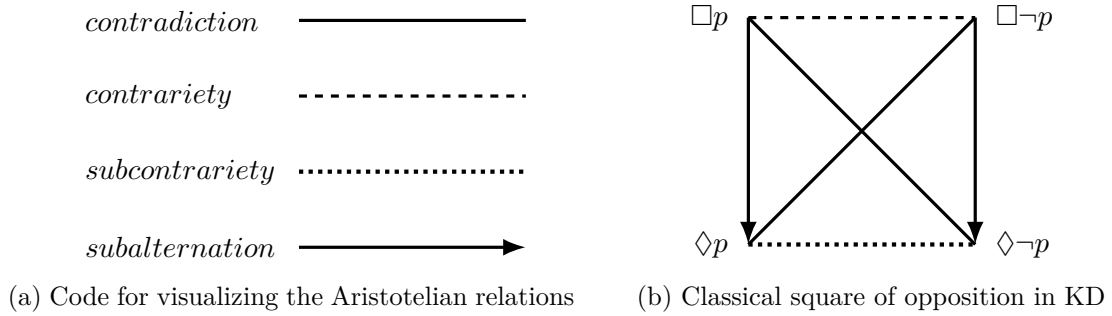


Figure 1

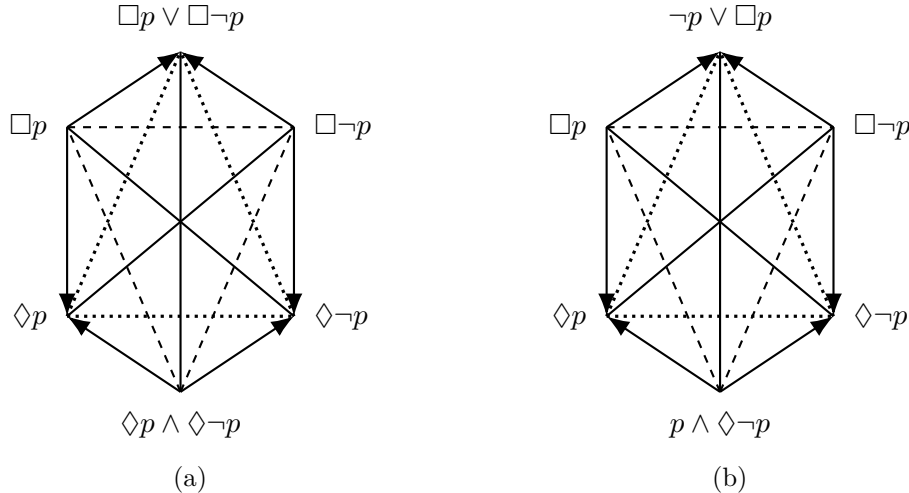


Figure 2: Examples of JSB hexagons in modal logics (a) KD and (b) KT

2.3 Partial orders

The way Aristotelian diagrams are presented above is somewhat informal. The underlying mathematics can be made more explicit. For this, we need to see that Aristotelian diagrams are closely related to Boolean algebras. Kolman, Busby and Ross gave a clear explanation of the mathematical notions needed for understanding Boolean algebras [14, Ch. 6], which is summarized in this section.

First the notion of partial order has to be introduced. A *partial order* is a relation R on a set A that has the following three properties:

- Reflexivity: $\forall a \in A, aRa$,
- Antisymmetry: $\forall a, b \in A$, if aRb and bRa , then $a = b$,
- Transitivity: $\forall a, b, c \in A$, if aRb and bRc , then aRc .

The combination of a set A with such a partial order (A, R) is called a *partially ordered set*, in short *poset*. A partial order is often noted as \leq , as will be done from here on.

Example 2.2. Let A be a collection of subsets of a set S . Then the relation \subseteq of set inclusion is a partial order on the set A . So (A, \subseteq) is a poset.

Example 2.3. The set \mathbb{Z}^+ of the positive integer numbers with the relation of divisibility (aRb if and only if a divides b) is a poset.

Two elements a, b in a poset are *comparable* if they are related: aRb or bRa . In example 2.3 two elements are comparable if one of them divides the other. The numbers 2 and 7 are not comparable, since 2 does not divide 7 and 7 does not divide 2. When every pair of elements in a poset A is comparable, then the poset is called a *linearly ordered set* or *chain*. The corresponding partial order is called a *linear order*.

A *strict* partial order is a partial order that is not reflexive, but *irreflexive*: an element in a set with a strict partial order is not related to itself, i.e. it is not true that aRa for strict partial order R . The properties of transitivity and antisymmetry still hold for strict partial orders². We denote a strict partial order by $<$.

A poset can be visualized in a directed graph (digraph). In a digraph the vertices represent the elements in the set and the edges are arrows that show when two elements are comparable and which way the relation \leq goes, so for elements a, b in the set, an arrow points from a to b if and only if $a \leq b$. In figure 3a an example is given of the digraph of the poset $\{1, 2, 4, 5, 10, 20\}$ with the relation of divisibility as its partial order.

Because of the properties of a partial order we can simplify such a digraph a lot. First, we can delete the loops, since these are implied by the property of reflexivity of the partial order. When considering a strict partial order $<$, the loops are not in the digraph. Since the partial order is always mentioned when showing a digraph, one can see from the context whether or not reflexivity is a property of the corresponding partial order and therefore whether or not it is a strict partial order or a regular partial order. Next, we can delete a lot of arrows that are a consequence of the transitivity property (if $a \leq b$ and $b \leq c$, then $a \leq c$). For example, the arrow from 1 to 4 is implied by the two arrows from 1 to 2 and from 2 to 4. Last, we agree to draw a digraph of a poset with all edges pointing upwards, so that the arrows may be drawn as simple lines. The resulting diagram is shown in figure 3b. Such a diagram is called a *Hasse diagram*, after the German mathematician Helmut Hasse (1898–1979).

Two posets (A, \leq) and (A', \leq') are called *isomorphic* if there exists an isomorphism³ between them. In that case, the Hasse diagrams of the posets look the same.

²Note that the property of antisymmetry follows from irreflexivity and transitivity.

³A function $f : A \rightarrow A'$ is called an isomorphism from (A, \leq) to (A', \leq') if it is bijective and, for any $a, b \in A$, $a \leq b$ if and only if $f(a) \leq' f(b)$

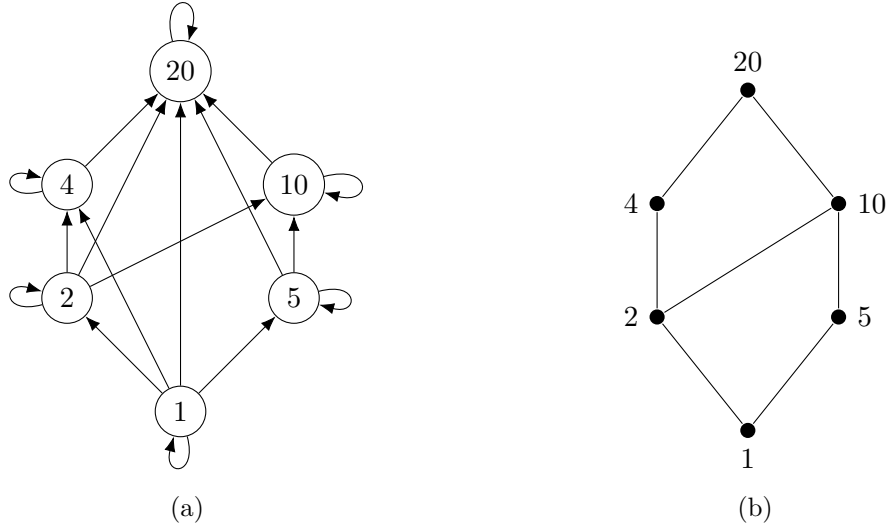


Figure 3: The directed graph and the Hasse diagram of the relation of divisibility on the set $\{1, 2, 4, 5, 10, 20\}$.

We recall the definitions for upper and lower bounds. For a subset B of a poset A an element $a \in A$ is called an *upper bound* of B if $c \leq a$ for all $c \in B$. Similarly, an element $b \in A$ is called a *lower bound* if $b \leq c$ for all $c \in B$. In addition, an element $a \in A$ is called a *least upper bound* of B if a is an upper bound of B such that $a \leq a'$ whenever a' is an upper bound of B . Similarly, an element $b \in A$ is called a *greatest lower bound* of B if b is a lower bound of B such that $b' \leq b$ whenever b' is a lower bound of B .

The last notion we need to introduce before defining a Boolean algebra is the *lattice*. A lattice is a poset (L, \leq) in which every subset $\{a, b\}$ consisting of two elements has a least upper bound and a greatest lower bound. The least upper bound of the subset is called the *join of a and b* and is denoted by $a \vee b$ and the greatest lower bound is called the *meet of a and b* and denoted by $a \wedge b$. A Boolean algebra is a lattice with the following two additional properties.

Definition 2.4. Distributive lattice A lattice (L, \leq) is distributive if the following two additional (equivalent) identities hold for all $a, b, c \in L$:

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

Definition 2.5. Complemented lattice A lattice (L, \leq) is complemented if it is bounded (containing a least element 0 and greatest element 1) and if every element $a \in L$ has a complement, i.e. an element $b \in L$ such that $a \vee b = 1$ and $a \wedge b = 0$.

As an example we consider again the poset $\{1, 2, 4, 5, 10, 20\}$ with the relation of divisibility as its partial order. This poset is distributive, since for any combination of any three elements in the set, the distributivity property holds. However, the poset is not complemented. Although it has a least element 1 and a greatest element 20, not every element has a complement: there does not exist an element b in the poset such that $2 \vee b = 20$ and $2 \wedge b = 1$. The same holds for 10.

2.4 Boolean algebras

Now that we have defined partial orders and lattices, we can define the Boolean algebra.

Definition 2.6. Boolean algebra A Boolean algebra is a complemented distributive lattice (L, \leq) with least element 0 and greatest element 1.

We can denote a Boolean algebra mathematically using a 6-tuple of all six components that form a Boolean algebra together:

$$B = \langle X, \perp, \top, \wedge, \vee, \neg \rangle.$$

We have the set X containing all elements on the Boolean algebra; the least element \perp (also called 'bottom element'); the greatest element \top (also called 'top element'); a conjunction operator \wedge ; a disjunction operator \vee ; and a negation operator \neg . Applying the conjunction operator on two or more elements in X gives an element that appears lower in the Boolean algebra. Applying the disjunction operator on two or more elements in X gives an element that appears higher in the Boolean algebra. The negation operator gives the complement of an element. The structure has to satisfy a number of axioms for the components to form a Boolean algebra together.

Example 2.7. An example of a Boolean algebra is the Hasse diagram of a *power set* with the relation of set inclusion as its partial order. The bottom element is the empty set, the top element the full set, and in between all possible subsets are given. Such a diagram of a power set is shown in figure 4a for the set $\{1, 2, 3\}$. In this example, we have that the conjunction operator is the *intersection* of sets, the disjunction operator is the *union* of sets, and the negation operator is taking the complement of a set. The least and greatest elements are the empty set and the full set.

The elements in the power set can be represented by *bitstrings*: sequences of 0's and 1's. A bitstring represents a characteristic function that corresponds with one of the subsets in the power set: for every element in the set there is one place in the bitstring, which will be a 0 if it is not in the subset, and a 1 if it is. For the set in this example we use bitstrings of length 3 since it has three elements. The resulting Boolean algebra can be seen in figure 4b.

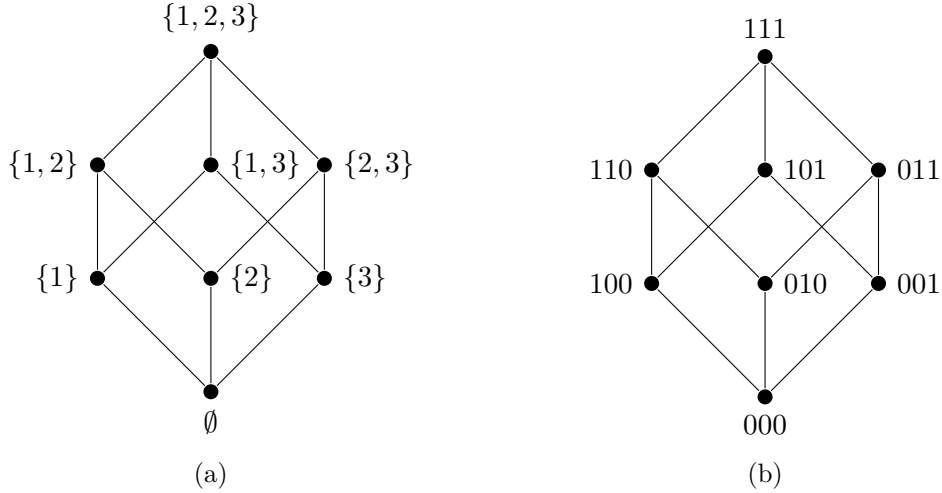


Figure 4: (a) Hasse diagram of the poset $(P(\{1, 2, 3\}), \subseteq)$ and (b) the same diagram with the subsets on the vertices represented as bitstrings.

A finite Boolean algebra is generated by its *atoms*: the first elements above the bottom element. In the previous example the singletons of the three elements that are in the complete set are the atoms: $\{1\}$, $\{2\}$ and $\{3\}$. From only these three singletons we would be able to create the entire Boolean algebra by taking unions. As Demey and Smessaert point out in [7] a way of generalizing Boolean algebras is using bitstrings on the vertices instead of specific elements. In figure 5 we see four Boolean algebras of different sizes with bitstrings on the vertices. The Boolean algebra B_n is the Boolean algebra that is generated

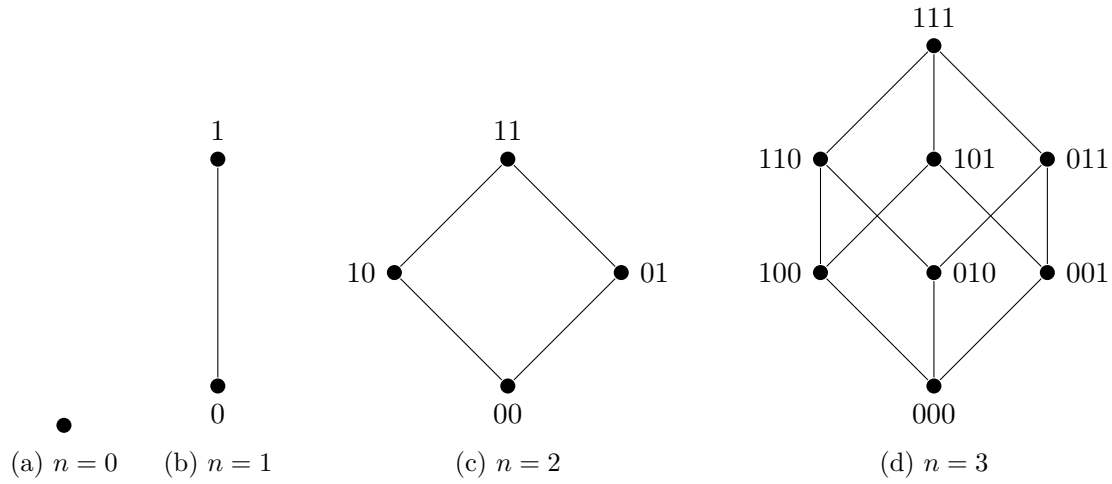


Figure 5: Hasse diagrams of the Boolean algebra B_n for $n = 0, 1, 2, 3$.

by n atoms. With this number of atoms one can compute the size of the Boolean algebra: the size is equal to 2^n . One can think of the diagram B_n as the Hasse diagram of the power set of the set $\{1, 2, \dots, n\}$, where the subsets are represented by bitstrings of length n : for every element in the set there is one fixed place in the bitstrings. Whenever the element is in the subset, there will be a 1 on its place, if not, there will be a 0. A Hasse diagram is in fact a Boolean algebra if and only if it is isomorphic to B_n for some $n \in \mathbb{N}$.

3 Aristotelian structures

Demey notes in his paper [6] that the Aristotelian relation of subalternation on a fragment F of formulas is a strict partial order and hence the set F in combination with subalternation is a poset. Furthermore, since for each formula in the fragment its negation is also in the fragment, we can view the contradictory relation as a unary and involutive⁴ function \neg on F . It is not hard to see that the four Aristotelian relations can be characterized in terms of the last relation (subalternation) and equivalence using this negation function. We use the strict partial order symbol $<$ for the strict subalternation relation. Two formulas φ and ψ are:

contradictory	iff	$\varphi \equiv \neg\psi$ (and, equivalently, $\neg\varphi \equiv \psi$),
contrary	iff	$\varphi < \neg\psi$ (and, equivalently, $\psi < \neg\varphi$),
subcontrary	iff	$\neg\varphi < \psi$ (and, equivalently, $\neg\psi < \varphi$).

Using this interdefinability of the Aristotelian relations, we can define the Aristotelian structure as a 3-tuple $\langle F, <, \neg \rangle$ that has the following properties:

- $<$ is the strict partial order of subalternation on F
- \neg is an involution on F
- $\forall \varphi, \psi \in F, (\varphi < \psi \Leftrightarrow \neg\psi < \neg\varphi)$

To connect the theory of posets and Hasse diagrams to Aristotelian diagrams, we look at a Hasse diagram of a fragment of formulas from modal logic.⁵ Every Aristotelian diagram can be turned into a Hasse diagram, showing the strict partial ordering in the fragment. The fragment of formulas is treated as a poset with logical entailment as its partial order. From the Aristotelian diagram we pick only the subalternation edges and make sure these point upwards. Figure 6a shows the Hasse diagram of a fragment of formulas in logic system S5. In figure 6b the top and bottom elements \top and \perp are added. As indicated in [5], these elements represent the *least* and *greatest* elements of the fragment. The top element is a tautology and the bottom element is a contradiction. These formulas are called *non-contingent*. A formula is *contingent* if and only if it may in some worlds be true and in some worlds be false. We see here that the diagram is in fact a Boolean algebra after adding the top and bottom elements, as it is isomorphic to the Boolean algebra B_3 .

The atoms are the strongest consistent formulas in the visualized fragment. The stronger a formula, the more other formulas it implies. The bottom element \perp is the strongest of all formulas and stronger than the atoms, but it is not *in* the fragment. The three formulas above the atoms are weaker than the atoms, since they imply less other formulas. The top element \top is the weakest.

The Boolean algebra is, as said before, completely generated by its atoms. In the case of logical formulas one can compute the entire algebra out of the three strongest formulas using only disjunctions. In figure 6b one can see that the formula $\Diamond p$ is equivalent to the disjunction of the left two atoms in the Boolean algebra:

$$\Box p \vee (\Diamond p \wedge \Diamond \neg p) \equiv_{S5} \Diamond p$$

To investigate the properties of Aristotelian diagrams, we distinguish between individual Aristotelian diagrams and families of Aristotelian diagrams. A *family of Aristotelian diagrams* (in short, *Aristotelian family*) is defined as a class C of Aristotelian diagrams that are all isomorphic to each other, and any diagram that is isomorphic to a member

⁴A function is unary if it has only one argument. A function is involutive if it is equal to its own inverse. Equivalently, applying an involution twice yields identity.

⁵In [8] the exact mathematical computation that links the Hasse diagram to the Aristotelian diagram of a fragment is given.

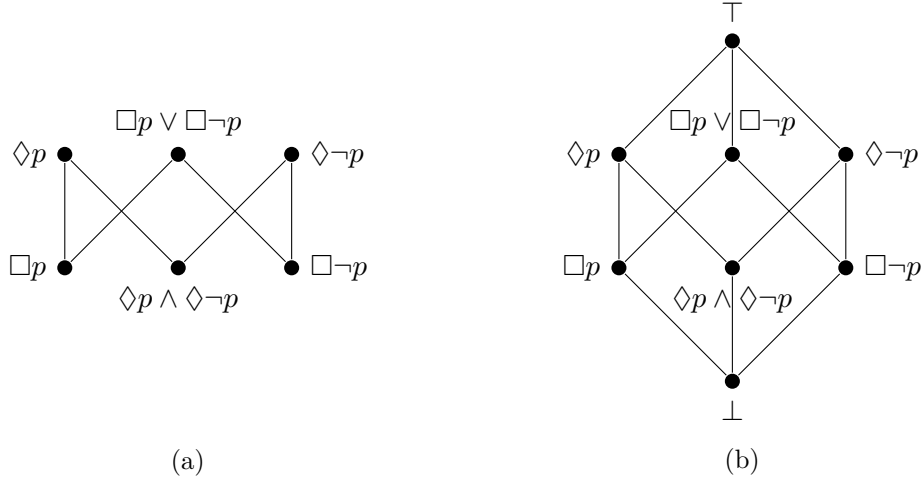


Figure 6: (a) Hasse diagram of a fragment of formulas in modal logic S5 and (b) the Boolean algebra after adding the top and bottom element

of C is a member of C itself. Using this definition, we can generalize our diagrams. In figure 7 we see two generic descriptions of Aristotelian families. Figure 7a shows us the generic description of the Aristotelian family of classical squares: instead of using specific examples of formulas on the vertices, the generic fragment $\{\varphi_1, \varphi_2, \neg\varphi_1, \neg\varphi_2\}$ is used. This fragment can be used without loss of generality because of the assumption that for every formula φ in a fragment, its negation $\neg\varphi$ is in the fragment also. Figure 7b shows the generic description of the Aristotelian family of JSB hexagons with generic fragment $\{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1, \neg\varphi_2, \neg\varphi_3\}$.

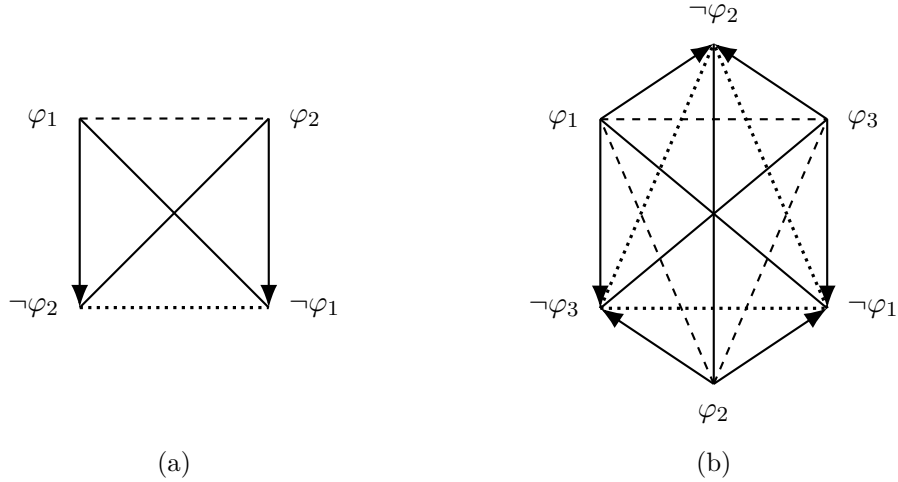


Figure 7: Generic diagrams of the Aristotelian families of (a) classical squares and (b) JSB hexagons

For both Aristotelian families one can make a Hasse diagram from the generic Aristotelian diagram. For the Aristotelian family of classical squares, we get the diagram in figure 8a. The Aristotelian family of JSB hexagons gives the Hasse diagram shown in figure 8b. The arrows representing the relation of subalternation in figure 7 are changed into lines, where the direction of the entailment is always upwards.

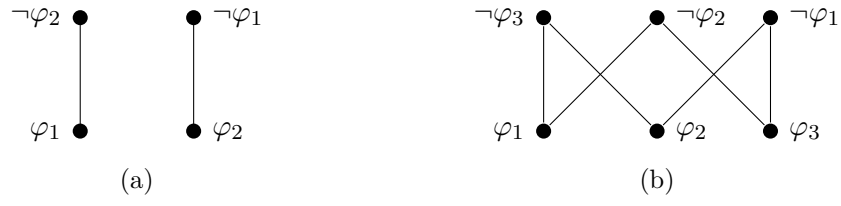


Figure 8: Hasse diagram of (a) the generic Aristotelian square (fig. 7a) and (b) the generic Aristotelian JSB hexagon (fig. 7b)

4 Maximal Boolean Complexity of an Aristotelian family

The connection between Aristotelian diagrams and Hasse diagrams enables us to investigate and compute certain properties of Aristotelian diagrams that are hard to find otherwise. Demey introduces the maximal Boolean complexity of an Aristotelian family [6]. He gives an algorithm to compute it, but the mathematics behind it is hardly made explicit. The results of computing the maximal Boolean complexity of an Aristotelian diagram can be used to classify Aristotelian diagrams systematically into Aristotelian families and Boolean subfamilies. We will try and build a similar algorithm step by step, using the mathematics behind Aristotelian diagrams, instead of using only the diagrams.

4.1 Computing the maximal Boolean complexity

The *complexity of a Boolean algebra* is equal to the number of atoms in it. So the complexity of an algebra B_n with 2^n vertices is equal to n . In this section we want to find out how to obtain the *Boolean complexity of an Aristotelian diagram*. This involves more steps than just counting the number of atoms in a Boolean algebra. From the fragment of an Aristotelian diagram, we use conjunctions to create the strongest formulas that can be made from the fragment. These conjunctions can be used as atoms to create a Boolean algebra that contains the fragment from the diagram. The complexity of this Boolean algebra, which is equal to the number of atoms just computed, is the Boolean complexity of the Aristotelian diagram. Since we are interested in properties of families of Aristotelian diagrams, we look at the *maximal Boolean complexity of an Aristotelian family*, which is defined as follows.

Definition 4.1. Maximal Boolean complexity *The maximal Boolean complexity of an Aristotelian family is equal to the complexity of the largest Boolean algebra that can be made from the Aristotelian diagrams in the family.*

To find the Boolean complexity of a given fragment F , we have to find the atoms. These atoms are equal to the conjunction of the elements in a maximal consistent upwards closed subset of $\langle F, <, \neg \rangle$. For a subset $A \subseteq F$ we have the following.

Definition 4.2. *A is consistent if there are no two formulas $\varphi, \psi \in A$ such that $\varphi < \neg\psi$ or $\varphi = \neg\psi$.*

Definition 4.3. *A is maximal consistent if adding any other formula from F to A makes A inconsistent. As a consequence, A is maximal consistent if and only if for any consistent set $B \subseteq F$ that includes A ($A \subseteq B$) we have that $B \equiv A$.*

Definition 4.4. *A is upwards closed if for formulas $\varphi, \psi \in F$ we have: if $\varphi \in A$ and $\varphi < \psi$, then $\psi \in A$ also.*

One can prove that in fact the last follows from maximal consistency of a subset: when A is a maximal consistent subset of F and for formulas $\varphi, \psi \in F$ we have $\varphi \in A$ and $\varphi < \psi$, but $\psi \notin A$, then the formula ψ can be added to A without making it inconsistent. Therefore, upwards closedness follows from maximal consistency.

The set of the conjunctions of each maximal consistent upwards closed subset of a fragment F in logical system S is called the *partition of S induced by F* . The size of the partition of a fragment is equal to the Boolean complexity of the Boolean algebra that can be created using the elements in the partition as its atoms.

4.2 Examples of partitions

We take a look at an example of a partition of an Aristotelian diagram.

Example 4.5. Figure 2a shows a JSB hexagon in the logical system KD visualizing the Aristotelian relation between the formulas in the fragment $F = \{\Box p, \Diamond p, \Diamond p \wedge \Diamond \neg p, \Diamond \neg p, \Box \neg p, \Box p \vee \Box \neg p\}$. The Hasse diagram of this fragment is shown in figure 9. The maximal KD-consistent upwards closed subsets following from this strictly partially ordered set are:

$$\begin{aligned} &\{\Box p, \Diamond p, \Box p \vee \Box \neg p\}, \\ &\{\Box \neg p, \Diamond \neg p, \Box p \vee \Box \neg p\}, \\ &\{\Diamond p, \Diamond \neg p, \Box p \vee \Box \neg p\}, \\ &\{\Diamond p, \Diamond \neg p, \Diamond p \wedge \Diamond \neg p\} \end{aligned}$$

We see in these sets that whenever a formula φ implies another formula ψ , it is never the case that only φ is in the set and ψ is not. The upwards closedness property is satisfied. Furthermore, one cannot add another formula from the fragment to either of these sets without making the set KD-inconsistent. So the maximal consistence property is also satisfied. To form the partition that follows from the fragment, we have to make the conjunctions of the found subsets:

$$\begin{aligned} \Box p \wedge \Diamond p \wedge (\Box p \vee \Box \neg p) &\equiv_{\text{KD}} \Box p \\ \Box \neg p \wedge \Diamond \neg p \wedge (\Box p \vee \Box \neg p) &\equiv_{\text{KD}} \Box \neg p \\ \Diamond p \wedge \Diamond \neg p \wedge (\Box p \vee \Box \neg p) &\equiv_{\text{KD}} \Diamond p \wedge \Diamond \neg p \\ \Diamond p \wedge \Diamond \neg p \wedge (\Diamond p \wedge \Diamond \neg p) &\equiv_{\text{KD}} \Diamond p \wedge \Diamond \neg p \end{aligned}$$

One can see that the conjunctions of the last two subsets are equivalent to each other. So we find that the partition of KD induced by F consists of three elements: $\{\Box p, \Box \neg p, \Diamond p \wedge \Diamond \neg p\}$. The Boolean complexity of this Aristotelian diagram is therefore equal to 3.

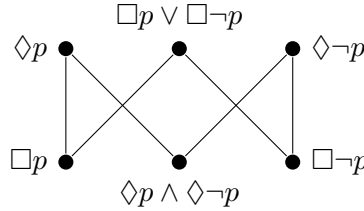


Figure 9: Hasse diagram of the KD-fragment in example 4.5

By using a generic fragment instead of a specific fragment, we can find the maximal Boolean complexity of an Aristotelian family.

Example 4.6. We take the generic fragment $\{\varphi_1, \varphi_2, \varphi_3, \neg \varphi_1, \neg \varphi_2, \neg \varphi_3\}$ with subalternation as partial order and the Hasse diagram shown in figure 8b. We have to find all maximal consistent upwards closed subsets of this fragment. The first thing we notice is that a formula φ_i and its negation can never be in one subset together, since that would make it inconsistent. Also, each subset has to contain for every φ_i either the formula φ_i itself or its negation $\neg \varphi_i$. If not, one could add this formula or its negation to the subset without making it inconsistent. Thus, from all three φ 's the formula itself or its negation has to be in each subset. Third, we have to make sure that the subset is upwards closed. For example, φ_1 implies both $\neg \varphi_2$ and $\neg \varphi_3$, so when φ_1 is in the subset, both φ_2 and φ_3 cannot be. We find the following partition:

$$\{\{\varphi_1, \neg \varphi_2, \neg \varphi_3\}, \{\neg \varphi_1, \varphi_2, \neg \varphi_3\}, \{\neg \varphi_1, \neg \varphi_2, \varphi_3\}, \{\neg \varphi_1, \neg \varphi_2, \neg \varphi_3\}\}$$

As we can see the partition has four elements. Therefore, the Aristotelian family of JSB hexagons has maximal Boolean complexity 4.

4.3 Algorithm in a mathematical setting

Demey gives an algorithm in his article [6] to compute the partition of a fragment in an Aristotelian family. In this section, his algorithm will be redesigned into a mathematical setting. By computing the partition of the generic description of an Aristotelian family, we can obtain its maximal Boolean complexity by simply counting the number of elements in the partition.

The input of the algorithm is an Aristotelian structure $\langle F, <, \neg \rangle$, where F is the fragment of formulas in the diagram, $<$ is the strict partial order of subalternation and \neg is the involutive negation function. The output of the algorithm will be the partition; the set of all maximal consistent upwards closed subsets of the fragment F .

The Boolean algebra for an Aristotelian structure has the following components. The set of all formulas in the algebra is equal to the *power set of the partition of F* , i.e. all possible subsets of the partition of F . The bottom element is the empty set, and the top element is the full partition of F . The conjunction operator is the intersection of sets \cap , the disjunction operator the union of sets \cup and the negation function is taking the complement of a set (for a set A , the complement is given by A^C). So we have the following representation of a Boolean algebra for an Aristotelian diagram \mathcal{A} :

$$B(\mathcal{A}) = \langle \mathcal{P}(\Pi(F)), \emptyset, \Pi(F), \cap, \cup, \cdot^C \rangle.$$

From this mathematical description one can see that the partition is the only thing needed to create a Boolean algebra from an Aristotelian diagram. So two things can be done after computing the partition of a diagram. First, a Boolean algebra corresponding with an Aristotelian diagram can be formed as we have seen above. Second, the maximal Boolean complexity of an Aristotelian family can be derived from the size of the partition. This can be done by taking a generic fragment of a family as input of the algorithm. The output will be the maximal partition of this family, of which the size is equal to the maximal Boolean complexity.

In section 4.2 we computed the partition of the generic description of the Aristotelian family of JSB hexagons by obtaining all subsets of the fragment that were both maximal consistent and upwards closed. Recall that a set S is maximal consistent if and only if for every formula φ in F either φ itself or the negation $\neg\varphi$ is in S . The set S is upwards closed if and only if for every two formulas φ and ψ , if φ is in S and $\varphi < \psi$, then ψ is in S also. This gives us the following mathematical description of the partition $\Pi(\mathcal{A})$ of an Aristotelian diagram $\mathcal{A} = \langle F, <, \neg \rangle$.

$$\Pi(\mathcal{A}) = \left\{ S \subseteq F \mid \begin{array}{ll} \forall \varphi \in F & (\text{either } \varphi \in S \text{ or } \neg\varphi \in S) \text{ and} \\ \forall \varphi, \psi \in F & (\varphi \in S \wedge \varphi < \psi) \Rightarrow \psi \in S \end{array} \right\}$$

Because of the property of a partition that every set contains exactly one of the elements of every pair $\{\varphi, \neg\varphi\}$ in F , we can say that the size of each set is equal to half the size of the fragment F . To simplify the algorithm, we can write the fragment F as follows without loss of generality: $F = \{\varphi_1, \varphi_2, \dots, \varphi_p, \neg\varphi_1, \neg\varphi_2, \dots, \neg\varphi_p\}$. Since F contains $2p$ elements, we have that the sets in the partition obtained from the algorithm contain p elements.

Roughly, the steps in the algorithm for obtaining the partition are as follows. We initialize the partition Π_0 as a set containing only the empty set. Then, for every $i = 1, \dots, p$, we consider the pair $\{\varphi_i, \neg\varphi_i\}$. Since every set in the partition will contain either φ_i or $\neg\varphi_i$, we take two steps for every set X in Π_{i-1} : if we can add φ_i to X without

making the set inconsistent, then we add the set $X \cup \{\varphi_i\}$ to Π_i , and similarly, if we can add $\neg\varphi_i$ to X without making the set inconsistent, we add $X \cup \{\neg\varphi_i\}$ to Π_i .

The only thing left to find out before putting together the algorithm is how to test whether a set is consistent or not. As said before, a set is consistent if and only if there are no two formulas φ, ψ in the set such that $\varphi = \neg\psi$ or $\varphi < \neg\psi$. Thus, checking that X remains consistent after adding the element φ_i means in fact checking that X does not contain an element ψ such that $\psi = \neg\varphi_i$ or $\psi < \neg\varphi_i$ (or, equivalently, $\varphi_i < \neg\psi$). Since we treat every pair $\{\varphi_i, \neg\varphi_i\}$ one by one in the algorithm, the first condition will never be the case. We only have to check the second condition: $\psi < \neg\varphi_i$. Algorithm 1 shows the pseudocode for the obtained algorithm for creating the partition of a given Aristotelian diagram.

<p>Input : Aristotelian structure $\langle F = \{\varphi_1, \dots, \varphi_p, \neg\varphi_1, \dots, \neg\varphi_p\}, <, \neg \rangle$</p> <p>Output: partition Π of the Aristotelian structure</p> <p>initialize $\Pi_0 = \{\emptyset\};$</p> <p>foreach i <i>in</i> $\{1, \dots, p\}$ do</p> <p> foreach <i>set</i> X <i>in</i> Π_{i-1} do</p> <p> if X <i>contains no element</i> ψ <i>such that</i> $\psi < \neg\varphi_i$ then</p> <p> add $X \cup \{\varphi_i\}$ to Π_i; <i>//</i> $X \cup \{\varphi_i\}$ <i>is consistent</i></p> <p> end</p> <p> if X <i>contains no element</i> ψ <i>such that</i> $\psi < \varphi_i$ then</p> <p> add $X \cup \{\neg\varphi_i\}$ to Π_i; <i>//</i> $X \cup \{\neg\varphi_i\}$ <i>is consistent</i></p> <p> end</p> <p> end</p> <p>end</p> <p>$\Pi = \Pi_p;$</p> <p>return Π</p>
--

Algorithm 1: Computing the partition of an Aristotelian diagram

This algorithm strongly relates to Demey's algorithm. Although his algorithm works with conjunctions instead of sets in the partition, the basic concept is the same: for every element in the fragment, checking whether we can add the element or its negation to the sets in the partition without making them inconsistent.

4.4 Time complexity of the algorithm

For determining the time complexity of the algorithm, we estimate the number of elementary computation steps. First, we have two *foreach*-loops and after that two *if*-loops, which both contain a number of steps to find out whether the if-statement is true or not. We know that at least one and at most two of the if-statements are true for every set X in Π_{i-1} . We also know that for $i = 1$ both if-statements are true, since the only set X in Π_0 is empty during that step. This $i = 1$ gives two steps: adding $\{\varphi_1\}$ to Π_1 and adding $\{\neg\varphi_1\}$ to Π_1 . For $i \geq 2$ every if-statement takes one step per element in the set X . Since the number of steps taken depends heavily on the specific entailment relations between the formulas, we can only compute the minimum and maximum of the number of steps, i.e. the best-case and the worst-case scenario.

The minimum number of steps is based on the assumption that for every set X in Π_{i-1} for $i \geq 2$, only one of the if-statements is satisfied, and therefore the final set Π_p only contains two sets; for every $i \geq 2$ the two sets from the first i are only expanded and no extra sets are added. In this case we take $2(1 + 2(i - 1))$ steps for every $i \geq 2$; there are two sets in Π_{i-1} , where for each we add one set to Π_i and check a condition two times for all $i - 1$ elements in the set. Including the two steps taken for the first i , we obtain the

minimum number of steps taken by the algorithm: $2 + \sum_{i=2}^p 2(1 + 2(i - 1))$, which can be simplified to

$$2 + 2(p - 1) + 4 \sum_{i=1}^{p-1} i = 2 + 2(p - 1) + 4 \cdot \frac{p(p - 1)}{2} = \mathcal{O}(p^2).$$

We find that the best-case scenario gives us a quadratic time complexity.

The worst-case scenario is based on the assumption that both if-statements are satisfied for every set X . For every i , Π_i will double in size with respect to Π_{i-1} and the sets in Π_i will grow in size with one element: for $i = 1$ we obtain two sets containing one element each, $i = 2$ gives us four sets containing two elements, $i = 3$ gives eight sets containing three elements, and so on. For every i the algorithm takes $2^{i-1} \cdot (2 + 2(i - 1))$ steps; there are 2^{i-1} sets in Π_{i-1} , where for each set we add two sets to Π_i and check a condition 2 times for all $i - 1$ elements in the set. So the maximum number of steps taken by the algorithm is:

$$\sum_{i=1}^p 2^{i-1} \cdot (2 + 2(i - 1)) = \mathcal{O}(2^p \cdot p^2) = \mathcal{O}(2^p).$$

Generally, the worst-case scenario is used for the actual time complexity of an algorithm. Therefore, we can say that our algorithm has exponential time complexity.

4.5 Examples of computing the maximal Boolean complexity

We take a look at an example using the algorithm. Ciucci, Dubois and Prade [3] give an analysis on an Aristotelian diagram that is derived from the square of opposition (figure 7a). Figure 10a shows this diagram; the *cube of opposition*. As can be seen it contains two replicas of the square, connected to each other by the Aristotelian relations of contrariety, subcontrariety and subalternation.

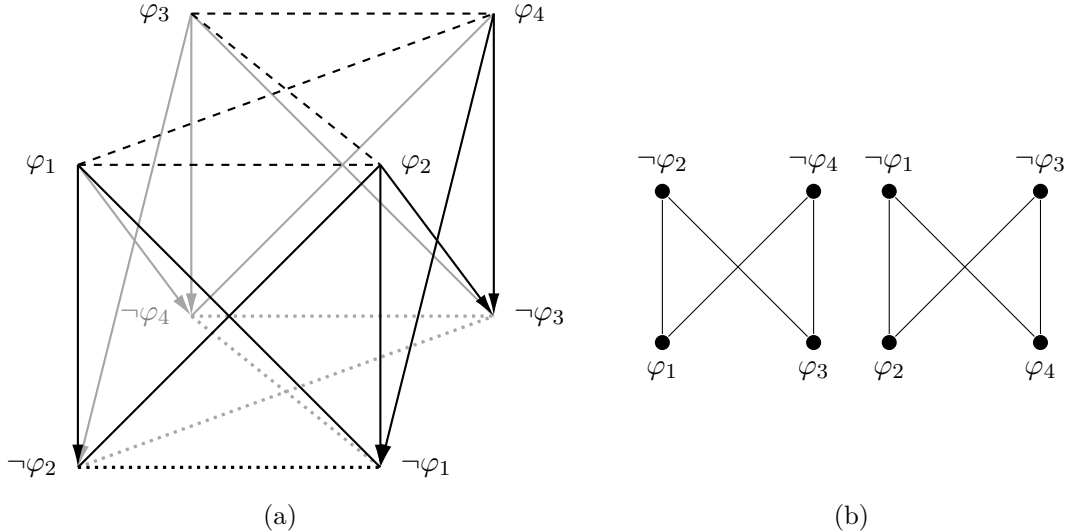


Figure 10: (a) Generic cube of opposition with (b) its Hasse diagram

Example 4.7. Figure 10b shows the Hasse diagram from the cube of opposition with subalternation as strict partial order. The comparable formulas form two separate identical Hasse diagrams. To compute the partition using the algorithm we use as input the Aristotelian structure $\langle F, <, \neg \rangle$, where F is the fragment $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_1, \neg\varphi_2, \neg\varphi_3, \neg\varphi_4\}$. We initialize the partition Π_0 as the set $\{\emptyset\}$ and then we have to take a series of steps for every $i = 1, 2, 3, 4$. We start with $i = 1$.

The set Π_0 contains only one set, the empty set. Since the empty set contains no elements at all, both if-statements are satisfied and we add $\emptyset \cup \{\varphi_1\} = \{\varphi_1\}$ and $\emptyset \cup \{\neg\varphi_1\} = \{\neg\varphi_1\}$ to Π_1 ; we find

$$\Pi_1 = \{\{\varphi_1\}, \{\neg\varphi_1\}\}.$$

Then we have $i = 2$. The set Π_1 contains two sets. For both we have to see whether adding φ_2 or $\neg\varphi_2$ will make the set inconsistent. The first set $\{\varphi_1\}$ contains the element φ_1 , which entails $\neg\varphi_2$, so the first if-statement is not satisfied. The second if-statement is satisfied, since the set does not contain an element that entails φ_2 . So we add $\{\varphi_1, \neg\varphi_2\}$ to Π_2 . The second set $\{\neg\varphi_1\}$ contains no elements that entail φ_2 as well as $\neg\varphi_2$, so we add both $\{\neg\varphi_1, \varphi_2\}$ and $\{\neg\varphi_1, \neg\varphi_2\}$ to Π_2 . This gives us the set

$$\Pi_2 = \{\{\varphi_1, \neg\varphi_2\}, \{\neg\varphi_1, \varphi_2\}, \{\neg\varphi_1, \neg\varphi_2\}\}.$$

For $i = 3$ we have to check all three sets from Π_2 . For the first set both if-statements are satisfied, so we add the two sets $\{\varphi_1, \neg\varphi_2, \varphi_3\}$ and $\{\varphi_1, \neg\varphi_2, \neg\varphi_3\}$ to Π_3 . For the second set only the second statement is satisfied, so we add $\{\neg\varphi_1, \varphi_2, \neg\varphi_3\}$ to Π_3 . The last set again satisfies both if-statements, so we add the sets $\{\neg\varphi_1, \neg\varphi_2, \varphi_3\}$ and $\{\neg\varphi_1, \neg\varphi_2, \neg\varphi_3\}$ to Π_3 . We end up with the set

$$\Pi_3 = \{\{\varphi_1, \neg\varphi_2, \varphi_3\}, \{\varphi_1, \neg\varphi_2, \neg\varphi_3\}, \{\neg\varphi_1, \varphi_2, \neg\varphi_3\}, \{\neg\varphi_1, \neg\varphi_2, \varphi_3\}, \{\neg\varphi_1, \neg\varphi_2, \neg\varphi_3\}\}.$$

At last, for $i = 4$, we check all five sets in Π_3 . For the first, second and fourth set, only the second if-statement is satisfied, so we add these three sets, each with the extra element $\neg\varphi_4$ added, to Π_4 . The third and fifth set satisfy both if-statements. So for each we add two sets to Π_4 ; one with φ_4 and one with $\neg\varphi_4$ added.

The final set we obtain is:

$$\begin{aligned} \Pi = \{ & \{\varphi_1, \neg\varphi_2, \varphi_3, \neg\varphi_4\}, \{\varphi_1, \neg\varphi_2, \neg\varphi_3, \neg\varphi_4\}, \{\neg\varphi_1, \varphi_2, \neg\varphi_3, \varphi_4\}, \{\neg\varphi_1, \varphi_2, \neg\varphi_3, \neg\varphi_4\}, \\ & \{\neg\varphi_1, \neg\varphi_2, \varphi_3, \neg\varphi_4\}, \{\neg\varphi_1, \neg\varphi_2, \neg\varphi_3, \varphi_4\}, \{\neg\varphi_1, \neg\varphi_2, \neg\varphi_3, \neg\varphi_4\} \}. \end{aligned}$$

This partition contains seven elements. Therefore, we can now say that the *family of classical cubes* has maximal Boolean complexity 7.

As we can see in this example, it takes a lot of steps to compute the maximal Boolean complexity of an Aristotelian family when using the algorithm. One can do this a lot faster by hand using the symmetry in the Hasse diagram. We show this for the same example.

Example 4.8. We look at the Hasse diagram of the generic cube of opposition in figure 10b again. We want to find the number of maximal consistent upwards closed sets in this diagram, without using the algorithm. It is not hard to see that taking the top four elements $\neg\varphi_1$, $\neg\varphi_2$, $\neg\varphi_3$ and $\neg\varphi_4$ will form one set in the partition; from each pair $\{\varphi_i, \neg\varphi_i\}$ we have one element and every element that is implied by one of the four elements is included in the set.

Now we try to replace an element in this set with an element from the bottom row. In figure 11a we see that we can replace $\neg\varphi_3$ by φ_3 in the set and still end up with a maximal consistent upwards closed set $\{\neg\varphi_1, \neg\varphi_2, \varphi_3, \neg\varphi_4\}$. Knowing this, we can use the symmetry of the diagram to come up with the set $\{\neg\varphi_1, \varphi_2, \neg\varphi_3, \neg\varphi_4\}$, where $\neg\varphi_2$ is replaced by φ_2 , as we can see in the same figure. In the same way we can replace a formula from the top of the diagram with one of the outer formulas on the bottom row of the diagram: replacing $\neg\varphi_1$ by φ_1 gives us $\{\varphi_1, \neg\varphi_2, \neg\varphi_3, \neg\varphi_4\}$. With the symmetry of the diagram we can replace $\neg\varphi_4$ by φ_4 also, see figure 11b. So by trying to replace one element and using the symmetry, we can count four new sets in the partition.

We saw that adding *one* element from the bottom row can be done for every element, and

we can also try this for *two* elements. From the left side, we can replace $\neg\varphi_1$ and $\neg\varphi_3$ by φ_1 and φ_3 without making the set inconsistent: $\{\varphi_1, \neg\varphi_2, \varphi_3, \neg\varphi_4\}$. The symmetry of the diagram gives us a similar set on the right side, as we can see in figure 11c. Any other combinations of two elements from the bottom row and two elements from the top row cannot form a consistent set for the partition, so we can count two new sets to the partition.

Adding more than two elements from the bottom row is not possible without making the set inconsistent, so we have reached all maximal consistent upwards closed subsets. We see again that the maximal Boolean complexity of the cube of opposition is 7.

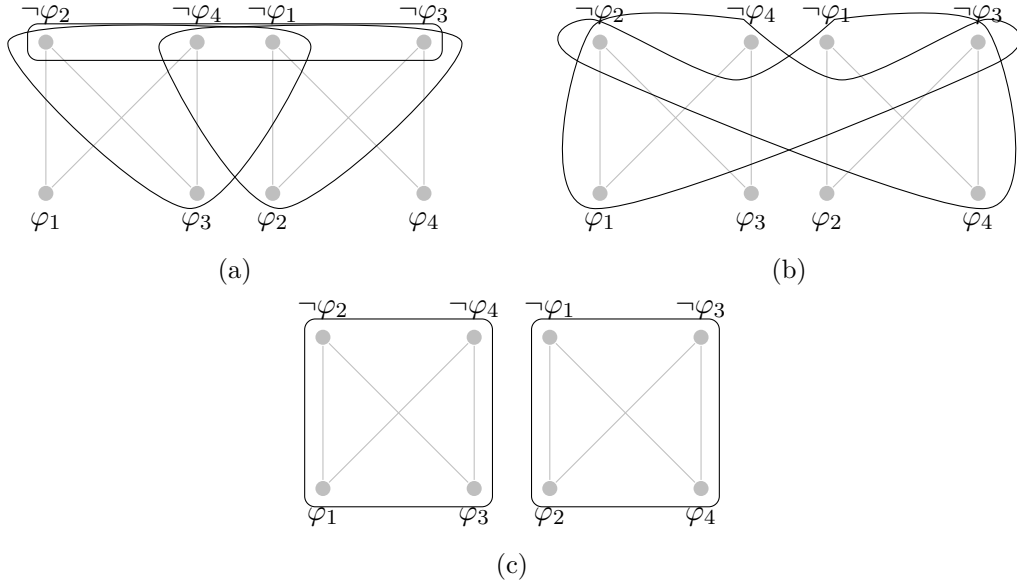


Figure 11: The seven sets in the partition of the cube of opposition

Another example that is interesting to look at is the Aristotelian rhombic dodecahedron, shown in figure 12a. The Aristotelian relation of subalternation is only visualized in this figure. The rhombic dodecahedron is an example of a 3D Aristotelian diagram. Since we have fourteen formulas, using the algorithm for this example will take again many steps, so we will work out this example using only the Hasse diagram.

Example 4.9. Figure 12b shows the Hasse diagram of the rhombic dodecahedron. To find the partition, we want to find all subsets that are upwards closed and maximal consistent. Maximal consistency is satisfied if the set contains either φ_i or $\neg\varphi_i$ for all $i = 1, \dots, 7$. We start off with the top four formulas in the Hasse diagram $\varphi_1, \varphi_2, \varphi_3$ and φ_4 . From the middle row of the diagram we can add at most three formulas for the set to remain consistent. We can do this systematically: starting off without negations, i.e. adding the formulas φ_5, φ_6 and φ_7 , and allowing one negation more in the set each time will give us all possible sets containing the four elements from the top row and three formulas from the second row. We obtain the following eight sets having zero, one, two or three negations:

$$\begin{aligned} &\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7\} \\ &\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_5, \varphi_6, \varphi_7\} \quad \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \neg\varphi_6, \varphi_7\} \quad \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \neg\varphi_7\} \\ &\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_5, \neg\varphi_6, \varphi_7\} \quad \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_5, \varphi_6, \neg\varphi_7\} \quad \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \neg\varphi_6, \neg\varphi_7\} \\ &\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_5, \neg\varphi_6, \neg\varphi_7\} \end{aligned}$$

In the Hasse diagram we can see that from the bottom row at most one formula can be in each set. For any combination of two or more formulas from the bottom row the same

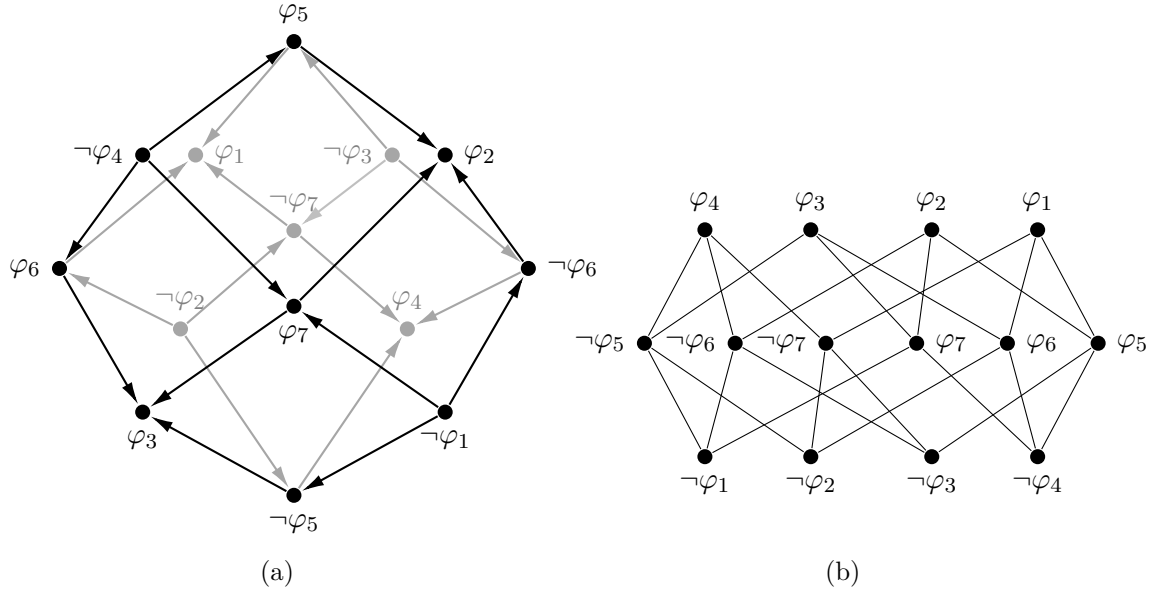


Figure 12: (a) Generic rhombic dodecahedron and its (b) Hasse diagram

situation occurs: following the upwards entailments, a formula from the middle row as well as its negation have to be in the set, which will make the set inconsistent. Including one formula from the bottom row automatically forms the rest of the set through the entailments. We obtain the following four sets, one for each element in the bottom row:

$$\begin{aligned}
 &\{\neg\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_5, \neg\varphi_6, \varphi_7\} \quad \{\varphi_1, \neg\varphi_2, \varphi_3, \varphi_4, \neg\varphi_5, \varphi_6, \neg\varphi_7\} \\
 &\{\varphi_1, \varphi_2, \neg\varphi_3, \varphi_4, \varphi_5, \neg\varphi_6, \neg\varphi_7\} \quad \{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_4, \neg\varphi_5, \varphi_6, \varphi_7\}
 \end{aligned}$$

Any other combination of the formulas will not form maximal consistent and upwards closed sets, so with these twelve sets we have found the partition of the Aristotelian rhombic dodecahedron. Therefore, the maximal Boolean complexity of this diagram is 12.

5 Conclusion

The aim of this thesis was to analyze and work out the mathematics behind Aristotelian diagrams and their properties. The research question matching this purpose was: what are the mathematical structures underlying Aristotelian diagrams and how can these be used to compute their properties? Hasse diagrams were introduced and explored to find that the essence of Aristotelian diagrams lies in the entailment relations in the visualized fragment of formulas. Combined with an involutive negation function, one can represent a diagram in a way that makes explaining and computing certain properties less complicated. Demey's article about the maximal Boolean complexity of Aristotelian families [6] gives a peek at these underlying mathematics, but in the actual computations they are hardly worked out. I introduced and defined the Aristotelian structure $\langle F, <, \neg \rangle$ and used this to redesign Demey's algorithm for computing the partition of an Aristotelian diagram (and therefore its Boolean complexity). The new algorithm is designed in a mathematical setting, which makes it easier to understand for mathematicians.

An interesting topic regarding the maximal Boolean complexity of Aristotelian diagrams is non-binary entailment relations, where more than one formulas together entail another formula. Introducing an entailment structure $\langle E, \triangleright, \neg \rangle$ satisfying certain properties, where E is a set of formulas and \triangleright is the entailment relation, would enable us to investigate its properties. How would this, for example, influence the computation of the partition and the maximal Boolean complexity? This question is left for further research.

References

- [1] Abrusci V.M., Casadio C., Medaglia T., Porcaro C. *Universal vs. particular reasoning: a study with neuroimaging techniques*. Logic Journal of the IGPL 21, 1017–1027 (2013)
- [2] Amgoud L., Prade H. *A formal concept view of formal argumentation*. In L. C. van der Gaag, editor, Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2013), LNCS 7958, pages 1–12. Springer, Berlin, 2013
- [3] Ciucci D., Dubois D., Prade H. *Structures of opposition induced by relations: The Boolean and the gradual cases* [Published online: 12 September 2015] Springer International Publishing Switzerland
- [4] Ciucci D., Dubois D., Prade H. *Oppositions in rough set theory*. In T. Li, H. S. Nguyen, G. Wang, J. Grzymala-Busse, R. Janicki, A. E. Hassanien, and H. Yu, editors, Rough Sets and Knowledge Technology, LNCS 7414, pages 504–513. Springer, Berlin, 2012
- [5] Davey B.A., Priestley H.A. *Introduction to Lattices and Order*. Cambridge Mathematical Textbooks, Cambridge University Press, 1990
- [6] Demey L. *Computing Maximal Boolean Complexity of Families of Aristotelian Diagrams*. Journal of Logic and Computation, Oxford University Press, 2018 [Available as P27 at <<http://www.logicalgeometry.org/papers-published.htm>>]
- [7] Demey L., Smessaert H. *Combinatorial bitstring semantics for arbitrary logical fragments* Journal of Philosophical logic, 47: 325, 2018.
- [8] Demey L., Smessaert H. *The Relationship between Aristotelian and Hasse Diagrams*. [In: Dwyer T., Purchase H., Delaney A. (eds) Diagrammatic Representation and Inference. Diagrams 2014. Lecture Notes in Computer Science, vol 8578]. Springer, Berlin, Heidelberg, 2014
- [9] Dubois D., Prade H. *From Blanche’s hexagonal organization of concepts to formal concept analysis and possibility theory*. Logica Universalis, 6:149–169, 2012.
- [10] Dubois D., Prade H., Rico A. *The cube of opposition – a structure underlying many knowledge representation formalisms*. In Q. Yang and M. Wooldridge, editors, Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), pages 2933–2939. AAAI Press, Palo Alto, CA, 2015.
- [11] Garson J., *"Modal Logic"*. The Stanford Encyclopedia of Philosophy, Spring 2016 Edition, Edward N. Zalta (ed.) <<https://plato.stanford.edu/entries/logic-modal/>>
- [12] Horn L.R. *A Natural History of Negation*. University of Chicago Press (1989)
- [13] Joerden J. *Logik im Recht*. Springer (2010)
- [14] Kolman B., Busby R.C., Ross S.C. *Discrete Mathematical Structures*. Pearson Prentice Hall, 2004
- [15] Massin O. *Pleasure and its contraries*. Review of Philosophy and Psychology 5, 15–40 (2014)
- [16] Mikhail J., *Universal moral grammar: theory, evidence and the future*. Trends in Cognitive Sciences 11, 143–152 (2007)
- [17] Priest G. *An Introduction to Non-Classical Logic: From If to Is*. Cambridge University Press, 2008