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, after briefly discussing some key notions from logical geometry, 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 algorithm for automatically solving this task. This algorithm is naturally implemented within the paradigm of logic programming (in particular, Prolog). In order to illustrate the theoretical fruitfulness of this algorithm, 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 [38], 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 [5, 26, 32], fuzzy logic [28, 35, 36], propositional dynamic logic [11] and probabilistic logic [27, 40, 46]. Furthermore, in work on natural language, Aristotelian relations and the corresponding diagrams have been used in semantics [10, 31, 48], pragmatics [29, 30, 52] and computational linguistics [33, 34, 44, 45]. Finally, Aristotelian diagrams are also used extensively by computer scientists to study various knowledge representation formalisms, including rough set theory [6, 7, 51], formal concept analysis and possibility theory [7, 20, 21], formal argumentation theory [1, 2, 3], fuzzy set theory [8, 9, 22, 25], the theory of logical and analogical proportions

[42, 43] and multiple-criterion decision-making [23, 24]. 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. [6, 10, 20]. Dubois et al. [23] and Yao [51] 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 [15, 19] these remarks are further generalized to the applicability of Aristotelian diagrams in other areas.

In light of this trend toward more frequent and more diverse usage of Aristotelian diagrams, there is 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 [9, 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 [12, 16, 17, 19, 18]. Because of the rapidly growing complexity of Aristotelian diagrams,[1] it seems 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 algorithm for automatically solving this task. This algorithm is naturally implemented within the paradigm of logic programming (in particular, Prolog). In order to illustrate the theoretical fruitfulness of this algorithm, I will also show how it sheds new light on several well-known families of Aristotelian diagrams.

We will proceed as follows. In order to keep the paper relatively self-contained — especially for readers who are new to the framework of logical geometry —, I start in Section 2 by briefly introducing some of the fundamental notions from this framework, 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 issue of logic-sensitivity in Aristotelian diagrams. Next, Section 4 informally describes an algorithm for computing the maximal Boolean complexity of a given Aristotelian family, and Section 5 presents and discusses a Prolog implementation of this algorithm. In Section 6, then, I will illustrate the fruitfulness of this algorithm, by using it to actually compute the maximal

[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.

Boolean complexity of several well-known families 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 [16, 17]. For our current purposes it will suffice to define these 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. Note that CD_{S} can also be characterized as follows: $CD_{\mathsf{S}}(\varphi,\psi)$ iff $\varphi \equiv_{\mathsf{S}} \neg\psi$ iff $\neg\varphi \equiv_{\mathsf{S}} \psi$ [47, Lemma 3]; every formula is thus contradictory to exactly one other formula (up to logical equivalence), viz. its negation. Furthermore, it is well-known that the three other Aristotelian relations can all be characterized in terms of each other: $C_{\mathsf{S}}(\varphi,\psi)$ iff $SC_{\mathsf{S}}(\neg\varphi,\neg\psi)$ iff $SA_{\mathsf{S}}(\varphi,\neg\psi)$ [47, Lemmas 2 and 3].

An Aristotelian diagram visualizes a fragment of formulas $\mathscr{F} \subseteq \mathscr{L}_{\mathsf{S}}$, and the Aristotelian relations holding between those formulas; see Fig. 1(b-c) and Fig. 2(a-b) for some examples. We make the usual assumption that the formulas appearing in an Aristotelian diagram for $\mathscr{F}$ are S-contingent and pairwise non-S-equivalent, and that $\mathscr{F}$ itself is closed under negation.[2] More formally, an *Aristotelian diagram for* $\mathscr{F}$ *in* S is (the visualization of) an edge-labeled graph $\mathscr{G}$: the vertices of $\mathscr{G}$ are the contingent 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).

Note that $\langle \mathscr{F}, SA_{\mathsf{S}} \rangle$ is a strict partial order. Furthermore, since each formula has exactly one contradictory (viz. its negation), we can view CD_{S} as a unary function $\neg$ on $\mathscr{F}$. Recalling the interdefinability of C_{S}, SC_{S} and SA_{S}, an Aristotelian diagram can be characterized as (the visualization of) a 3-tuple $\langle \mathscr{F}, SA_{\mathsf{S}}, \neg \rangle$, where $SA_{\mathsf{S}}(\varphi,\psi)$ iff $SA_{\mathsf{S}}(\neg\psi,\neg\varphi)$ for all $\varphi,\psi \in \mathscr{F}$. Using more standard terminology from discrete mathematics, an Aristotelian diagram can be seen as (the visualization of) any 3-tuple $\langle W, <, - \rangle$ such that $\langle W, < \rangle$ is a strict partial order and $-$ is a unary function on W that satisfies $--w = w$ (i.e. $-$ is an involution), $-w \neq w$, $w \not< -w$, $-w \not< w$ and $w < v \Leftrightarrow -v < -w$ for all $w, v \in W$.

[2] So for all distinct φ, ψ that appear in an Aristotelian diagram for $\mathscr{F}$, it holds that $\not\models_{\mathsf{S}} \varphi$, $\not\models_{\mathsf{S}} \neg\varphi$, $\varphi \not\equiv_{\mathsf{S}} \psi$, and there exists a $\varphi' \in \mathscr{F}$ such that $\varphi' \equiv_{\mathsf{S}} \neg\varphi$.

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

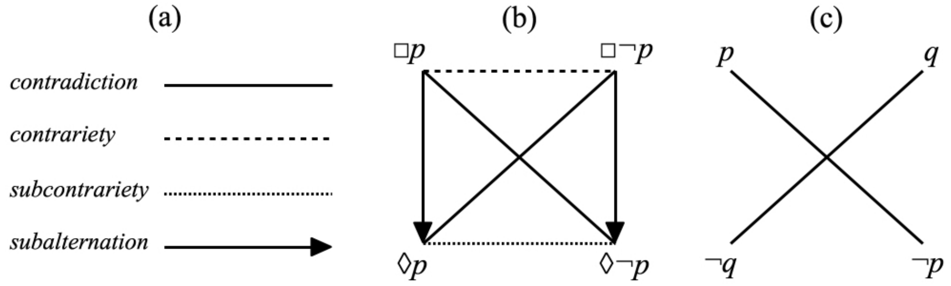
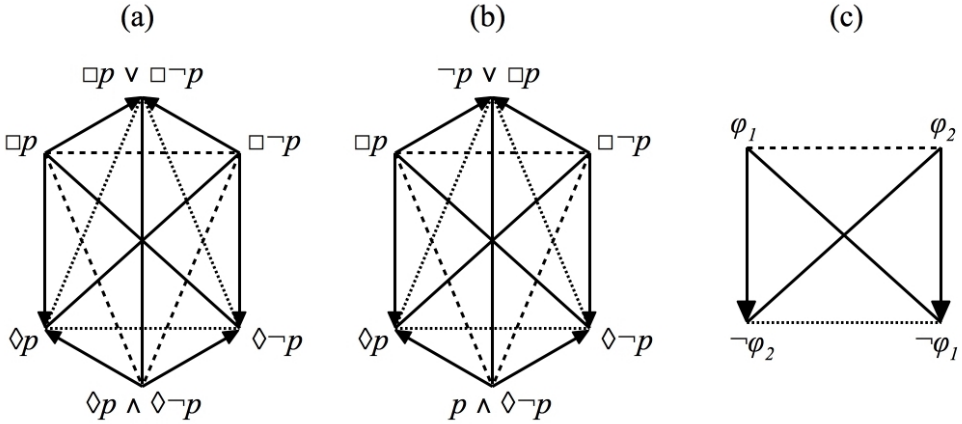


Figure 2: (a) (strong) JSB hexagon in KD, (b) (weak) JSB hexagon in KT, (c) generic description of the Aristotelian family of classical squares.



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 $\varphi, \psi \in \mathscr{F}$, it holds that $R_{\mathsf{S}}(\varphi, \psi)$ iff $R_{\mathsf{S}'}(\gamma(\varphi), \gamma(\psi))$ [19, Definition 4]. For example, the diagrams in Fig. 2(a-b) are Aristotelian isomorphic to each other.

We define a *family of Aristotelian diagrams* — or *Aristotelian family*, for short — as a maximal 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 in Fig. 1(b) belongs to the Aristotelian family of *classical squares*, that in Fig. 1(c) belongs to the family of *degenerate squares*, and those in Fig. 2(a-b) both belong to the family of *Jacoby-Sesmat-Blanché (JSB) hexagons*.

It will be useful to draw a distinction between an Aristotelian family and a 'generic' description of that family. The former is an infinite collection (a proper class, even) of concrete Aristotelian diagrams coming from all kinds of logical systems; the latter is a more abstract 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 concrete member of the Aristotelian family of classical squares, whereas Fig. 2(c) shows a generic description of that family. 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. 2(c) we find that $\mathscr{F}_{classicalsquare} = \{\varphi_1, \varphi_2, \neg\varphi_1, \neg\varphi_2\}$. Just like concrete Aristotelian diagrams, the generic description of an Aristotelian family $\mathscr{A}$ can be seen as a 3-tuple $\langle \mathscr{F}_{\mathscr{A}}, SA, \neg \rangle$, with *SA* as specified in the generic description. For example, for the family of classical squares, we have $SA(\varphi_1, \neg\varphi_2)$ and $SA(\varphi_2, \neg\varphi_1)$; cf. Fig. 2(c)

Another central notion in logical geometry is that of the Boolean closure of a given fragment or Aristotelian diagram [11, 17, 19]. Consider a finite fragment $\mathscr{F} \subseteq \mathscr{L}_{\mathsf{S}}$, and let $\mathbb{B}(\mathsf{S})$ be the Lindenbaum-Tarski algebra of S. The *Boolean closure of* $\mathscr{F}$ *in* S, denoted $\mathbb{B}_{\mathsf{S}}(\mathscr{F})$, is the smallest Boolean subalgebra of $\mathbb{B}(\mathsf{S})$ that contains $\mathscr{F}$, i.e. it is the Boolean subalgebra $\mathbb{B}$ of $\mathbb{B}(\mathsf{S})$ such that (i) $\mathscr{F} \subseteq \mathbb{B}$, and (ii) for all Boolean subalgebras $\mathbb{B}'$ of $\mathbb{B}(\mathsf{S})$ such that $\mathscr{F} \subseteq \mathbb{B}'$, it holds that $\mathbb{B} \subseteq \mathbb{B}'$.[3] 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).

In its theoretical study of Aristotelian diagrams, logical geometry makes extensive use of *bitstring semantics* [19, 49]. 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 [19]; 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

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

(where $+\varphi = \varphi$ and $-\varphi = \neg\varphi$, and where α is to be read up to S-equivalence).[4] For example, letting $\mathscr{F}$ be the set of formulas that appear in the square in Fig. 1(b), we have that $\Pi_{\mathsf{KD}}(\mathscr{F}) = \{\Box p, \Diamond p \wedge \Diamond\neg p, \Box\neg p\}$.

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,

[3] Note that the Lindenbaum-Tarski algebra $\mathbb{B}(\mathsf{S})$ is itself a Boolean algebra, because of the assumptions that we made about the logical system S.

[4] This definition of $\Pi_{\mathsf{S}}(\mathscr{F})$ applies to any fragment $\mathscr{F}$. However, if $\mathscr{F}$ is assumed to be closed under negation, as is done in this paper, then the definition of $\Pi_{\mathsf{S}}(\mathscr{F})$ can be simplified substantially; in particular, the length of the conjunctions of $\mathscr{F}$-formulas can be cut in half [19, Subsection 3.3].

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 the disjunction of those $\alpha \in \Pi_\mathsf{S}(\mathscr{F})$ such that $\models_\mathsf{S} \alpha \to \varphi$. 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 $\beta_\mathsf{S}^\mathscr{F}(\varphi) = 1010$ means that $\varphi \equiv_\mathsf{S} \alpha_1 \vee \alpha_3$.

One can show 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 $\{0,1\}^{|\Pi_\mathsf{S}(\mathscr{F})|}$ [19, Theorem 1]. The bitstring semantics of $\mathscr{F}$ (in particular, the bitstring length) 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 [19, 20, 39, 51], 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).[5]

3 Maximal Boolean Complexity

One of the main ongoing lines of research in logical geometry involves the systematic 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 [19, 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 [39].

This example shows that within Aristotelian families, there may 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 $\mathscr{A}$:

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

[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.

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 sections I will therefore describe and implement an algorithm for systematically determining the maximal Boolean complexity of any Aristotelian family $\mathscr{A}$. It is important to note that this problem from logical geometry can also be reformulated using more standard terminology from discrete mathematics. Recall that the generic description of $\mathscr{A}$ can be seen as a 3-tuple $\langle \mathscr{F}_{\mathscr{A}}, SA, \neg \rangle$. One can then show that the maximal Boolean complexity of (diagrams belonging to) $\mathscr{A}$ corresponds precisely to the number of *maximal consistent upward closed subsets* of $\langle \mathscr{F}_{\mathscr{A}}, SA, \neg \rangle$.[6] This characterization can be very helpful for determining maximal Boolean complexities, especially of smaller Aristotelian families.

The task of determining the maximal Boolean complexity of a given Aristotelian family is of considerable theoretical importance within logical geometry. 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. The largest possible partition $\Pi^{\mathscr{A}}_{max}$ and bitstring semantics $\beta^{\mathscr{A}}_{max}$ that can be induced by (diagrams belonging to) the Aristotelian family $\mathscr{A}$ also provide us with a deeper understanding of the Boolean properties of that Aristotelian family in general. This will be extensively illustrated in Section 6.

These advantages are all situated at a relatively abstract/theoretical level.[7] To provide some further context to the task of determining the maximal Boolean com-

[6] A set $G \subseteq \mathscr{F}_{\mathscr{A}}$ is said to be *upward closed* iff $\varphi \in G$ and $SA(\varphi, \psi)$ imply that $\psi \in G$. Furthermore, G is said to be *consistent* iff there exist no $\varphi, \psi \in G$ such that $SA(\varphi, \neg\psi)$. Finally, G is *maximal consistent* iff it is consistent and all its proper supersets are inconsistent. One can show that if G is maximal consistent, then it is always upward closed.

[7] This theoretical nature should not be exaggerated. Logical geometry, and the project of developing a systematic classification of Aristotelian diagrams in particular, are ultimately motivated

plexity 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 a 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.

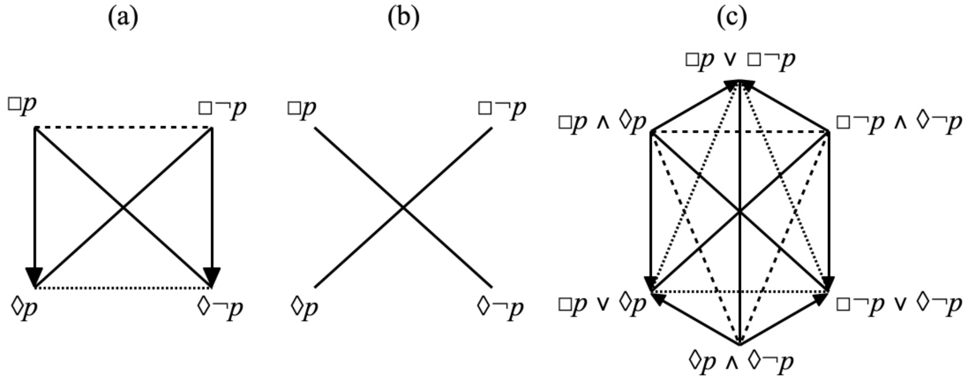
This practical problem is related to a theoretical issue that is well-known in logical geometry, viz. the fact that Aristotelian diagrams are highly *logic-sensitive* [12, 13, 19, 37, 41]. This issue has several manifestations, and the clearest one is probably the fact that the Aristotelian relation holding between two formulas partially depends on the background logic. For example, consider the fragment $\mathscr{F} = \{\Box p, \Box\neg p, \Diamond p, \Diamond\neg p\}$. One can 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, as shown in Fig. 3(b). Furthermore, $\mathscr{F}$ induces different partitions in these two logics: we have $\Pi_{\mathsf{KD}}(\mathscr{F}) = \{\Box p, \Diamond p \wedge \Diamond\neg p, \Box\neg p\}$ (and thus $\mathbb{B}_{\mathsf{KD}}(\mathscr{F}) \cong \{0,1\}^3$), but $\Pi_{\mathsf{K}}(\mathscr{F}) = \{\Box p \wedge \Diamond p, \Diamond p \wedge \Diamond\neg p, \Box\neg p \wedge \Diamond\neg p, \Box p \wedge \Box\neg p\}$ (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, we no longer refer to a specific logical system, but rather replace 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.

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 the

by the issues that appear in concrete applications of these diagrams. For example, recall Ciucci et al.'s [9] open questions about the logical properties of the Aristotelian cube that they developed for knowledge representation purposes. One of these open questions concerns the Boolean closure of their cube, or equivalently, its Boolean complexity. The algorithm described in this paper allows us to easily compute an upper bound for this value: the maximal Boolean complexity of Ciucci et al.'s cube turns out to be 7 (so its Boolean closure contains at most $2^7 - 2 = 126$ contingent elements).

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



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 $\Diamond\neg p$, that $\Box p$ is contrary to $\Box\neg p$, etc.), 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 [13].

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. 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 show that the Aristotelian relations holding among the formulas of $\mathscr{F}$ are 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: $\mathscr{F}$ 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 description 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.

[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 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.

4 Computing Maximal Boolean Complexity

Consider an arbitrary logical system S and fragment $\mathscr{F} = \{\varphi_1, \ldots, \varphi_m\} \subseteq \mathscr{L}_{\mathsf{S}}$. The fragment $\mathscr{F}$ gives rise in S to an Aristotelian diagram belonging to some Aristotelian family $\mathscr{A}$. Up to logical equivalence, the induced partition $\Pi_{\mathsf{S}}(\mathscr{F})$ 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, the partition will get smaller and smaller. We will therefore now explore the notions of S-(in)consistency 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 $\mathscr{A}$.

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:

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

[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 (cf. Footnote 8). 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.

[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.

This formula is KD-inconsistent. Furthermore, it is also JSB-inconsistent, because its first two conjuncts are KD-contrary in the JSB hexagon.

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

  This formula is also KD-inconsistent. However, it is *not* JSB-inconsistent, because it does not contain two conjuncts that are contradictory or contrary in the JSB hexagon.

We can now specify a straightforward method for computing the largest possible partition that can be induced by an Aristotelian family $\mathscr{A}$ (recall from Section 2 that $\mathscr{F}_{\mathscr{A}}$ is the fragment of formulas appearing in the generic description of $\mathscr{A}$):

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 $\mathscr{F}_{\mathscr{A}} = \{\varphi_1, \ldots, \varphi_m\}$) 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 $|\Pi^{\mathscr{A}}_{max}|$.

I will finish this section by providing pseudocode for the algorithm informally described above. The algorithm takes as input the generic description of some Aristotelian family $\mathscr{A}$, and yields as output the maximal partition $\Pi^{\mathscr{A}}_{max}$. The key idea is to build the conjunctions in $\Pi^{\mathscr{A}}_{max}$ 'conjunct by conjunct', while making sure to maintain $\mathscr{A}$-consistency (cf. the comments in the pseudocode). Because the algorithm relies heavily on the relations of contradiction and contrariety (cf. supra), we assume that the generic description of $\mathscr{A}$ is represented as a 3-tuple $\langle \mathscr{F}_{\mathscr{A}}, C, \neg \rangle$. This representation can easily be obtained from the strict partial order representation $\langle \mathscr{F}_{\mathscr{A}}, SA, \neg \rangle$ that was introduced in Section 2, by putting $C(\varphi, \psi) :\Leftrightarrow SA(\varphi, \neg\psi)$.

```
initialize Π as the empty set
foreach φ ∈ 𝓕_𝒜 do
    if Π is empty then
        add φ, ¬φ to Π
    else
        foreach conjunction γ ∈ Π do
            if γ contains a conjunct δ such that δ = ¬φ or C(δ, φ) then
                delete γ from Π                      // φ ∧ γ is 𝒜-inconsistent
            else
                replace γ with φ ∧ γ                 // φ ∧ γ is 𝒜-consistent
            end
            if γ contains a conjunct δ such that δ = φ or C(δ, ¬φ) then
                delete γ from Π                      // ¬φ ∧ γ is 𝒜-inconsistent
            else
                replace γ with ¬φ ∧ γ                // ¬φ ∧ γ is 𝒜-consistent
            end
        end
    end
end
return Π
```

5 The Prolog Implementation

I will now present a Prolog [4] implementation of the algorithm described above. 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 [19, Section 5.2] and [14, Section 5.1]. The full code of the Prolog program can be found in the online appendix to this paper.

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 and a list of the contrariety relations holding among those formulas. This essentially corresponds to the 3-tuple $\langle \mathscr{F}_{\mathscr{A}}, C, \neg \rangle$ that was discussed in the previous section. Since Aristotelian diagrams are closed under negation (which is represented by the functor `not` in the program), we specify only half of the formulas. For example, the family of classical squares is represented by the lists `[phi1,phi2]` and `[c(phi1,phi2)]`; cf. Fig. 2(c). Finally, a 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]]`.

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`. With this definition in place, it is trivial to also define the predicate `maxbooleancomplexity/3`.

The conjunctions in the partition computed by `maxpartition/3` tend to be unnecessarily long. 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, 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`. 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

I will now illustrate the theoretical fruitfulness of the proposed algorithm, by applying it(s Prolog implementation) 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. In order to keep this section relatively brief, we deal with small Aristotelian families, such as squares and hexagons. Because of their limited size, the maximal Boolean complexity of these families can also be computed 'by hand' pretty rapidly.[11] However, this does *not* mean that the Prolog program is superfluous altogether: for larger Aristotelian families (e.g. with 14 formulas or even more) — which effectively occur in the literature —, computing the maximal Boolean complexity by hand can be quite tedious, and being able to do this automatically becomes a major advantage.

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. 2(c). We can now use the Prolog program to compute the maximal partition induced by this family, and find that $\Pi^{classicalsquare}_{max} = \{\varphi_1, \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\}$. Hence, the maximal Boolean complexity of this Aristotelian family is $|\Pi^{classicalsquare}_{max}| = 3$. Furthermore, the minimal Boolean

[11] Indeed, all these computations *have* also been performed by hand. The manually obtained results systematically correspond to the algorithm's output.

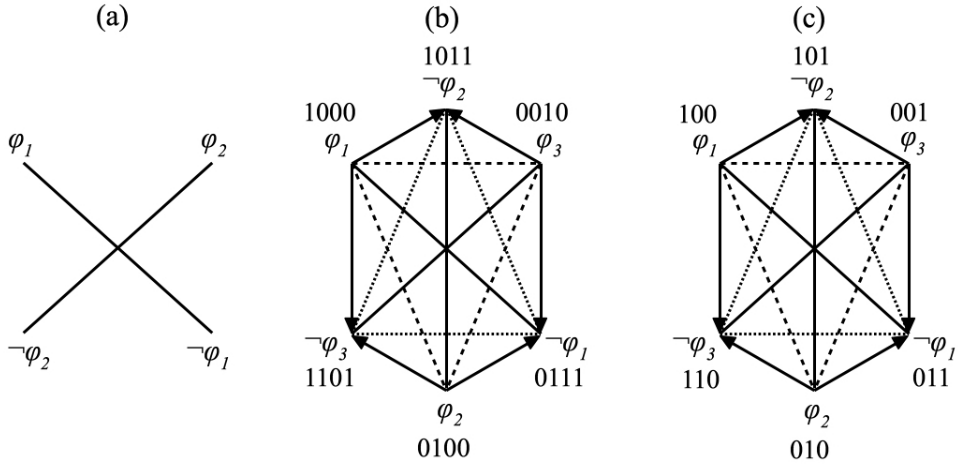
complexity of this Aristotelian family is also 3. After all, if an Aristotelian diagram has Boolean complexity 2, then it can be represented by bitstrings of length 2, and since there exist only $2^2 - 2 = 2$ contingent bitstrings of length 2, the diagram cannot be a classical square (which contains 4 contingent formulas). 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. 4(a). Using the Prolog program, we 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. that do not stand in any Aristotelian relation at all), and it is well-known that representing unconnected formulas requires bitstrings of length at least 4 [19, 49]. 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 algorithm also sheds new light on this. 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 Aristotelian family $\mathscr{A}$ of squares and logical system S, then the conjunctions in the partition induced by $\mathscr{A}$ will be not only $\mathscr{A}$-consistent, but also S-consistent.[12] This changes drastically when we move beyond the squares.

[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 stand in some Aristotelian relation in S after all.

Figure 4: (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*. Two concrete examples of this family were shown in Fig. 2(a-b). We can again use the Prolog program to compute the maximal partition induced by this family, and find that $\Pi^{JSB}_{max} = \{\varphi_1, \varphi_2, \varphi_3, \neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3\}$. Hence the maximal Boolean complexity of this Aristotelian family is $|\Pi^{JSB}_{max}| = 4$. The Prolog program also computes the maximal bitstring representation β^{JSB}_{max}; this makes use of bitstrings of length 4, as shown in Fig. 4(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. 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 not necessarily S-consistent: there exist JSB hexagons and logics for which this conjunction is consistent, but also 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 JSB family, 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 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. Consequently, the bitstrings used to represent a strong JSB hexagon are *substrings* of the bitstrings used to represent a weak JSB hexagon: they are obtained by deleting the fourth bit position, which corresponds to $\neg\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3$; cf. Fig. 4(b-c).

The correspondence between JSB hexagons and tripartitions is thus more subtle than it is sometimes taken to be [6, 20, 40, 51]. 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. However, the converse does not universally hold: not every JSB hexagon induces a tripartition. 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 an algorithm and a Prolog implementation for computing 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. Especially for the larger Aristotelian families,[13] the task of computing their maximal Boolean complexity can no longer easily be carried out by hand; this is where the Prolog program will prove its true worth.

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 an algorithm 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

[13]For example, Wessels [50] discusses Aristotelian diagrams that contain up to 16 formulas.

can create an entire pipeline of computational tools in logical geometry.

Finally, several theoretical questions regarding the maximal Boolean complexity algorithm have not been properly addressed in this paper. Is the algorithm correct? What is its time complexity? How does it compare to related algorithms, if any? These questions will also have to be left for future research.

Acknowledgements

Thanks to Hans Smessaert, Margaux Smets and two anonymous reviewers for their valuable feedback on an earlier version of this paper. Special thanks are due to one of the anonymous reviewers for his/her extensive assistance in exploring the connections with more standard parts of discrete mathematics. 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] L. Amgoud and H. Prade. Can AI models capture natural language argumentation? *International Journal of Cognitive Informatics and Natural Intelligence*, 6(3):19–32, 2012.

[2] L. Amgoud and H. Prade. Towards a logic of argumentation. In E. Hüllermeier et al., editors, *Scalable Uncertainty Management 2012*, LNCS 7520, pages 558–565. Springer, Berlin, 2012.

[3] L. Amgoud and H. Prade. 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.

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

[5] W. Carnielli and C. Pizzi. *Modalities and Multimodalities*. Springer, Dordrecht, 2008.

[6] D. Ciucci, D. Dubois, and H. Prade. 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.

[7] D. Ciucci, D. Dubois, and H. Prade. The structure of oppositions in rough set theory and formal concept analysis – toward a new bridge between the

two settings. In C. Beierle and C. Meghini, editors, *Foundations of Information and Knowledge Systems (FoIKS 2014)*, LNCS 8367, pages 154–173. Springer, Berlin, 2014.

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

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

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

[11] L. Demey. Structures of oppositions for public announcement logic. In J.-Y. Béziau and D. Jacquette, editors, *Around and Beyond the Square of Opposition*, pages 313–339. Springer, Basel, 2012.

[12] L. Demey. Interactively illustrating the context-sensitivity of Aristotelian diagrams. In H. Christiansen, I. Stojanovic, and G. Papadopoulos, editors, *Modeling and Using Context*, LNCS 9405, pages 331–345. Springer, Berlin, 2015.

[13] L. Demey. *Logic-Sensitivity in Aristotelian Diagrams*. Book manuscript, 2018.

[14] L. Demey. The logical geometry of Russell's theory of definite descriptions. *Submitted*, 2018.

[15] L. Demey. Aristotelian diagrams in the debate on future contingents. *Sophia*, forthcoming.

[16] L. Demey. Metalogic, metalanguage and logical geometry. *Logique et Analyse*, forthcoming.

[17] L. Demey and H. Smessaert. Metalogical decorations of logical diagrams. *Logica Universalis*, 10:233–292, 2016.

[18] L. Demey and H. Smessaert. Logical and geometrical distance in polyhedral Aristotelian diagrams in knowledge representation. *Symmetry*, 9(10):204, 2017.

[19] L. Demey and H. Smessaert. Combinatorial bitstring semantics for arbitrary logical fragments. *Journal of Philosophical Logic*, 47:325–363, 2018.

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

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

[22] D. Dubois and H. Prade. Gradual structures of oppositions. In L. Magdalena, J. L. Verdegay, and F. Esteva, editors, *Enric Trillas: A Passion for Fuzzy Sets*, SFSC 322, pages 79–91. Springer, Berlin, 2015.

[23] D. Dubois, H. Prade, and A. Rico. 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.

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

[25] D. Dubois, H. Prade, and A. Rico. Graded cubes of opposition and possibility theory with fuzzy events. *International Journal of Approximate Reasoning*, 84:168–185, 2017.

[26] M. Fitting and R. L. Mendelsohn. *First-Order Modal Logic*. Kluwer, Dordrecht, 1998.

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

[28] I. Glöckner. *Fuzzy Quantifiers*. Springer, Berlin, 2006.

[29] L. R. Horn. *A Natural History of Negation*. University of Chicago Press, Chicago, IL, 1989.

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

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

[32] W. Lenzen. How to square knowledge and belief. In J.-Y. Béziau and D. Jacquette, editors, *Around and Beyond the Square of Opposition*, pages 305–311. Springer, Basel, 2012.

[33] B. MacCartney and C. Manning. Modeling semantic containment and exclusion in natural language inference. In D. Scott and H. Uszkoreit, editors, *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 521–528. Coling 2008, Manchester, 2008.

[34] B. MacCartney and C. Manning. Natural logic and natural language inference. In H. Bunt, J. Bos, and S. Pulman, editors, *Computing Meaning*, pages 129–147. Springer, Dordrecht, 2014.

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

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

[37] E. J. Nelson. The square of opposition. *The Monist*, 42:269–278, 1932.

[38] T. Parsons. The traditional square of opposition. In E. N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. CSLI, Stanford, CA, 2012.

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

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

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

[42] H. Prade and G. Richard. From analogical proportion to logical proportions. *Logica Universalis*, 7:441–505, 2013.

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

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

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

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

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

[48] H. Smessaert and L. Demey. Béziau's contributions to the logical geometry of modalities and quantifiers. In A. Koslow and A. Buchsbaum, editors, *The Road to Universal Logic*, pages 475–493. Springer, Basel, 2015.

[49] H. Smessaert and L. Demey. The unreasonable effectiveness of bitstrings in logical geometry. In J.-Y. Béziau and G. Basti, editors, *The Square of Opposition: A Cornerstone of Thought*, pages 197–214. Springer, Basel, 2017.

[50] U. Wessels. *Die gute Samariterin. Zur Struktur der Supererogation*. De Gruyter, Berlin, 2002.

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

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