Computing the Maximal Boolean Complexity of Families of Aristotelian Diagrams

Lorenz Demey

Abstract

Logical geometry provides a broad framework for systematically studying the logical (and other) properties of Aristotelian diagrams. The main aim of this paper is to present and illustrate the foundations of a computational approach to logical geometry. In particular, I describe a logical problem concerning Aristotelian diagrams that is of considerable theoretical importance, viz. the task of finding the maximal Boolean complexity of a given family of Aristotelian diagrams, and I then present and discuss a simple method for automatically solving this task. This method is naturally implemented within the paradigm of logic programming (in particular, Prolog). In order to illustrate the theoretical fruitfulness of this implementation, I also show how it sheds new light on several well-known families of Aristotelian diagrams.

Keywords: Aristotelian diagram, logical geometry, bitstring semantics, Aristotelian family, Boolean subfamily, logic programming.

1 Introduction

Aristotelian diagrams are compact visual representations of the elements of some logical, lexical or conceptual field, and certain logical relations holding between them, viz. the relations of contradiction, contrariety, subcontrariety and subalternation. These diagrams (and the relations that they visualize) have a long and rich history in philosophy and logic [46], but today they are also widely used in other areas, including natural language semantics and artificial intelligence. For example, in contemporary logic research, Aristotelian diagrams are used to study various (families of) logical systems, such as modal/epistemic logic [8, 14, 29, 37, 38], fuzzy logic [32, 42, 43, 44, 63], propositional dynamic logic [14, 30] and probabilistic logic [31, 48, 49, 57]. Furthermore, in work on natural language, Aristotelian relations and the corresponding diagrams have been used in semantics [13, 36, 56, 60], pragmatics [33, 34, 35, 65, 67] and computational linguistics [39, 40, 54, 55]. Finally, Aristotelian diagrams are also used extensively by computer scientists to study various knowledge representation formalisms, including rough set theory [9, 10, 66], formal concept analysis and possibility theory [10, 22, 23], formal argumentation theory [1, 2, 3, 4], fuzzy set theory [11, 12, 24, 27], the theory of logical and analogical proportions [41, 51, 52, 53] and multiple-criterion decision-making [25, 26, 28]. The most important type of Aristotelian diagram used in these fields is, without a doubt, the so-called square of opposition, but various researchers have recently also started to use other, more complex diagrams, such as hexagons, cubes, etc. [9, 13, 22]. Dubois et al. [25] and Yao [66] make some general remarks on the heuristic usefulness of Aristotelian diagrams in the theoretical foundations of artificial intelligence, emphasizing their role in drawing comparisons across individual formalisms and in discovering new notions. In [16, 21] these remarks are further generalized to the applicability of Aristotelian diagrams in other areas.

In light of this recent trend toward more frequent and more diverse usage of Aristotelian diagrams, there has been a growing need to study these diagrams also from a more theoretical point of view. For example, after discussing a certain Aristotelian cube for specific knowledge representation purposes, Ciucci et al. go on to ask a number of general questions regarding the logical properties of this diagram, leaving many of them as questions for future research [12, Section 3.4]. The aim of *logical geometry* is to offer a broad theoretical framework in which many of these questions can systematically be addressed [15, 19, 20, 21, 59]. Furthermore, because of the rapidly growing complexity of Aristotelian diagrams,[1] it seems highly desirable to be able to answer some of these theoretical questions in an automated (computer-assisted) fashion.

The main aim of this paper is to make a modest beginning with exactly such a computational approach to logical geometry. In particular, I will describe a specific logical problem concerning Aristotelian diagrams that is of considerable theoretical importance, viz. the task of finding the maximal Boolean complexity of a given family of Aristotelian diagrams, and I will present and discuss a simple method for automatically solving this task. This method is naturally implemented within the paradigm of logic programming (in particular, Prolog). In order to illustrate the theoretical fruitfulness of this implementation, I will also show how it sheds new light on several well-known families of Aristotelian diagrams.

The paper is organized as follows. In Section 2, I briefly introduce some fundamental notions from logical geometry, focusing on those that are most relevant for the purposes of this paper. In Section 3, I then describe the task of determining the maximal Boolean complexity of a given family of Aristotelian diagrams, and discuss its theoretical importance. I will also compare this task to a number of related tasks in logical geometry, and discuss its relation to the well-known issue of logic-sensitivity in Aristotelian diagrams. Next, Section 4 informally describes a method for computing the maximal Boolean complexity of a given Aristotelian family, and Section 5 presents and discusses a Prolog implementation of this method. In Section 6, then, I will illustrate the fruitfulness of this implementation, by using it to actually compute the maximal Boolean complexity of several well-known fami-

[1] For example, it can be shown that there exist 2 families of Aristotelian squares, 5 families of hexagons, and 18 families of octagons/cubes. Determining which, and how many, families of Aristotelian diagrams exist beyond the octagons, is a matter of ongoing research.

lies of Aristotelian diagrams, and discussing the broader theoretical importance of these results. Finally, in Section 7, I summarize the paper, and offer some suggestions for further research.

2 Aristotelian Diagrams and Bitstring Semantics

The Aristotelian relations can be defined on various levels of generality and abstractness. The most general definition is formulated in terms of Boolean algebras [19, 20], but for our current purposes it will suffice to define the Aristotelian relations relative to some logical system S. Hence, consider a logical system S, which is assumed to have the usual Boolean operators, and a model-theoretic semantics $\models_{\mathsf{S}}$. The formulas $\varphi, \psi \in \mathscr{L}_{\mathsf{S}}$ are said to be

S-*contradictory*	iff	$\models_{\mathsf{S}} \neg(\varphi \wedge \psi)$	and	$\models_{\mathsf{S}} \varphi \vee \psi$,
S-*contrary*	iff	$\models_{\mathsf{S}} \neg(\varphi \wedge \psi)$	and	$\not\models_{\mathsf{S}} \varphi \vee \psi$,
S-*subcontrary*	iff	$\not\models_{\mathsf{S}} \neg(\varphi \wedge \psi)$	and	$\models_{\mathsf{S}} \varphi \vee \psi$,
in S-*subalternation*	iff	$\models_{\mathsf{S}} \varphi \to \psi$	and	$\not\models_{\mathsf{S}} \psi \to \varphi$.

These relations are abbreviated as CD_{S}, C_{S}, SC_{S} and SA_{S}, respectively. Informally, the relations CD_{S}, C_{S} and SC_{S} are defined in terms of whether the formulas can be true together and whether they can be false together,[2] whereas SA_{S} is defined in terms of truth propagation [59].

An Aristotelian diagram visualizes a fragment of formulas $\mathscr{F} \subseteq \mathscr{L}_{\mathsf{S}}$, and the Aristotelian relations holding between those formulas. Because of various historical and systematic reasons [59, Subsection 2.1], Aristotelian diagrams are usually only drawn for fragments satisfying certain conditions. In particular, throughout this paper we will assume that the formulas in $\mathscr{F}$ are S-contingent and pairwise non-S-equivalent, and that $\mathscr{F}$ itself is closed under negation.[3] More formally, an *Aristotelian diagram for* $\mathscr{F}$ *in* S is a diagram that visualizes an edge-labeled graph $\mathscr{G}$. The vertices of $\mathscr{G}$ are the formulas of $\mathscr{F}$, and the edges of $\mathscr{G}$ are labeled by the Aristotelian relations holding between those formulas, i.e. if $\varphi, \psi \in \mathscr{F}$ stand in some Aristotelian relation in S, then this is visualized according to the code in Fig. 1(a). Some examples of Aristotelian diagrams are shown in Fig. 1(b-c) and Fig. 2(a-c).

The notion of an *Aristotelian isomorphism* captures what it means for two Aristotelian diagrams to be 'essentially the same', from the perspective of their Aristotelian relations. Suppose that $\mathscr{D}$ is an Aristotelian diagram for the fragment $\mathscr{F}$ in the logical system S, while $\mathscr{D}'$ is a diagram for the fragment $\mathscr{F}'$ in the system S$'$. We say that $\mathscr{D}$ and $\mathscr{D}'$ are Aristotelian isomorphic to each other iff there exists a bijection $\gamma \colon \mathscr{F} \to \mathscr{F}'$ such that for all Aristotelian relations R and formulas

[2] The $\neg(\varphi \wedge \psi)$ part specifies whether the formulas can be true together; similarly, given the equivalence of $\varphi \vee \psi$ and $\neg(\neg\varphi \wedge \neg\psi)$, the $\varphi \vee \psi$ part specifies whether the formulas can be false together.

[3] So for all distinct $\varphi, \psi \in \mathscr{F}$, it holds that $\not\models_{\mathsf{S}} \varphi$, $\not\models_{\mathsf{S}} \neg\varphi$, $\not\models_{\mathsf{S}} \varphi \leftrightarrow \psi$, and there exists a $\varphi' \in \mathscr{F}$ such that $\models_{\mathsf{S}} \varphi' \leftrightarrow \neg\varphi$.

Figure 1: (a) Code for visualizing the Aristotelian relations, (b) classical square of opposition in KD, (c) degenerate square in classical propositional logic (CPL).

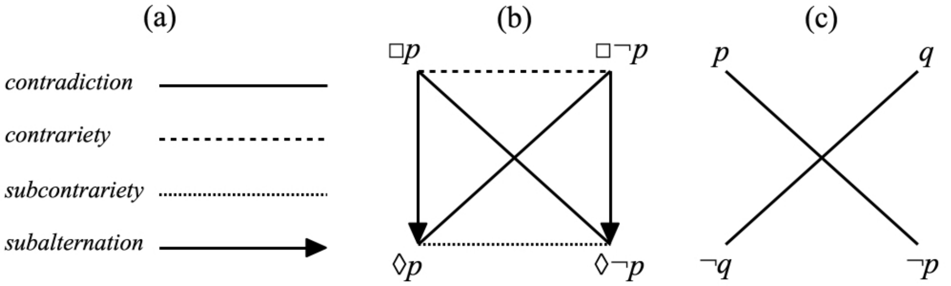
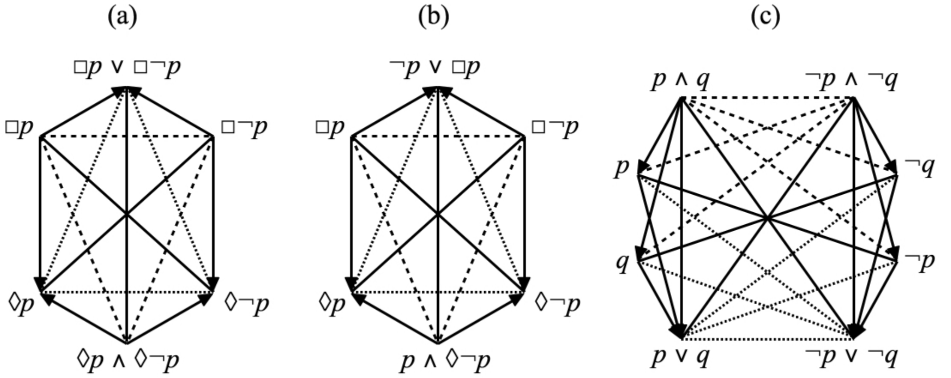


Figure 2: (a) (strong) JSB hexagon in KD, (b) (weak) JSB hexagon in KT, (c) Buridan octagon in CPL.



$\varphi, \psi \in \mathscr{F}$, it holds that $R_{\mathsf{S}}(\varphi, \psi)$ iff $R_{\mathsf{S}'}(\gamma(\varphi), \gamma(\psi))$ [21, Definition 4]. For example, the diagrams in Fig. 2(a-b) are Aristotelian isomorphic to each other; both of them are called 'Jacoby-Sesmat-Blanché (JSB) hexagons'.

One of the main ongoing lines of research in logical geometry involves developing a systematic classification of Aristotelian diagrams. For these purposes we make crucial use of the notion of Aristotelian isomorphism. In particular, a *family of Aristotelian diagrams* — or *Aristotelian family*, for short — is defined to be a maximally large class of Aristotelian isomorphic diagrams, i.e. a class $\mathscr{C}$ such that (i) any two diagrams belonging to $\mathscr{C}$ are Aristotelian isomorphic to each other, and (ii) if a diagram $\mathscr{D}$ belongs to $\mathscr{C}$, and $\mathscr{D}$ is Aristotelian isomorphic to another diagram $\mathscr{D}'$, then $\mathscr{D}'$ also belongs to $\mathscr{C}$. For example, the diagram shown in Fig. 1(b) belongs to the Aristotelian family of *classical squares*, while Fig. 1(c) shows an example of another Aristotelian family, viz. the *degenerate squares* (in which there are no Aristotelian relations whatsoever, except for the two contradiction relations). Furthermore, the diagrams in Fig. 2(a-b) both belong to the Aristotelian family of *JSB hexagons*, and finally, Fig. 2(c) shows an example of the family of *Buridan octagons* (also cf. Footnote 1).

Another central notion in logical geometry is that of the Boolean closure of a given fragment or Aristotelian diagram [14, 20, 21]. Consider a finite fragment $\mathscr{F} \subseteq \mathscr{L}_{\mathsf{S}}$. The *Boolean closure of* $\mathscr{F}$ *in* S, denoted $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$, is the smallest Boolean algebra that contains $\mathscr{F}$, i.e. such that (i) $\mathscr{F} \subseteq \mathbb{B}_{\mathsf{S}}(\mathscr{F})$, and (ii) for all Boolean algebras $\mathbb{B}$ such that $\mathscr{F} \subseteq \mathbb{B}$, it holds that $\mathbb{B}_{\mathsf{S}}(\mathscr{F}) \subseteq \mathbb{B}$. Similarly, if $\mathscr{D}$ is an Aristotelian diagram for $\mathscr{F}$, then any Aristotelian diagram that visualizes all (contingent) formulas from $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$ is said to be the *Boolean closure of the diagram* $\mathscr{D}$. For example, it is well-known that the Boolean closure (in KD) of the classical square in Fig. 1(b) is the JSB hexagon shown in Fig. 2(a), and that the Boolean closure (in CPL) of the Buridan octagon in Fig. 2(c) is a rhombic dodecahedron, a 3D diagram with 14 contingent formulas [38, 58, 61].

In its theoretical study of Aristotelian diagrams, logical geometry makes extensive use of *bitstring semantics* [21, 62]. Bitstrings are combinatorial representations of formulas that provide a concrete grip on the logical behavior of a given fragment (in particular, its Boolean complexity and the Aristotelian relations holding among its formulas). A systematic technique for assigning bitstrings to any finite fragment $\mathscr{F}$ of formulas in any logical system S is described in detail in [21]; here we will focus on those aspects that are relevant for our current purposes. Given a fragment $\mathscr{F} = \{\varphi_1, \ldots, \varphi_m\} \subseteq \mathscr{L}_{\mathsf{S}}$, the *partition of* S *induced by* $\mathscr{F}$ is defined as

$$\Pi_{\mathsf{S}}(\mathscr{F}) := \{\alpha \in \mathscr{L}_{\mathsf{S}} \mid \alpha \equiv_{\mathsf{S}} \pm\varphi_1 \wedge \cdots \wedge \pm\varphi_m, \text{ and } \alpha \text{ is } \mathsf{S}\text{-consistent}\}$$

(where $+\varphi = \varphi$ and $-\varphi = \neg\varphi$). The set $\Pi_{\mathsf{S}}(\mathscr{F})$ is called a 'partition' of (the class of models of) S because its elements are (i) jointly exhaustive, i.e. $\mathsf{S} \models \bigvee \Pi_{\mathsf{S}}(\mathscr{F})$, and (ii) mutually exclusive, i.e. $\mathsf{S} \models \neg(\alpha \wedge \beta)$ for distinct $\alpha, \beta \in \Pi_{\mathsf{S}}(\mathscr{F})$. It can be shown that every formula $\varphi \in \mathbb{B}_{\mathsf{S}}(\mathscr{F})$ is S-equivalent to a disjunction of elements of $\Pi_{\mathsf{S}}(\mathscr{F})$:[4]

$$\varphi \;\equiv_{\mathsf{S}}\; \bigvee\{\alpha \in \Pi_{\mathsf{S}}(\mathscr{F}) \mid \models_{\mathsf{S}} \alpha \rightarrow \varphi\}.$$

The bitstring semantics $\beta_{\mathsf{S}}^{\mathscr{F}}$ maps each formula $\varphi \in \mathbb{B}_{\mathsf{S}}(\mathscr{F})$ to its bitstring representation $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi) \in \{0,1\}^{|\Pi_{\mathsf{S}}(\mathscr{F})|}$, which 'keeps track' of which formulas of $\Pi_{\mathsf{S}}(\mathscr{F})$ enter into this disjunction. For example, if $\Pi_{\mathsf{S}}(\mathscr{F}) = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, then $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi) = 1011$ means that $\varphi \equiv_{\mathsf{S}} \alpha_1 \vee \alpha_3 \vee \alpha_4$. Note that $|\Pi_{\mathsf{S}}(\mathscr{F})|$ is the *length* of the bitstring $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi)$.

The Aristotelian relations can be lifted from the level of $\mathscr{L}_{\mathsf{S}}$-formulas to the level of bitstrings: two bitstrings $b_1, b_2 \in \{0,1\}^n$ are said to be

n-contradictory	iff	$b_1 \wedge b_2 = 0\cdots 0$	and	$b_1 \vee b_2 = 1\cdots 1$,
n-contrary	iff	$b_1 \wedge b_2 = 0\cdots 0$	and	$b_1 \vee b_2 \neq 1\cdots 1$,
n-subcontrary	iff	$b_1 \wedge b_2 \neq 0\cdots 0$	and	$b_1 \vee b_2 = 1\cdots 1$,
in n-subalternation	iff	$b_1 \wedge b_2 = b_1$	and	$b_1 \vee b_2 \neq b_1$.

[4]This can be understood as a kind of disjunctive normal form [64] that is 'relativized' to $\mathscr{F}$: the formula φ is rewritten as a disjunction of conjunctions of '$\mathscr{F}$-literals', i.e. formulas from $\mathscr{F}$ and their negations. (By contrast, the usual disjunctive normal form works with ordinary literals, i.e. propositional atoms and their negations.)

These relations will be abbreviated as CD_n, C_n, SC_n and SA_n, respectively. With these definitions in place, one can easily show that the Aristotelian relations holding between the formulas in (the Boolean closure of) $\mathscr{F}$ precisely correspond to those holding between the formulas' bitstring representations: for all Aristotelian relations R and formulas $\varphi, \psi \in \mathbb{B}_{\mathsf{S}}(\mathscr{F})$, it holds that $R_{\mathsf{S}}(\varphi, \psi)$ iff $R_{|\Pi_{\mathsf{S}}(\mathscr{F})|}(\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi), \beta_{\mathsf{S}}^{\mathscr{F}}(\psi))$ [21, Theorem 2].

Furthermore, it can also be shown that the Boolean closure $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$ of a fragment $\mathscr{F}$ in a logical system S is isomorphic to the Boolean algebra $\{0,1\}^{|\Pi_{\mathsf{S}}(\mathscr{F})|}$ [21, Theorem 1]. The bitstring semantics of $\mathscr{F}$ (in particular, the length of the bitstrings) thus directly captures the Boolean complexity of $\mathscr{F}$. For example, it is well-known that the 4-formula fragment visualized by the classical square in Fig. 1(b) induces a tripartition [21, 22, 66], and hence is represented by bitstrings of length 3. The Boolean closure of this fragment thus has 2^3 formulas, of which $2^3 - 2 = 6$ are contingent, and shown in the Boolean closure of the square, i.e. the JSB hexagon in Fig. 2(a). Similarly, the 8-formula fragment visualized by the Buridan octagon in Fig. 2(c) induces a quadripartition [21, 22], and hence is represented by bitstrings of length 4. The Boolean closure of this fragment thus has 2^4 formulas, of which $2^4 - 2 = 14$ are contingent, and shown in the Boolean closure of the Buridan octagon, i.e. in the rhombic dodecahedron.[5]

The theoretical value of this approach should be clear. If we are interested in the Boolean complexity of $\mathscr{F}$, i.e. in the *size* of its Boolean closure, but not necessarily in the *individual formulas* in that Boolean closure, then it is better to compute the partition $\Pi_{\mathsf{S}}(\mathscr{F})$ that is induced by $\mathscr{F}$, rather than the Boolean closure $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$ itself, since the former already allows us to determine the size of $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$. Furthermore, computing $\Pi_{\mathsf{S}}(\mathscr{F})$ is exponentially less work than computing $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$, as should be clear from the following upper bounds: $|\Pi_{\mathsf{S}}(\mathscr{F})| \leq 2^{|\mathscr{F}|}$, whereas $|\mathbb{B}_{\mathsf{S}}(\mathscr{F})| = 2^{|\Pi_{\mathsf{S}}(\mathscr{F})|} \leq 2^{2^{|\mathscr{F}|}}$ [21, Footnote 50].

The bitstring length $|\Pi_{\mathsf{S}}(\mathscr{F})|$ thus provides a direct measure of the Boolean complexity of $\mathscr{F}$. Furthermore, the bitstrings themselves are also useful, in particular when we want to determine the Aristotelian relations holding between formulas. Suppose, for example, that we have formulas $\varphi_1, \varphi_2, \psi_1, \psi_2 \in \mathscr{F}$. The Aristotelian relations holding among these formulas will be visualized by any Aristotelian diagram for $\mathscr{F}$. However, if $\varphi_1 \wedge \varphi_2$ and $\psi_1 \vee \psi_2$ are *not* in $\mathscr{F}$, then these two formulas, and a fortiori the relation holding between them, will *not* be visualized by an Aristotelian diagram for $\mathscr{F}$. Nevertheless, once we have a bitstring semantics $\beta_{\mathscr{F}}^{\mathsf{S}}$, we can straightforwardly compute $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_1 \wedge \varphi_2)$ and $\beta_{\mathsf{S}}^{\mathscr{F}}(\psi_1 \vee \psi_2)$, determine the Aristotelian relation holding between these two bitstrings, and conclude that the same Aristotelian relation holds between the formulas. For example, if we know that $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_1) = 111000$, $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_2) = 110100$, $\beta_{\mathsf{S}}^{\mathscr{F}}(\psi_1) = 000010$ and

[5] Henceforth in this paper, I will often talk, informally, about the partition induced by/the Boolean complexity of some Aristotelian *diagram*. Strictly speaking, this should be understood as the partition induced by/the Boolean complexity of the *fragment* of formulas visualized by that diagram. Furthermore, when I say that the Boolean closure of an Aristotelian diagram is isomorphic to some Boolean algebra, this should be understood modulo the restriction to contingent formulas.

$\beta_{\mathsf{S}}^{\mathscr{F}}(\psi_2) = 000001$, then we can calculate $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_1 \wedge \varphi_2) = \beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_1) \wedge \beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_2) = 111000 \wedge 110100 = 110000$ and $\beta_{\mathsf{S}}^{\mathscr{F}}(\psi_1 \vee \psi_2) = \beta_{\mathsf{S}}^{\mathscr{F}}(\psi_1) \vee \beta_{\mathsf{S}}^{\mathscr{F}}(\psi_2) = 000010 \vee 000001 = 000011$; we then note that $\beta_{\mathsf{S}}^{\mathscr{F}}(\varphi_1 \wedge \varphi_2)$ and $\beta_{\mathsf{S}}^{\mathscr{F}}(\psi_1 \vee \psi_2)$ are 6-contrary (since $11000 \wedge 00011 = 00000$ and $11000 \vee 00011 \neq 11111$), and conclude that the formulas $\varphi_1 \wedge \varphi_2$ and $\psi_1 \vee \psi_2$ themselves are S-contrary.

3 Maximal Boolean Complexity

As was already described in Section 2, one of the main ongoing lines of research in logical geometry involves the classification of Aristotelian diagrams into distinct families, based on the notion of Aristotelian isomorphism. A complicating factor in this project is that diagrams belonging to the same Aristotelian family can have different Boolean complexities [21, Section 5]. For example, the diagrams in Fig. 2(a-b) are Aristotelian isomorphic to each other, and thus belong to the same Aristotelian family, viz. the family of JSB hexagons. However, the hexagon in Fig. 2(a) induces (in the modal logic KD) the tripartition $\{\Box p, \Diamond p \wedge \Diamond\neg p, \Box\neg p\}$, and thus its Boolean closure is isomorphic to $\{0,1\}^3$; by contrast, the hexagon in Fig. 2(b) induces (in KT) the quadripartition $\{\Box p, p \wedge \Diamond\neg p, \neg p \wedge \Diamond p, \Box\neg p\}$, and thus its Boolean closure is isomorphic to $\{0,1\}^4$. Because of these differences in Boolean complexity, the diagram in Fig. 2(a) is said to be a *strong* JSB hexagon, while that in Fig. 2(b) is called a *weak* JSB hexagon [47].

Examples such as these show that within Aristotelian families, there may also exist *Boolean subfamilies*. Diagrams belonging to distinct Boolean subfamilies of the same Aristotelian family are Aristotelian isomorphic to each other, but have different Boolean complexities. From a classificatory perspective, this means that for any given Aristotelian family $\mathscr{A}$, one should also make a systematic subclassification of the Boolean subfamilies of $\mathscr{A}$. A first step toward such a subclassification involves determining the lower and upper bounds on the Boolean complexity of (diagrams belonging to) $\mathscr{A}$:

- What is the *minimal* Boolean complexity of (diagrams belonging to) $\mathscr{A}$?

  In other words: what is the *smallest* number n such that there exists a logical system S and fragment $\mathscr{F} \subseteq \mathscr{L}_{\mathsf{S}}$ such that the Aristotelian diagram for $\mathscr{F}$ in S belongs to the Aristotelian family $\mathscr{A}$, and $|\Pi_{\mathsf{S}}(\mathscr{F})| = n$?

- What is the *maximal* Boolean complexity of (diagrams belonging to) $\mathscr{A}$?

  In other words: what is the *largest* number n such that there exists a logical system S and fragment $\mathscr{F} \subseteq \mathscr{L}_{\mathsf{S}}$ such that the Aristotelian diagram for $\mathscr{F}$ in S belongs to the Aristotelian family $\mathscr{A}$, and $|\Pi_{\mathsf{S}}(\mathscr{F})| = n$?

For example, concerning the Aristotelian family of JSB hexagons, we know that the Boolean subfamily of strong JSB hexagons has Boolean complexity 3, while the Boolean subfamily of weak JSB hexagons has Boolean complexity 4 (cf. supra); however, one can also ask whether there exist JSB hexagons with a

Boolean complexity of (i) strictly less than 3, or (ii) strictly higher than 4. Question (i) is relatively straightforward to answer. If an Aristotelian diagram has Boolean complexity 2, then it can be represented by bitstrings of length 2, and since there exist only $2^2 - 2 = 2$ contingent bitstrings of length 2, the diagram cannot be a hexagon (which contains $6 > 2$ contingent formulas), and thus a fortiori not a JSB hexagon. The minimal Boolean complexity of the Aristotelian family of JSB hexagons is thus indeed 3. By contrast, question (ii) is less trivial to answer.

In the next section I will therefore present a method for systematically determining the maximal Boolean complexity of any given Aristotelian family; furthermore, in Section 5, this method will be implemented computationally, in the logic programming language Prolog. The Prolog program will provide a generic description of the largest possible partition $\Pi^{\mathscr{A}}_{max}$ that can be induced by (diagrams belonging to) the Aristotelian family $\mathscr{A}$. The number $|\Pi^{\mathscr{A}}_{max}|$ is thus the maximal Boolean complexity of $\mathscr{A}$. Furthermore, the program also automatically computes the bitstring semantics $\beta^{\mathscr{A}}_{max}$ based on this maximal partition $\Pi^{\mathscr{A}}_{max}$, which enables us to assign a maximal bitstring representation to (diagrams belonging to) $\mathscr{A}$.

The task of automatically determining the maximal Boolean complexity of a given Aristotelian family is of considerable theoretical importance within the framework of logical geometry. First of all, as was already explained above, this issue is directly relevant toward obtaining a systematic classification of families of Aristotelian diagrams and, especially, their Boolean subfamilies. Secondly, the generic description of the largest possible partition $\Pi^{\mathscr{A}}_{max}$ and bitstring semantics $\beta^{\mathscr{A}}_{max}$ induced by the Aristotelian family $\mathscr{A}$ also provides us with a deeper understanding of the Boolean properties of that Aristotelian family in general. For example, (diagrams belonging to) the Boolean subfamilies of $\mathscr{A}$ that have *non*-maximal Boolean complexity induce partitions that can be seen as *subsets* of the maximal partition $\Pi^{\mathscr{A}}_{max}$, and thus receive bitstring representations that are *substrings* of those assigned by the maximal bitstring representation $\beta^{\mathscr{A}}_{max}$. (These claims will be further explained and illustrated in Section 6).

These advantages are all situated at a relatively abstract/theoretical level. To provide some further context to the task of determining the maximal Boolean complexity of a given Aristotelian family, I will finish this section by drawing a comparison with two related tasks, which are more concrete/application-oriented in nature. However, the first of these other tasks presents us with major practical difficulties when we want to solve it in an automated, computational fashion, while the second one turns out to be underspecified and thus cannot be systematically solved at all.

The first related task is, given some fragment $\mathscr{F}$ in some logical system S, to compute the partition $\Pi_{\mathsf{S}}(\mathscr{F})$ induced by that fragment in that logic (and hence also its Boolean complexity $|\Pi_{\mathsf{S}}(\mathscr{F})|$). This task starts from a specific fragment and, especially, a specific logical system. Consequently, automatically solving this talk will require making use of dedicated reasoning algorithms (theorem provers, satisfiability checkers, etc.) for every specific (fragment and) logical system that we happen to be interested in, which is practically unfeasible. By contrast, the task of determining the maximal Boolean complexity of an Aristotelian family starts from that Aristotelian family in general, regardless of any specific diagram belonging to that family, and consequently, it can automatically be solved using general-purpose logic programming tools only.

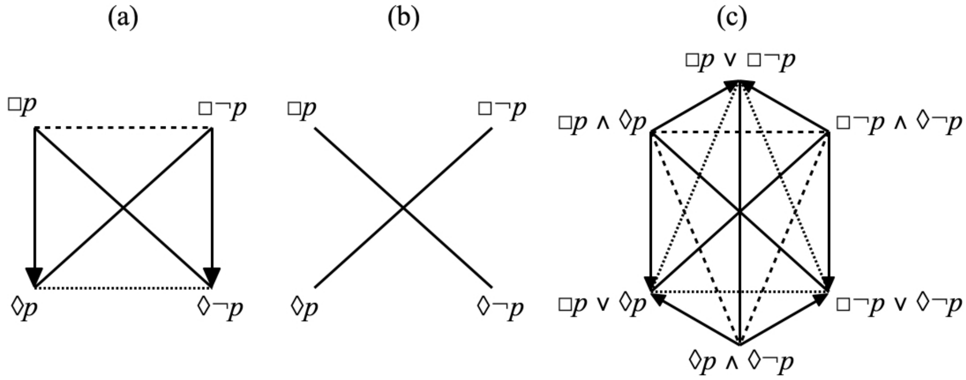
This practical problem is related to a theoretical issue that is well-known in logical geometry, viz. the fact that Aristotelian diagrams are highly *logic-sensitive* [15, 17, 21, 45, 50]. This issue has several manifestations, and the clearest one is probably the fact that the Aristotelian relation holding between two formulas partially depends on the logic that is being assumed in the background.[6] More formally, there exist logical systems S_1 and S_2 (with a common language $\mathscr{L} = \mathscr{L}_{\mathsf{S}_1} = \mathscr{L}_{\mathsf{S}_2}$) and formulas $\varphi, \psi \in \mathscr{L}$, such that φ and ψ stand in one Aristotelian relation in S_1, and in another Aristotelian relation in S_2. To give an extreme example, the formulas $\Box p$ and $\Box\neg p$ do not stand in any Aristotelian relation in the basic normal modal logic K, they are contrary in KD, subcontrary in KF (where F is the axiom $\Diamond p \to \Box p$), and contradictory in KDF.[7] If we have multiple formulas, these issues of logic-sensitivity multiply accordingly. For example, consider the fragment $\mathscr{F} = \{\Box p, \Box\neg p, \Diamond p, \Diamond\neg p\}$. One can easily show that in KD, the Aristotelian diagram for $\mathscr{F}$ is a classical square of opposition, as shown in Fig. 3(a), whereas in K, the Aristotelian diagram for that same fragment $\mathscr{F}$ is a degenerate square (or 'X of opposition' [5]), as shown in Fig. 3(b). Since the fragment $\mathscr{F}$ gives rise to diagrams belonging to different Aristotelian families in KD and K, it should not come as a surprise that it also induces different partitions in these two logical systems. In particular, it can be shown that $\Pi_{\mathsf{KD}}(\mathscr{F}) = \{\Box p, \Diamond p \wedge \Diamond\neg p, \Box\neg p\}$ (and thus $\mathbb{B}_{\mathsf{KD}}(\mathscr{F}) \cong \{0,1\}^3$), whereas $\Pi_{\mathsf{K}}(\mathscr{F}) = \{\Box p \wedge \Diamond p, \Diamond p \wedge \Diamond\neg p, \Box\neg p \wedge \Diamond\neg p, \Box p \wedge \Box\neg p\}$ (and thus $\mathbb{B}_{\mathsf{K}}(\mathscr{F}) \cong \{0,1\}^4$).

This brings us to the second related task. In order to avoid the issue of logic-sensitivity, the task description no longer refers to a specific logical system, but rather replaces this with information regarding the Aristotelian relations holding between the fragment's formulas. This is a natural move to make, since in logical geometry we are not primarily interested in fragments 'in isolation', but rather in Aristotelian *diagrams*, which represent fragments of formulas *together with* the Aristotelian relations holding between those formulas. The second related task thus is, given some fragment of formulas and the Aristotelian relations holding between those formulas, to compute the partition induced by that fragment and those relations.

[6]Other manifestations of logic-sensitivity include the fact that the formulas visualized by an Aristotelian diagram are assumed to be S-contingent and pairwise non-S-equivalent (cf. supra). After all, a formula might be S_1-contingent but not S_2-contingent, and two formulas might be non-S_1-equivalent but S_2-equivalent, for different logical systems S_1 and S_2. These specific issues need not concern us further in this paper, but see [17] for a more systematic discussion.

[7]The logic K is sound and complete with respect to the class of all Kripke frames; the logics KD, KF and KDF are sound and complete with respect to the classes of serial, partially functional, and totally functional Kripke frames, respectively. For a Kripke frame $\langle W, R\rangle$ and $w \in W$, we write $R[w] := \{v \in W \mid wRv\}$; seriality then means that $\forall w \in W : |R[w]| \geq 1$, partial functionality means that $\forall w \in W : |R[w]| \leq 1$, and total functionality means that $\forall w \in W : |R[w]| = 1$.

Figure 3: (a) Classical square in KD, (b) degenerate square for the same fragment in K, (c) JSB hexagon in KD and K.



In some cases this second task can indeed be solved without taking into account the details of any specific logical system. For example, if we are given, once again, the fragment $\mathscr{F} = \{\Box p, \Box\neg p, \Diamond p, \Diamond\neg p\}$, together with the information that these formulas constitute a classical square of opposition (i.e. that $\Box p$ is contradictory to $\Box\neg p$, that $\Box p$ is contrary to $\Box\neg p$, that there is a subalternation from $\Box p$ to $\Diamond p$, etc.), then one can show that it induces the partition $\{\Box p, \Diamond p \wedge \Diamond\neg p, \Box\neg p\}$, without referring to any specific logical system [17].

However, there also exist cases where this second task simply cannot be solved uniformly. This is due to another, more subtle manifestation of the issue of logic-sensitivity in Aristotelian diagrams: there exist logical systems S_1 and S_2 and fragments $\mathscr{F}$ such that (i) the formulas of $\mathscr{F}$ stand in exactly the same Aristotelian relations in S_1 and in S_2 — in other words, $\mathscr{F}$ gives rise to diagrams belonging to the same Aristotelian family in S_1 and in S_2 —, and yet, (ii) $\mathscr{F}$ has different Boolean complexities in S_1 and S_2 [17]. For example, consider the fragment $\mathscr{F} = \{\Box p \wedge \Diamond p, \Box\neg p \wedge \Diamond\neg p, \Box p \vee \Diamond p, \Box\neg p \vee \Diamond\neg p, \Box p \vee \Box\neg p, \Diamond p \wedge \Diamond\neg p\}$. One can easily show that the Aristotelian relations holding among the formulas of $\mathscr{F}$ are exactly the same in KD as in K: in both logical systems, this fragment gives rise to a JSB hexagon, as shown in Fig. 3(c). However, it can be shown that $\Pi_{\mathsf{KD}}(\mathscr{F}) = \{\Box p, \Diamond p \wedge \Diamond\neg p, \Box\neg p\}$ (and thus $\mathbb{B}_{\mathsf{KD}}(\mathscr{F}) \cong \{0,1\}^3$), whereas $\Pi_{\mathsf{K}}(\mathscr{F}) = \{\Box p \wedge \Diamond p, \Diamond p \wedge \Diamond\neg p, \Box\neg p \wedge \Diamond\neg p, \Box p \wedge \Box\neg p\}$ (and thus $\mathbb{B}_{\mathsf{K}}(\mathscr{F}) \cong \{0,1\}^4$). (In other words: the fragment $\mathscr{F}$ gives rise to a *strong* JSB hexagon in KD, but to a *weak* JSB hexagon in K.)

This example shows that the Boolean complexity of a given fragment cannot always be uniquely determined from the Aristotelian relations holding among the fragment's formulas.[8] The second related task is thus *underspecified*: the task de-

[8] Essentially, the reason for this is that the Aristotelian relations are strictly *binary* in nature, whereas the Boolean complexity of a fragment involves considerations that go beyond binary relations. For example, consider the three (!) formulas $\Box p \wedge \Diamond p$, $\Box\neg p \wedge \Diamond\neg p$ and $\Diamond p \wedge \Diamond\neg p$ from the fragment $\mathscr{F}$ introduced above. If we restrict ourselves to binary relations, then we note that both in

scription does not always provide enough information to guarantee a unique solution. By contrast, the *highest possible* Boolean complexity of a given fragment can indeed be uniquely determined from the Aristotelian relations holding among the fragment's formulas — or equivalently: the maximal Boolean complexity $|\Pi^{\mathscr{A}}_{max}|$ of a given Aristotelian family $\mathscr{A}$ can be uniquely determined. This latter formulation is exactly the original task we were interested in, and to which we turn now.

4 Computing Maximal Boolean Complexity

In this section, I will informally describe a straightforward method for computing the maximal Boolean complexity of a given Aristotelian family. In the next section, I will then present a Prolog implementation of this method, and discuss some of its design features.

Consider an arbitrary logical system S and fragment $\mathscr{F} = \{\varphi_1, \dots, \varphi_m\} \subseteq \mathscr{L}_\mathsf{S}$. The fragment $\mathscr{F}$ gives rise in S to an Aristotelian diagram belonging to some Aristotelian family $\mathscr{A}$. Recall that the partition induced by $\mathscr{F}$ in S is defined as follows:

$$\Pi_\mathsf{S}(\mathscr{F}) := \{\alpha \in \mathscr{L}_\mathsf{S} \mid \alpha \equiv_\mathsf{S} \pm\varphi_1 \wedge \cdots \wedge \pm\varphi_m, \text{ and } \alpha \text{ is S-consistent}\}.$$

Up to logical equivalence, the partition $\Pi_\mathsf{S}(\mathscr{F})$ thus consists of S-consistent conjunctions of (possibly negated) formulas from $\mathscr{F}$. The condition of S-consistency can be understood as a 'filter' on the conjunctions that end up in the partition $\Pi_\mathsf{S}(\mathscr{F})$: if more and more conjunctions are S-inconsistent (i.e. no longer pass the filter), the partition will get smaller and smaller. Since we are interested in determining the largest possible partition (i.e. the smallest number of S-inconsistent formulas), we will now explore the notion of S-inconsistency in some more detail.

A conjunction $\psi_1 \wedge \cdots \wedge \psi_k$ — with all conjuncts being (possibly negated) formulas from $\mathscr{F}$ — will be called $\mathscr{A}$*-inconsistent* iff it contains two conjuncts ψ_i and ψ_j that are S-contradictory or S-contrary to each other.[9] Analogously, the conjunction $\psi_1 \wedge \cdots \wedge \psi_k$ will be called $\mathscr{A}$*-consistent* iff it does not contain conjuncts ψ_i and ψ_j that are S-contradictory or S-contrary to each other. The notions of $\mathscr{A}$-(in)consistency are defined relative to the Aristotelian family $\mathscr{A}$, because they explicitly refer to the Aristotelian relations specified by $\mathscr{A}$ (in particular, its contradiction and contrariety relations).

K and in KD, these formulas are pairwise contrary to each other, as shown in Fig. 3(c). However, if we consider all three formulas simultaneously, then we do note a difference between the two logical systems: in KD the disjunction of these three formulas is a tautology, but in K it is not.

[9]The difference between S-inconsistency and $\mathscr{A}$-inconsistency is thus that for the S-inconsistency of a conjunction, we take *all* conjuncts into account, whereas for its $\mathscr{A}$-inconsistency we require that the inconsistency can be 'pinpointed' to just *two* conjuncts. This is essentially due to the strictly binary nature of the Aristotelian relations (also cf. Footnote 8). Furthermore, note that we only refer to the relations of contradiction and contrariety in the definition of $\mathscr{A}$-inconsistency, because these are the only two Aristotelian relations that imply the S-inconsistency of the conjunction of their relata: $CD_\mathsf{S}(\psi_i, \psi_j)$ and $C_\mathsf{S}(\psi_i, \psi_j)$ both imply that $\models_\mathsf{S} \neg(\psi_i \wedge \psi_j)$, whereas $SC_\mathsf{S}(\psi_i, \psi_j)$ and $SA_\mathsf{S}(\psi_i, \psi_j)$ do not imply this.

One can easily show that if a conjunction of (possibly negated) $\mathscr{F}$-formulas is $\mathscr{A}$-inconsistent, then it is also S-inconsistent.[10] However, the converse does not hold: a conjunction can be S-inconsistent without being $\mathscr{A}$-inconsistent, i.e. while still being $\mathscr{A}$-consistent (a concrete example will be provided below). We can thus distinguish between two types of S-inconsistent conjunctions of (possibly negated) $\mathscr{F}$-formulas: (i) those that are $\mathscr{A}$-inconsistent and (ii) those that are $\mathscr{A}$-consistent.

For an illustration of this distinction, consider the fragment $\mathscr{F} = \{\Box p, \Box\neg p, \Diamond p, \Diamond\neg p, \Diamond p \wedge \Diamond\neg p, \Box p \vee \Box\neg p\}$. In KD this fragment gives rise to the diagram shown in Fig. 2(a), which belongs to the Aristotelian family of JSB hexagons. As to the contradiction and contrariety relations among $\mathscr{F}$-formulas, we thus have:

$$\begin{array}{ll} CD_{\mathsf{KD}}(\Box p, \Diamond\neg p), & C_{\mathsf{KD}}(\Box p, \Diamond p \wedge \Diamond\neg p), \\ CD_{\mathsf{KD}}(\Box\neg p, \Diamond p), & C_{\mathsf{KD}}(\Box\neg p, \Diamond p \wedge \Diamond\neg p), \\ CD_{\mathsf{KD}}(\Box p \vee \Box\neg p, \Diamond p \wedge \Diamond\neg p), & C_{\mathsf{KD}}(\Box p, \Box\neg p). \end{array}$$

Now consider the following conjunctions of $\mathscr{F}$-formulas:

- $\Box p \wedge \Box\neg p \wedge (\Diamond p \wedge \Diamond\neg p)$

  It is easy to show that this formula is KD-inconsistent. Furthermore, it is also JSB-inconsistent, because its first two conjuncts are KD-contrary in the JSB hexagon (cf. the list of contradictions and contrarieties above).

- $\Diamond p \wedge \Diamond\neg p \wedge (\Box p \vee \Box\neg p)$

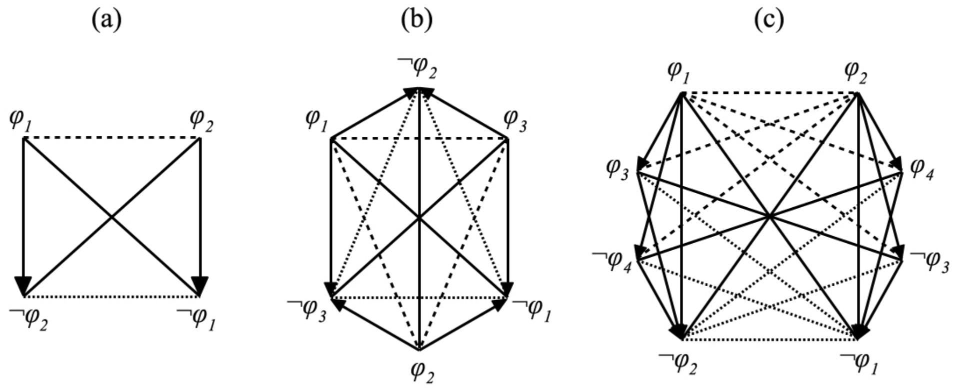
  It is again easy to show that this formula is KD-inconsistent. However, it is *not* JSB-inconsistent, because it does not contain two conjuncts that are contradictory or contrary in the JSB hexagon. In particular:

  - $\Diamond p$ and $\Diamond\neg p$ are neither KD-contradictory nor KD-contrary,
  - $\Diamond p$ and $\Box p \vee \Box\neg p$ are neither KD-contradictory nor KD-contrary,
  - $\Diamond\neg p$ and $\Box p \vee \Box\neg p$ are neither KD-contradictory nor KD-contrary.

Finally, it will also be useful to draw a distinction between the particular diagrams belonging to some Aristotelian family, and a more abstract, 'generic' description of that family (which does not refer to any specific logical system, but just specifies a configuration of Aristotelian relations holding between formulas). For example, the diagram in Fig. 1(b) is a particular member of the Aristotelian family of classical squares, the diagrams in Fig. 2(a–b) are two particular members of the family of JSB hexagons, and the diagram in Fig. 2(c) is a particular member of the family of Buridan octagons. By contrast, the Aristotelian families of classical squares, JSB hexagons and Buridan octagons can generically be described as in Fig. 4(a-c), respectively. The fragment of formulas appearing in the generic description of the Aristotelian family $\mathscr{A}$ will henceforth be called $\mathscr{F}_{\mathscr{A}}$; for example, in Fig. 4(a-c) we find that:

[10] Proof: if the conjunction $\psi_1 \wedge \cdots \wedge \psi_k$ is $\mathscr{A}$-inconsistent, then it contains conjuncts ψ_i, ψ_j that are S-contradictory or S-contrary to each other; hence $\models_{\mathsf{S}} \neg(\psi_i \wedge \psi_j)$, and thus a fortiori also $\models_{\mathsf{S}} \neg(\psi_1 \wedge \cdots \wedge \psi_k)$, i.e. the entire conjunction is S-inconsistent.

Figure 4: Generic descriptions of the Aristotelian families of (a) classical squares, (b) JSB hexagons, and (c) Buridan octagons.



- $\mathscr{F}_{classicalsquare} = \{\varphi_1, \varphi_2, \neg\varphi_1, \neg\varphi_2\}$,
- $\mathscr{F}_{JSB} = \{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1, \neg\varphi_2, \neg\varphi_3\}$,
- $\mathscr{F}_{Buridan} = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \neg\varphi_1, \neg\varphi_2, \neg\varphi_3, \neg\varphi_4\}$.

We are now ready to specify a straightforward method for computing the largest possible partition that can be induced by any Aristotelian family $\mathscr{A}$:

1. compute the conjunctions of (possibly negated) formulas from $\mathscr{F}_{\mathscr{A}}$,
2. discard the conjunctions that are $\mathscr{A}$-inconsistent, but
3. keep the conjunctions that are $\mathscr{A}$-consistent.

This method is guaranteed to yield the *largest* possible partition that can be induced by $\mathscr{A}$, because we only discard a conjunction (i.e. make the partition smaller) if we are forced to do so by the contradiction and contrariety relations present in $\mathscr{A}$, i.e. if that conjunction is $\mathscr{A}$-inconsistent. All other, $\mathscr{A}$-consistent conjunctions are kept on board, and thus the resulting partition will be as large as is 'allowed' by $\mathscr{A}$. More formally, the largest possible partition that can be induced by the Aristotelian family $\mathscr{A}$ (with $\mathscr{F}_{\mathscr{A}} = \{\varphi_1, \dots, \varphi_m\}$) looks as follows:

$$\Pi^{\mathscr{A}}_{max} = \{\pm\varphi_1 \wedge \cdots \wedge \pm\varphi_m \mid \pm\varphi_1 \wedge \cdots \wedge \pm\varphi_m \text{ is } \mathscr{A}\text{-consistent}\},$$

and consequently, the maximal Boolean complexity of $\mathscr{A}$ is $|\Pi^{\mathscr{A}}_{max}|$.

5 The Prolog Implementation

I will now describe a computational implementation of the method described above. As was already discussed in Section 3, the task of computing the maximal Boolean complexity of an Aristotelian family is not complicated by matters of logic-sensitivity, and hence we only need to make use of general-purpose programming tools. More concretely, the implementation is done in the logic programming language Prolog [6, 7]. It crucially relies on recursion to compute the maximal partition, which closely resembles the way in which partitions are computed 'by hand' in concrete applications — for specific examples, see [21, Section 5.2] and [18, Section 5.1].

The full code of the Prolog program can be found in the online appendix to this paper. It only makes use of basic language features, which are compatible with all the main Prolog implementations (e.g. SICStus, SWI-Prolog and YAP). Consequently, it does not require a local installation of Prolog, but can even be run on online tools such as SWISH (a limited subset of SWI-Prolog).[11] This should help to increase the program's accessibility and usability among the broad and interdisciplinary community of researchers who use and study Aristotelian diagrams.

I begin by discussing some design decisions regarding data representation. An Aristotelian family $\mathscr{A}$ is represented in the program by means of two lists: a list of formulas (essentially corresponding to the fragment $\mathscr{F}_{\mathscr{A}}$ introduced above) and a list of the Aristotelian relations holding among those formulas. Since Aristotelian diagrams are closed under negation, we specify only half of the formulas; for example, recall from Figure 4(a) that $\mathscr{F}_{classicalsquare} = \{\varphi_1, \varphi_2, \neg\varphi_1, \neg\varphi_2\}$; this will not be represented as `[phi1, phi2, not(phi1), not(phi2)]`, but simply as `[phi1, phi2]`. Furthermore, the list of Aristotelian relations only specifies the contrarieties (with `c(phi,psi)` meaning that φ and ψ are contrary to each other). Since Aristotelian diagrams are closed under negation (represented by the functor `not`), there is no need to explicitly list the contradiction relations (every formula is contradictory to exactly one formula, viz. its negation); furthermore, given the contradictions (implicitly represented by means of negation) and the contrarieties (explicitly represented in the list of Aristotelian relations), all other Aristotelian relations can be derived: if $C(\varphi,\psi)$, then $SC(\neg\varphi,\neg\psi)$, $SA(\varphi,\neg\psi)$ and $SA(\psi,\neg\varphi)$ [59, Lemmas 2 and 3]. For example:

- the family of classical squares is represented by the lists `[phi1,phi2]` and `[c(phi1,phi2)]`; cf. Fig. 4(a)
- the family of JSB hexagons is represented by the lists `[phi1,phi2,phi3]` and `[c(phi1,phi2), c(phi1,phi3), c(phi2,phi3)]`; cf. Fig. 4(b)
- the family of Buridan octagons is represented by the lists `[phi1,phi2,phi3,phi4]` and `[c(phi1,phi2), c(phi1,phi4), c(phi2,phi3), c(phi1,not(phi3)), c(phi2,not(phi4))]`; cf. Fig. 4(c).

Finally, the aim of the Prolog program is to compute a (maximal) partition; this partition is represented as a list of lists, where the inner lists should be read as conjunctions. For example, a partition of the form $\{\varphi_1 \wedge \varphi_2, \varphi_3 \wedge \neg\varphi_4, \neg\varphi_5 \wedge \varphi_6\}$ is represented as `[[phi1,phi2], [phi3,not(phi4)], [not(phi5),phi6]]`.

[11] See `https://swish.swi-prolog.org`.

The main predicate of the Prolog program is `maxpartition/3`. In particular,

```
maxpartition(+Fragment, +Contrarieties, -Partition)
```

means that `Partition` is the largest possible partition that can be induced by the Aristotelian family represented by `Fragment` and `Contrarieties`. This predicate is defined recursively. The base case states that if the fragment contains just a single formula φ, then it induces the partition $\{\varphi, \neg\varphi\}$ (regardless of `Contrarieties`). In the recursive case, we consider fragments containing more than one formula, say $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$; we first recursively compute the partition induced by $\{\varphi_2, \ldots, \varphi_m\}$, and then add $\pm\varphi_1$ as a conjunct to each of the conjunctions in this partition induced by $\{\varphi_2, \ldots, \varphi_m\}$. Furthermore, when adding $\pm\varphi_1$ as a conjunct to a conjunction, we make sure to maintain $\mathscr{A}$-consistency (with $\mathscr{A}$ being the Aristotelian family that is represented by `Fragment` and `Contrarieties`): if the conjunction contains the contradictory of $\pm\varphi_1$ (implicitly represented by means of negation, i.e. the functor `not`) or one of the contraries of $\pm\varphi_1$ (explicitly represented in `Contrarieties`) as one of its conjuncts, then we discard that conjunction (since adding $\pm\varphi_1$ to it would have led to an $\mathscr{A}$-inconsistent formula).

Once the definition of the predicate `maxpartition/3` is in place, it becomes trivial to also define the predicate `maxbooleancomplexity/3`. In particular,

```
maxbooleancomplexity(+Fragment, +Contrarieties, -MBC)
```

means that `MBC` is the maximal Boolean complexity of the Aristotelian family represented by `Fragment` and `Contrarieties`.

The conjunctions in the partition computed by `maxpartition/3` tend to be unnecessarily long. For example, if we use this predicate to compute the maximal partition induced by the family of JSB hexagons — cf. Fig. 4(b) —, it yields conjunctions such as $\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3$. However, since the JSB hexagon has subalternations from φ_1 to $\neg\varphi_2$ and to $\neg\varphi_3$, this conjunction can be simplified to φ_1. The predicate `maxpartitionsimple/3` does exactly the same as `maxpartition/3`, but simplifies the conjunctions as much as possible, based on the available `Contrarieties`. (For example, given the contrariety $C(\varphi, \psi)$, the conjunction $\varphi \wedge \neg\psi$ can be simplified to φ, while $\neg\varphi \wedge \psi$ can be simplified to ψ.)

Finally, once the maximal partition has been computed (using either `maxpartition/3` or `maxpartitionsimple/3`), we can also compute the maximal bitstring representation that it gives rise to. In particular,

```
maxbitstrings(+Fragment, +Contrarieties, -Bitstrings)
```

means that `Bitstrings` contains the bitstring representations of the formulas in `Fragment`, based on the maximal partition that is induced by the Aristotelian family represented by `Fragment` and `Contrarieties`. For example, the family of JSB hexagons — cf. Fig. 4(b) — induces the maximal partition $\Pi^{JSB}_{max} = \{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3\}$, which gives rise to the maximal bitstring representation β^{JSB}_{max}. Since $|\Pi^{JSB}_{max}| = 4$, the representation β^{JSB}_{max} works with bitstrings of length 4. For example, we trivially have $\beta^{JSB}_{max}(\varphi_1) = 1000$ — since φ_1 is (equivalent to) the first formula in Π^{JSB}_{max} —, but also $\beta^{JSB}_{max}(\neg\varphi_2) = 1011$ — since $\neg\varphi_2$ is equivalent to the disjunction of the first, third and fourth formula in Π^{JSB}_{max}. The variable `Bitstrings` is a list of lists, where the inner lists are bitstrings together with the formulas that they represent. For example, in the case of the family of JSB hexagons, `Bitstrings` has the form

```
[[1,0,0,0,phi1], [0,1,0,0,phi2], [0,0,1,0,phi3]],
```

which means that $\beta^{JSB}_{max}(\varphi_1) = 1000$, $\beta^{JSB}_{max}(\varphi_2) = 0100$ and $\beta^{JSB}_{max}(\varphi_3) = 0010$.

6 Theoretical Fruitfulness

As was already explained in Section 3, the task of determining the maximal Boolean complexity of any given Aristotelian family is of considerable theoretical importance within the framework of logical geometry. The Prolog program described in Section 5 solves this task in an automated, computational fashion. I will now illustrate the theoretical fruitfulness of this program, by applying it to some well-known Aristotelian families; I will also show that these results shed new light on the systematic classification of Aristotelian families and their Boolean subfamilies.

6.1 The Aristotelian Families of Classical and Degenerate Squares

We start by considering the best-known and most widely used Aristotelian family, viz. the family of *classical squares*. A concrete example of this family was shown in Fig. 1(b); its generic description is in Fig. 4(a). We can now use the Prolog program to compute the maximal partition induced by this family:

```
?- maxpartitionsimple([phi1,phi2], [c(phi1,phi2)], Partition).
Partition = [[phi1], [phi2], [not(phi1),not(phi2)]].
```

We thus find that $\Pi^{classicalsquare}_{max} = \{\varphi_1, \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\}$, and hence the maximal Boolean complexity of this Aristotelian family is $|\Pi^{classicalsquare}_{max}| = 3$.

Furthermore, the minimal Boolean complexity of this Aristotelian family is also 3. After all, if an Aristotelian diagram has Boolean complexity 2, then it can be represented by bitstrings of length 2, and since there exist only $2^2 - 2 = 2$ contingent bitstrings of length 2, the diagram cannot be a square (which contains $4 > 2$ contingent formulas), and thus a fortiori not a classical square.

Since the maximal and the minimal Boolean complexity of the family of classical squares coincide, this Aristotelian family is *Boolean homogeneous*: all diagrams belonging to it have the same Boolean complexity (viz. 3). Using more classification-oriented terminology: the Aristotelian family of classical squares does not have distinct Boolean subfamilies.

We now turn to the other Aristotelian family of squares, viz. the family of *degenerate squares*. A concrete example of this family was shown in Fig. 1(c); its generic description is in Fig. 5(a). Diagrams in this family do not contain any pairs of contrary formulas, so when running the Prolog program, we specify `Contrarieties` to be the empty list `[]`:

```
?- maxpartitionsimple([phi1,phi2], [], Partition).
Partition = [[phi1,phi2], [phi1,not(phi2)],
             [not(phi1),phi2], [not(phi1),not(phi2)]].
```

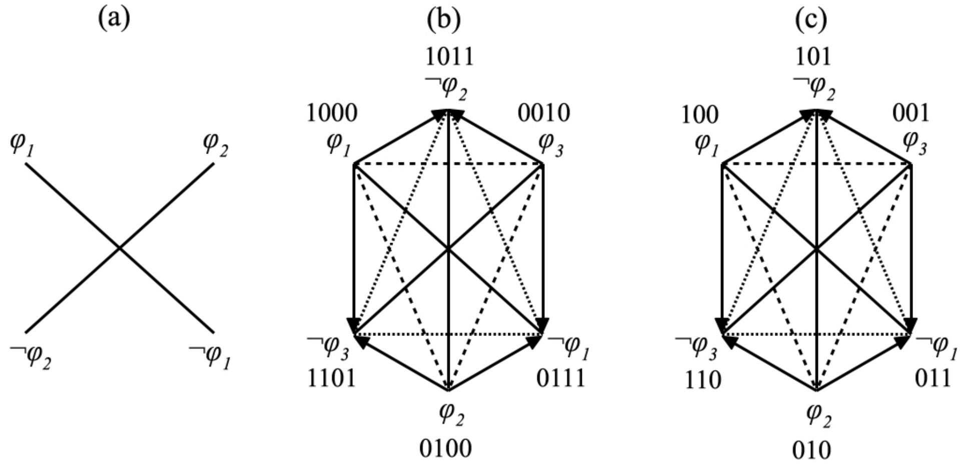
We thus find that $\Pi_{max}^{degeneratesquare} = \{\varphi_1 \wedge \varphi_2, \varphi_1 \wedge \neg\varphi_2, \neg\varphi_1 \wedge \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\}$, and hence the maximal Boolean complexity of this Aristotelian family is 4.

Furthermore, it can be shown that the minimal Boolean complexity of this Aristotelian family is also 4. After all, its diagrams contain formulas that are *unconnected* (i.e. do not stand in any Aristotelian relation at all), and it is well-known in logical geometry that representing unconnected formulas requires bitstrings of length at least 4 [21, 62]. Once again, the maximal and the minimal Boolean complexity of the family of degenerate squares coincide, so this Aristotelian family, too, is Boolean homogeneous: all diagrams belonging to it have the same Boolean complexity (viz. 4).

There exist only two Aristotelian families of squares (viz. the classical ones and the degenerate ones; cf. Footnote 1), and we have now found that both these families are Boolean homogeneous. From a classificatory perspective: as long as we restrict ourselves to Aristotelian squares, the issue of Boolean subfamilies simply does not arise. The Prolog program also sheds new light on the underlying reason for this situation. Recall from Section 4 that there exist conjunctions of formulas that are $\mathscr{A}$-consistent, but S-inconsistent (for some Aristotelian family $\mathscr{A}$ and logical system S). However, this can only happen if the conjunction contains *at least three* conjuncts (recall Footnotes 8 and 9 on the strictly *binary* nature of the Aristotelian relations). However, Aristotelian squares contain only *two* formulas φ_1 and φ_2 (and their negations), and hence give rise to conjunctions of only two conjuncts. Consequently, if we consider any square — i.e. any diagram belonging to any Aristotelian family $\mathscr{A}$ of squares — that is defined in some logical system S, then the conjunctions in the partition induced by this square will be not only $\mathscr{A}$-consistent, but also S-consistent.[12] This changes drastically when we move beyond the squares, as we will do next.

[12] In the case of the partition induced by a classical square, the (simplified) conjunctions φ_1 and φ_2 are S-consistent because they appear as formulas in an Aristotelian diagram, while the conjunction $\neg\varphi_1 \wedge \neg\varphi_2$ is S-consistent because of the S-subcontrariety of $\neg\varphi_1$ and $\neg\varphi_2$. In the case of the partition induced by a degenerate square, all four conjunctions are S-consistent because of the S-unconnectedness of φ_1 and φ_2: if either of these four conjunctions were S-inconsistent, then φ_1 and φ_2 would effectively stand in some Aristotelian relation in S after all (and thus no longer be S-unconnected).

Figure 5: (a) Generic description of the Aristotelian family of degenerate squares, (b) maximal bitstring representation of the Aristotelian family of JSB hexagons (= bitstring representation of the Boolean subfamily of weak JSB hexagons), (c) bitstring representation of the Boolean subfamily of strong JSB hexagons.



6.2 The Aristotelian Family of JSB Hexagons

We now study the second most widely used Aristotelian family after the classical squares, viz. the family of *JSB hexagons*. Concrete examples of this family were shown in Fig. 2(a-b); its generic description is in Fig. 4(b).

We can now use the Prolog program to compute the maximal partition induced by this family:

```
?- maxpartitionsimple([phi1,phi2,phi3],
                      [c(phi1,phi2), c(phi1,phi3), c(phi2,phi3)],
                      Partition).
Partition = [[phi1], [phi2], [phi3],
             [not(phi1),not(phi2),not(phi3)]].
```

We thus find that $\Pi^{JSB}_{max} = \{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3\}$, and hence the maximal Boolean complexity of this Aristotelian family is $|\Pi^{JSB}_{max}| = 4$. We can also use the `maxbitstrings/3` predicate to compute the maximal bitstring representation β^{JSB}_{max}; this makes use of bitstrings of length 4, as shown in Fig. 5(b).

The concrete JSB hexagon in Fig. 2(b) has indeed the maximal Boolean complexity of 4, but the one in Fig. 2(a) has a lower Boolean complexity of 3; we have already seen in Section 3 that the former induces a quadripartition (in KT), whereas the latter induces a tripartition (in KD). There exist no JSB hexagons with Boolean complexity smaller than 3, because of cardinality considerations that should be familiar by now. From a classificatory perspective: the Aristotelian family of JSB hexagons has exactly two Boolean subfamilies, viz. the *strong* JSB hexagons (which have Boolean complexity 3) and the *weak* JSB hexagons (which have Boolean complexity 4).

The Prolog program also sheds new light on the relation between these two Boolean subfamilies. Recall that $\Pi^{JSB}_{max} = \{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1 \wedge \varphi_2 \wedge \neg\varphi_3\}$. For a concrete JSB hexagon (in some logical system S), the first three formulas in this partition also directly appear in the JSB hexagon itself, and will thus be not only JSB-consistent, but also S-consistent. By contrast, the fourth formula $\neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3$ is JSB-consistent, but it is not necessarily S-consistent: there exist concrete JSB hexagons and logics for which this conjunction is consistent, but also concrete JSB hexagons and logics for which it is inconsistent. For example, for the JSB hexagon in Fig. 2(b), this conjunction amounts to $\Diamond\neg p \wedge (\neg p \vee \Box p) \wedge \Diamond p$, which is KT-consistent (it is KT-equivalent to $\neg p \wedge \Diamond p$); by contrast, for the JSB hexagon in Fig. 2(a), this conjunction amounts to $\Diamond\neg p \wedge (\Box p \vee \Box\neg p) \wedge \Diamond p$, which is KD-inconsistent.

Returning to the generic description of the family of JSB hexagon in Fig. 4(b), we thus see that (i) a strong JSB hexagon induces the partition $\{\varphi_1, \varphi_2, \varphi_3\}$, whereas (ii) a weak JSB hexagon induces the (maximal) partition $\{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3\}$. The tripartition that is induced by the Boolean subfamily of strong JSB hexagons is thus a *subset* of the quadripartition that is induced by the Boolean subfamily of weak JSB hexagons. In terms of bitstring representations, this means that the bitstrings used to represent a strong JSB hexagon are *substrings* of the bitstrings that are used to represent a weak JSB hexagon: they are obtained by systematically deleting the fourth bit position, which corresponds to the conjunction $\neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3$. The bitstring representation of the Boolean subfamily of weak JSB hexagons (and hence: the maximal bitstring representation of the entire Aristotelian family of JSB hexagons) makes use of bitstrings of length 4, as shown in Fig. 5(b); by contrast, the bitstring representation of the Boolean subfamily of strong JSB hexagons makes use of bitstrings of length 3, as shown in Fig. 5(c).

These results also show that the correspondence between JSB hexagons and tripartitions is more subtle than it is sometimes taken to be [9, 22, 48, 66]. Every tripartition indeed gives rise to a JSB hexagon: if $\{\varphi_1, \varphi_2, \varphi_3\}$ is a partition, then we always obtain a JSB hexagon as in Fig. 4(b). However, the converse does not universally hold: not every JSB hexagon induces a tripartition (some JSB hexagons induce a quadripartition). To obtain a perfect correspondence, we should not refer to the entire Aristotelian family of JSB hexagons, but rather to one of its Boolean subfamilies: every tripartition gives rise to a *strong* JSB hexagon, *and vice versa*.

7 Conclusion

This paper has laid the foundations for a computational approach to logical geometry. I have described a Prolog program that automatically computes the maximal partition induced by (and thus also the maximal Boolean complexity of) any given family of Aristotelian diagrams. This constitutes a significant contribution to the ongoing effort toward developing a systematic classification of Aristotelian diagrams into Aristotelian families and Boolean subfamilies.

In this paper we have used the Prolog program to compute the maximal partitions induced by the Aristotelian families of classical squares, degenerate squares, and JSB hexagons; these results also shed new light on the logical properties of these Aristotelian families in general. In future research the Prolog program will be used to compute the maximal partitions induced by *all* Aristotelian families that are currently known (cf. Footnote 1), and to study their logical properties in further detail.

On a more ambitious level, we will also further extend the computational approach to logical geometry. One of the main goals in this respect is to develop a program for automatically determining all the families of Aristotelian diagrams (given some upper bound on diagram size or Boolean complexity, for example). After such a program has computed the distinct Aristotelian families, its output can be fed into the Prolog program described in this paper, which will then compute the maximal Boolean complexity of each Aristotelian family. In this way we can create an entire pipeline of computational tools in logical geometry.

Acknowledgements

Thanks to Hans Smessaert and Margaux Smets for their valuable feedback on an earlier version of this paper. The author holds a Postdoctoral Fellowship of the Research Foundation–Flanders (FWO). The research reported in this paper was partially carried out during a research stay at the Institut für Philosophie II of the Ruhr-Universität Bochum, which was financially supported by an FWO travel grant.

References

[1] Amgoud, L., Besnard, P., Hunter, A.: Foundations for a logic of arguments. In: Cabalar, P., Herzig, M.D.A., Pearce, D. (eds.) Logical Reasoning and Computation: Essays dedicated to Luis Fariñas del Cerro, pp. 95–107. IRIT, Toulouse (2016)

[2] Amgoud, L., Prade, H.: Can AI models capture natural language argumentation? International Journal of Cognitive Informatics and Natural Intelligence 6(3), 19–32 (2012)

[3] Amgoud, L., Prade, H.: Towards a logic of argumentation. In: Hüllermeier, E., et al. (eds.) Scalable Uncertainty Management 2012, pp. 558–565. LNCS 7520, Springer, Berlin (2012)

[4] Amgoud, L., Prade, H.: A formal concept view of formal argumentation. In: van der Gaag, L.C. (ed.) Symbolic and Quantitative Approaches to Reasoning

with Uncertainty (ECSQARU 2013), pp. 1–12. LNCS 7958, Springer, Berlin (2013)

[5] Béziau, J.Y., Payette, G.: Preface. In: Béziau, J.Y., Payette, G. (eds.) The Square of Opposition. A General Framework for Cognition, pp. 9–22. Peter Lang, Bern (2012)

[6] Blackburn, P., Bos, J., Striegnitz, K.: Learn Prolog Now! College Publications, London (2006)

[7] Bratko, I.: Prolog Programming for Artificial Intelligence (Third Edition). Addison-Wesley, Harlow (2001)

[8] Carnielli, W., Pizzi, C.: Modalities and Multimodalities. Springer, Dordrecht (2008)

[9] Ciucci, D., Dubois, D., Prade, H.: Oppositions in rough set theory. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) Rough Sets and Knowledge Technology, pp. 504–513. LNCS 7414, Springer, Berlin (2012)

[10] Ciucci, D., Dubois, D., Prade, H.: The structure of oppositions in rough set theory and formal concept analysis – toward a new bridge between the two settings. In: Beierle, C., Meghini, C. (eds.) Foundations of Information and Knowledge Systems (FoIKS 2014), pp. 154–173. LNCS 8367, Springer, Berlin (2014)

[11] Ciucci, D., Dubois, D., Prade, H.: Structures of opposition in fuzzy rough sets. Fundamenta Informaticae 142, 1–19 (2015)

[12] Ciucci, D., Dubois, D., Prade, H.: Structures of opposition induced by relations. The Boolean and the gradual cases. Annals of Mathematics and Artificial Intelligence 76, 351–373 (2016)

[13] Dekker, P.: Not *Only* Barbara. Journal of Logic, Language and Information 24, 95–129 (2015)

[14] Demey, L.: Structures of oppositions for public announcement logic. In: Béziau, J.Y., Jacquette, D. (eds.) Around and Beyond the Square of Opposition, pp. 313–339. Springer, Basel (2012)

[15] Demey, L.: Interactively illustrating the context-sensitivity of Aristotelian diagrams. In: Christiansen, H., Stojanovic, I., Papadopoulos, G. (eds.) Modeling and Using Context, pp. 331–345. LNCS 9405, Springer, Berlin (2015)

[16] Demey, L.: Aristotelian diagrams in the debate on future contingents. A methodological reflection on Hess's open future square of opposition. Submitted (2017)

[17] Demey, L.: Logic-Sensitivity in Aristotelian Diagrams. Book manuscript (2017)

[18] Demey, L.: The logical geometry of Russell's theory of definite descriptions. Submitted (2017)

[19] Demey, L.: Metalogic, metalanguage and logical geometry. Submitted (2017)

[20] Demey, L., Smessaert, H.: Metalogical decorations of logical diagrams. Logica Universalis 10, 233–292 (2016)

[21] Demey, L., Smessaert, H.: Combinatorial bitstring semantics for arbitrary logical fragments. Journal of Philosophical Logic (2017)

[22] Dubois, D., Prade, H.: From Blanché's hexagonal organization of concepts to formal concept analysis and possibility theory. Logica Universalis 6, 149–169 (2012)

[23] Dubois, D., Prade, H.: Formal concept analysis from the standpoint of possibility theory. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Formal Concept Analysis (ICFCA 2015), pp. 21–38. LNCS 9113, Springer, Berlin (2015)

[24] Dubois, D., Prade, H.: Gradual structures of oppositions. In: Magdalena, L., Verdegay, J.L., Esteva, F. (eds.) Enric Trillas: A Passion for Fuzzy Sets, pp. 79–91. SFSC 322, Springer, Berlin (2015)

[25] Dubois, D., Prade, H., Rico, A.: The cube of opposition – a structure underlying many knowledge representation formalisms. In: Yang, Q., Wooldridge, M. (eds.) Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), pp. 2933–2939. AAAI Press, Palo Alto, CA (2015)

[26] Dubois, D., Prade, H., Rico, A.: The cube of opposition and the complete appraisal of situations by means of Sugeno integrals. In: Esposito, F., et al. (eds.) Foundations of Intelligent Systems (ISMIS 2015), pp. 197–207. LNCS 9384, Springer, Berlin (2015)

[27] Dubois, D., Prade, H., Rico, A.: Graded cubes of opposition and possibility theory with fuzzy events. International Journal of Approximate Reasoning forthcoming (2017)

[28] Dubois, D., Prade, H., Rico, A.: Organizing families of aggregation operators into a cube of opposition. In: Kacprzyk, J., Filev, D., Beliakov, G. (eds.) Granular, Soft and Fuzzy Approaches for Intelligent Systems, pp. 27–45. Springer, Berlin (2017)

[29] Fitting, M., Mendelsohn, R.L.: First-Order Modal Logic. Kluwer, Dordrecht (1998)

[30] García-Cruz, J.D.: The hypercube of dynamic opposition. In: Béziau, J.Y., Buchsbaum, A., Correia, M. (eds.) 5th World Congress on the Square of Opposition. pp. 25–27. Pontificia Universidad Católica de Chile, Santiago (2016)

[31] Gilio, A., Pfeifer, N., Sanfilippo, G.: Transitivity in coherence-based probability logic. Journal of Applied Logic 14, 46–64 (2016)

[32] Glöckner, I.: Fuzzy Quantifiers. Springer, Berlin (2006)

[33] Horn, L.R.: A Natural History of Negation. University of Chicago Press, Chicago, IL (1989)

[34] Horn, L.R.: From *if* to *iff*: Conditional perfection as pragmatic strengthening. Journal of Pragmatics 32, 289–326 (2000)

[35] Horn, L.R.: Lie-toe-tease: double negatives and unexcluded middles. Philosophical Studies 174, 79–103 (2017)

[36] Katzir, R., Singh, R.: Constraints on the lexicalization of logical operators. Linguistics and Philosophy 36, 1–29 (2013)

[37] Lenzen, W.: How to square knowledge and belief. In: Béziau, J.Y., Jacquette, D. (eds.) Around and Beyond the Square of Opposition, pp. 305–311. Springer, Basel (2012)

[38] Luzeaux, D., Sallantin, J., Dartnell, C.: Logical extensions of Aristotle's square. Logica Universalis 2, 167–187 (2008)

[39] MacCartney, B., Manning, C.: Modeling semantic containment and exclusion in natural language inference. In: Scott, D., Uszkoreit, H. (eds.) Proceedings of the 22nd International Conference on Computational Linguistics, pp. 521–528. Coling 2008, Manchester (2008)

[40] MacCartney, B., Manning, C.: Natural logic and natural language inference. In: Bunt, H., Bos, J., Pulman, S. (eds.) Computing Meaning, pp. 129–147. Springer, Dordrecht (2014)

[41] Miclet, L., Prade, H.: Analogical proportions and square of oppositions. In: Laurent, A., et al. (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems 2014, Part II, pp. 324–334. CCIS 442, Springer, Berlin (2014)

[42] Murinová, P., Novák, V.: Analysis of generalized square of opposition with intermediate quantifiers. Fuzzy Sets and Systems 242, 89–113 (2014)

[43] Murinová, P., Novák, V.: Graded generalized hexagon in fuzzy natural logic. In: Carvalho, J.P., et al. (eds.) Information Processing and Management of

Uncertainty in Knowledge-Based Systems 2016, Part II, pp. 36–47. CCIS 611, Springer, Berlin (2016)

[44] Murinová, P., Novák, V.: Syllogisms and 5-square of opposition with intermediate quantifiers in fuzzy natural logic. Logica Universalis 10, 339–357 (2016)

[45] Nelson, E.J.: The square of opposition. The Monist 42, 269–278 (1932)

[46] Parsons, T.: The traditional square of opposition. In: Zalta, E.N. (ed.) Stanford Encyclopedia of Philosophy. CSLI, Stanford, CA (2012)

[47] Pellissier, R.: Setting n-opposition. Logica Universalis 2(2), 235–263 (2008)

[48] Pfeifer, N., Sanfilippo, G.: Probabilistic squares and hexagons of opposition under coherence. International Journal of Approximate Reasoning 88, 282–294 (2017)

[49] Pfeifer, N., Sanfilippo, G.: Square of opposition under coherence. In: Ferraro, M.B., et al. (eds.) Soft Methods for Data Science, pp. 407–414. AISC 456, Springer, Berlin (2017)

[50] Pizzi, C.: Generalization and composition of modal squares of opposition. Logica Universalis 10, 313–325 (2016)

[51] Prade, H., Richard, G.: From analogical proportion to logical proportions. Logica Universalis 7, 441–505 (2013)

[52] Prade, H., Richard, G.: Picking the one that does not fit – a matter of logical proportions. In: Pasi, G., Montero, J., Ciucci, D. (eds.) Proceedings of the 8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13), pp. 392–399. Atlantis Press, Amsterdam (2013)

[53] Prade, H., Richard, G.: On different ways to be (dis)similar to elements in a set. Boolean analysis and graded extension. In: Carvalho, J.P., et al. (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems 2016, Part II, pp. 605–618. CCIS 611, Springer, Berlin (2016)

[54] Saurí, R., Pustejovsky, J.: FactBank: a corpus annotated with event factuality. Language Resources and Evaluation 43, 227–268 (2009)

[55] Saurí, R., Pustejovsky, J.: Are you sure that this happened? Assessing the factuality degree of events in text. Computational Linguistics 38, 261–299 (2012)

[56] Seuren, P., Jaspers, D.: Logico-cognitive structure in the lexicon. Language 90, 607–643 (2014)

[57] Slater, H.: Back to Aristotle! Logic and Logical Philosophy 21, 275–283 (2011)

[58] Smessaert, H.: On the 3D visualisation of logical relations. Logica Universalis 3, 303–332 (2009)

[59] Smessaert, H., Demey, L.: Logical geometries and information in the square of opposition. Journal of Logic, Language and Information 23, 527–565 (2014)

[60] Smessaert, H., Demey, L.: Béziau's contributions to the logical geometry of modalities and quantifiers. In: Koslow, A., Buchsbaum, A. (eds.) The Road to Universal Logic, pp. 475–493. Springer, Basel (2015)

[61] Smessaert, H., Demey, L.: Visualising the Boolean algebra $\mathbb{B}_4$ in 3D. In: Jamnik, M., Uesaka, Y., Elzer Schwartz, S. (eds.) Diagrammatic Representation and Inference, pp. 289–292. LNCS 9781, Springer, Berlin (2016)

[62] Smessaert, H., Demey, L.: The unreasonable effectiveness of bitstrings in logical geometry. In: Béziau, J.Y., Basti, G. (eds.) The Square of Opposition: A Cornerstone of Thought, pp. 197–214. Springer, Basel (2017)

[63] Trillas, E., Seising, R.: Turning around the ideas of 'meaning' and 'complement'. In: Collan, M., Fedrizzi, M., Kacprzyk, J. (eds.) Fuzzy Technology, pp. 3–31. SFSC 335, Springer, Berlin (2016)

[64] van Dalen, D.: Logic and Structure (Fourth Edition). Springer, Berlin (2004)

[65] Van der Auwera, J.: Modality: The three-layered scalar square. Journal of Semantics 13, 181–195 (1996)

[66] Yao, Y.: Duality in rough set theory based on the square of opposition. Fundamenta Informaticae 127, 49–64 (2013)

[67] Ziegeler, D.: On the empty O-corner of the Aristotelian square: A view from Singapore English. Journal of Pragmatics 115, 1–20 (2017)