HELLENIC REPUBLIC

National and Kapodistrian University of Athens

Postgraduate Studies Programme "Theoretical Mathematics"

DIPLOMA THESIS

Approaches to the theory of aggregation

Sofia Kokonezi

Department of Mathematics

Supervisor:
Professor Lefteris Kirousis

Committee:
Lefteris Kirousis, Professor, Department of Mathematics, NKUA
Phokion G. Kolaitis, Distinguished Professor, Department of Computer Science and Engineering, UC Santa Cruz
Dimitrios M. Thilikos, Professor, Department of Mathematics, NKUA

November 23, 2019

Σύνοψη

Παρουσιάζουμε διάφορες επίσημες προσεγγίσεις στη θεωρία συμψηφισμού απόψεων, όπου οι όχι/ναι θέσεις μιας ομάδας ατόμων πάνω σε μια σειρά από m θέματα πρέπει να συναθροιστούν σε μια συλλογική απόφαση, και δείχνουμε ότι αυτές οι προσεγγίσεις είναι κατά μία έννοια «ισοδύναμες». Στη συνέχεια, εστιάζουμε σε δύο από αυτές τις προσεγγίσεις: *το αφαιρετικό πλαίσιο (abstract framework)* όπου το πεδίο της διαδικασίας συμψηφισμού είναι ένα υποσύνολο του $\{0,1\}^m$, το οποίο θεωρείται ότι αντιπροσωπεύει τις «λογικές» ψήφους και *το πλαίσιο που βασίζεται στους περιοριστές της ακαιρεότητας (integrity constraint based framework)*, όπου ένας τύπος της προτασιακής λογικής, που ονομάζεται *περιοριστής της ακαιρεότητας (integrity constraint)*, καθορίζει ποια διανύσματα θεωρούνται «λογικά», υπό την έννοια ότι το πεδίο της διαδικασίας συμψηφισμού είναι το σύνολο των απονομών αληθοτιμών που τον ικανοποιούν. Ενδιαφερόμαστε μόνο για διαδικασίες συμψηφισμού που διατηρούν αυτή την έννοια λογικότητας, χωρίς όμως να δίνουν όλη τη δύναμη απόφασης σε έναν μόνο ψηφοφόρο. Αυτές οι διαδικασίες ονομάζονται μη-δικτατορικοί συμψηφιστές. Παρέχουμε ικανές και αναγκαίες συνθήκες, που αφορούν στη συντακτική μορφή ενός περιοριστή ακεραιότητας, έτσι ώστε το πεδίο που περιγράφει να δέχεται έναν μη-δικτατορικό συμψηφιστή. Ονομάζουμε αυτό το είδος τύπων *δυνητικούς περιοριστές ακεραιότητας (possibility integrity constraints)*. Δείχνουμε ότι οι δυνητικοί περιοριστές ακεραιότητας είναι εύκολα αναγνωρίσιμοι και παρέχουμε αλγόριθμους οι οποίοι, δοθεντος ενός πεδίου $D \in \{0,1\}^m$, ελέγχουν σε χρόνο πολυωνυμικό στο μέγεθός του εάν δέχεται έναν μη-δικτατορικό συμψηφιστή, και παράγουν έναν δυνητικό περιοριστή ακεραιότητας που το περιγράφει, σε περίπτωση που αυτό συμβαίνει. Μελετάμε επίσης διάφορες υποκατηγορίες μη-δικτατορικών συμψηφιστών, συγκεκριμένα τοπικά μη-δικτατορικούς συμψηφιστές (locally non-dictatorial aggregators), συμψηφιστές που δεν είναι γενικευμένες δικτατορίες (not generalized dictatorships), ανωνυμικούς (anonymous), μονοτονικούς (monotone), ισχυρά δημοκρατικούς (StrongDem) και συστηματικούς συμψηφιστές (systematic aggregators). Χαρακτηρίζουμε συντακτικώς τους αντίστοιχους περιοριστές ακεραιότητας και αποδεικνύουμε ότι κάθε ένα από αυτά τα είδη περιοριστών ακεραιότητας μπορεί να αναγνωριστεί αποτελεσματικά. Τέλος, δείχνουμε ότι δοθέντος ενός πεδίου D, μπορούμε αμφότερα να ελέγξουμε αποτελεσματικά αν περιγράφεται από έναν τέτοιο τύπο και, σε περίπτωση που αυτό συμβαίνει, να τον κατασκευάσουμε.

NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

Abstract

Faculty Name
Department of Mathematics

Approaches to the theory of aggregation

by Sofia Kokonezi

We present various formal approaches to the theory of judgement aggregation, where no/yes positions of a group of individuals over a set of m issues need to be aggregated into a collective one, and show that these approaches are in a sense "equivalent". Then, we focus on two of these approaches: the *abstract framework* where the domain of the aggregation process is a subset of $\{0,1\}^m$, thought to represent the "rational" judgements and the *integrity constraint framework*, where a formula of propositional logic, called the *integrity constraint* defines which ballots are considered "rational", in the sense that the domain of the aggregation process is the set of its satisfying truth assignments. We are only interested in aggregation procedures that preserve this notion of rationality, without giving all decision power to a single voter. These procedures are called non-dictatorial aggregators. We provide necessary and sufficient conditions, regarding the syntactic type of an integrity constraint, so that the domain it describes admits a non-dictatorial aggregator. We call this type of formulas *possibility integrity constraints*. We show that possibility integrity constraints are easily recognisable and provide algorithms that, given a domain $D \subseteq \{0,1\}^m$, check in time polynomial in its size whether it admits a non-dictatorial aggregator, and actually produce a possibility integrity constraint that describes it in case it does. We also study various sub-classes of non-dictatorial aggregators, namely locally non-dictatorial aggregators, aggregators that are not generalized dictatorships, anonymous, monotone, StrongDem and systematic aggregators. We syntactically characterize the corresponding integrity constraints and show that each of these types of integrity constraints can be recognized efficiently. Finally, we show that given a domain, we can both efficiently check if it is described by such a formula and, in case it is, construct it.

Acknowledgements

I am extremely grateful to my advisor Professor Lefteris Kirousis, for his valuable guidance, helpful critique and the chance to be a part of his research project.

Also, I would like to express my sincere thanks to the other members of my thesis committee Phokion G. Kolaitis and Dimitrios M. Thilikos for generously providing their time and support, and to John Livieratos for his encouragement and our extensive conversations, that enhanced my understanding on the field of Judgement aggregation.

Lastly, I would like to express my gratefulness to my family and friends for the emotional support they offered me, not only during the writing of this work but throughout my entire experience in this postgraduate programme.

Contents

List of Tables

Chapter 1

Preliminaries

1.1 Introduction

Social choice theory is an interdisciplinary research area concerning collective decision making. In many aspects of our lives individual preferences and opinions need to be aggregated into a common and coherent thesis, conceived as the *social outcome*. This issue arises in many different contexts, ranging from juries to multi-membered courts, from legislative committees to referenda, from boards of companies to international organizations, from families and informal social groups to societies at large. These examples justify the interest shown to social choice theory by scientists from different fields, such as welfare economics, political science, voting theory, sociology, and even philosophy. Recently, social choice theory has received a lot of attention in artificial intelligence and computer science in general, since the methods of social choice have important applications in A.I., e.g. in multiagent systems.

The challenges of social decisions involving conflicting interests and opinions have been explored for a long time. As Amartya Kumar Sen [32] pointed out in his 1998 Nobel lecture, already in the fourth century B.C., Aristotle in Greece and Kautilya in India explored various ways in which individuals could take social decisions. However, the formal and systematic study of social choice theory got its own hypostasis only during the French Revolution. The pioneers of this leap were French mathematicians such as J. C. Borda (1781) and Marquis de Condorcet (1785), who addressed the problem of deriving the social decision based on the individual preferences in mathematical terms and declared the inception of social choice theory.

The first mathematical voting theory was formulated by Jean Charles de Borda. In his *Mémoire sur les élections au scrutin*, Borda showed that *plurality voting* the most common voting method, may elect the wrong candidate. Consider, for example, a voting procedure where a group of 100 people vote over 3 alternatives x, y and z. Their preferences are ordered as illustrated in table 1.1. In plurality voting, each individual votes for his first preference, thus x will be elected. However, it is easy to notice that x is the least preferable option for a majority of the voters.

	First preference	Second preference	Third preference
45 voters	x	y	z
35 voters	y	z	x
20 voters	z	y	x

TABLE 1.1: A problem with plurality voting.

In order to overcome this problem Borda proposed a voting method that was based on the preference order of the voters.

Marquis de Condorcet independently observed that plurality voting may result in the election of the wrong candidate as well. The solution he proposed was based on the pairwise comparison of alternatives, provided again that voters would give an ordering of their preferences. He also discovered a problem with *majority voting*, now known as *Condorcet's paradox*. Specifically, there are cases where circular ambiguities occur, i.e., there is no candidate who is preferred by voters to all other candidates. In such situations the group is unable to single out the "best" option. For instance, consider a hypothetical voting procedure where a group of 3 people votes over 3 alternatives, as illustrated in table 1.2. Observe that by pairwise comparison, option x prevails over y, option y over z and option z over x. In other words, this group holds an intransitive position and consequently there is no principled way of deciding the social outcome.

	First preference	Second preference	Third preference
voter 1	x	y	z
voter 2	y	z	x
voter 3	z	x	y

TABLE 1.2: An illustration of Condercet's paradox.

The same problem was studied by Charles Lutwidge Dodgson in 1874 (better known as "Lewis Carroll", author of *Alice's Adventures in Wonderland*), who contributed considerably in the development of social choice theory. Dodgson proposed several methods for the aggregation of preferences. His interest was focused on the circular ambiguities described above, which he called *majority cycles*, a term that has been used ever since.[1]

Besides the problem of selecting the winning candidate in an election, social choice theory has its origins in the normative analysis of welfare economics [31], a branch of modern economics that evaluates economic policies based on the welfare of the society at large. To this purpose measures of personal utility as well as the concept of *social welfare function* were introduced. In the 1930s, Lionel Robbins [29] raised several questions concerning the cardinality and interpersonal comparability of personal utility. Thereafter, apart from selecting the winning candidate in an election, social choice theorists also focused on defining social welfare.

This led to the construction of a list of "reasonable" axioms reflecting the desirable properties of a social welfare function. This formalization is attributed to the economist and Nobel Prize winner, John Kenneth Arrow. Concisely, these axioms are the following:

- *Universal domain*: a social welfare function has to accept as input any combination of individual preference orders.
- *Pareto condition*: whenever all members of a society rank alternative x above alternative y, then the society must also prefer x to y.
- *The independence of irrelevant alternatives*: the social preference over any two alternatives x and y must depend only on the individual preferences over those alternatives x and y (and not on other irrelevant alternatives).
- *Non-dictatorship*: requires that there exists no individual in the society such that, for any domain of the social welfare function, the collective preference is the same as that individual's preference (i.e., the dictator).

[1] For an extensive historical overview of the mathematical theory of voting, the book of D. Black is suggested [2].

In 1950, J. K. Arrow [1] proved that these conditions cannot be jointly satisfied. In particular, a dictator always occurs by solely demanding for the first three conditions to apply. This result, known as *"Arrow's impossibility theorem"*, triggered a very extensive research in the field, thus generating many other analogous results, and is generally acknowledged as the basis of modern social choice theory.

Another catalyst towards the development of modern social choice was *the discursive dilemma*, a descendant of *the doctrinal paradox*.

Doctrinal paradox

The doctrinal paradox may appear when a group of people has to form a collective decision over *logically interconnected* propositions. Such decision problems were not captured by the classical social choice theoretic models. We will illustrate this paradox through the classic example of Kornhauser and Sager [19]. Suppose that a three-member court has to rule on a breach of contract case between a plaintiff and a defendant. According to the contract law, the defendant will be sentenced liable (in what follows, this proposition is denoted by r) if and only if there was a valid contract (this proposition is denoted by p) and the defendant came in breach of it (this proposition is denoted by q). In terms of propositional logic this law is described by the formula $(p \wedge q) \leftrightarrow r$. Suppose that the three judges' views of the merits are as in table 1.3.

	Valid contract p	Breach q	Defendant liable r
Judge 1	1	1	1
Judge 2	1	0	0
Judge 3	0	1	0
Majority	1	1	0

TABLE 1.3: An illustration of the doctrinal paradox.

There are two ways in which the court can reach a verdict: either by ignoring the judges' reasoning and taking the majority vote directly on the conclusion *(conclusion-based method)* or by taking the majority vote on the premises and then deciding the outcome on r, via the rule$(p \wedge q) \leftrightarrow r$ that formalizes the contract law *(premise-based method)*. The problem is that the two methods of decision-making -on which the outcome depends- lead to different results. In particular, the defendant would be declared not liable under the conclusion-based method, whereas the defendant would be sentenced liable under the premise-based method.

Discursive dilemma

In 2001, the political philosopher Pettit [26] observed that the doctrinal paradox expresses a more extensive problem. As we have already mentioned, in the case illustrated in table 1.3 the court faces a dilemma as to which method should be chosen. In 2004, List and Pettit [22] constructed a new version of the same case where, apart from the three propositions p, q, r that respectively correspond to the issues "there is a valid contract", "there is a breach" and "the defendant is liable", they also included to the list of issues the proposition $(p \wedge q) \leftrightarrow r$, which formalizes the law. Now, suppose that the three judges cast their votes as illustrated in table 1.4.

	Valid contract p	Breach q	Legal doctrine $(p \wedge q) \leftrightarrow r$	Defendant liable r
Judge 1	1	1	1	1
Judge 2	1	0	1	0
Judge 3	0	1	1	0
Majority	1	1	1	0

TABLE 1.4: An illustration of the discursive dilemma.

Note that the judges unanimously accept the legal doctrine $(p \wedge q) \leftrightarrow r$ and that by a majority of two out of three the court also accepts the propositions $p, q, \neg r$. In other words, the court's decision includes the propositions $\{p, q, (p \wedge q) \leftrightarrow r, \neg r\}$.

The problem is that this set is *inconsistent*, as it logically implies both propositions r and $\neg r$. Indeed, both conditions for the defendant's liability are accepted, while at the same time the conclusion r is rejected and the defendant would be released. Moreover, it should be stressed out that this inconsistency appears despite the fact that each judge holds a consistent position.

Such a reconstruction of the doctrinal paradox is known as the *discursive dilemma* and, as Mongin observed, has a great consequence:

> The discursive dilemma shifts the stress away from the conflict of methods to the logical contradiction within the total set of propositions that the group accepts. [...] Trivial as this shift seems, it has far-reaching consequences, because all propositions are now being treated alike; indeed, the very distinction between premises and conclusions vanishes. This may be a questionable simplification to make in the legal context, but if one is concerned with developing a general theory, the move has clear analytical advantages. [24, p.2].

This shift prompted the formal investigation on the conditions under which consistent individual judgments may collapse into an inconsistent collective judgment, and *aggregation theory* stemmed.

The object of judgment aggregation is to study the class of *aggregators*. An n-ary aggregator is a function that on input a sequence of n feasible judgment assignments (the choices of the individuals) returns a feasible judgment assignment. The output of the aggregator is considered to be the social outcome. Intuitively, the aggregator corresponds to the "rule" by which, given the society's votes, the outcome is produced. Given a number of *agents/voters* that take positions over a set of logically interrelated propositions, the aim is to aggregate these positions in a logically consistent manner. Therefore, it stands to reason that these rules ought to satisfy some conditions. For instance, if all individuals agree on a certain issue, then this common position should be the social outcome for this issue. We could also demand something weaker: if none of the individuals supports a certain position then it cannot be the society's choice; or something stronger: if a position is embraced by the majority of the voters, then this position should be adopted by the society as well. As we have already discussed, the latter demand, which lies at the heart of democratic decisions and is in line with the desired fairness a voting procedure should have, may be rather problematic, as it affiliates with the paradoxes illustrated above.

In the remaining sections of this chapter, we present several formal approaches to the theory of judgment aggregation, for the *Boolean framework*. Namely, the *logic based framework*, where each issue corresponds to a logical formula; the *property based framework*, where an issue is thought of as a pair of properties $(H_j, H_i^{\complement})$; the *integrity* *constraint framework*, where the issues to be decided are represented by variables of a single logical formula, called the *integrity constraint*; and the *abstract framework*, where the only relative to the issues information is numerical. Furthermore, we show that these approaches are "equivalent", in the sense that there exists no aggregation problem that can be posed in one of these frameworks and not in another. However, it should be noted that each framework entails distinctive information that are usually lost while passing from one framework to another.

In Chapter 2, we proceed with some characterization results. In particular, in Section 2.1, we present a characterization result of Kirousis et al. [17] for possibility domains in the abstract framework. The results presented in Sections 2.2 and 2.3, comprise a more detailed presentation, including proofs, of the characterization results in the work of Dìaz et al. [9], for possibility domains in the integrity constraint framework.

In Chapter 3, we look into several categories of aggregation procedures that have been introduced in the field of judgment aggregation. Specifically, *locally non-dictatorial aggregators* in Section 3.1, aggregators that are not generalized dictatorships in Section 3.2, and anonymous, monotone and StrongDem aggregators in 3.3. We, furthermore, provide characterizations for the domains that are closed under each of those types of aggregation rules. Then, in Section 3.4 we study the notion of systematicity and investigate how it effects our results.

In Chapter 4, we show that the integrity constraints that characterize each type of domains mentioned above, can be recognised efficiently. Then we use this fact to prove that the identification problem for each of these types of domains can be solved in time polynomial in the size of the domain. The results presented in Chapters 3 and 4 have so far appeared only in the work of Dìaz et al. [9].

In what follows we work in the *Boolean framework*, unless explicitly mentioned otherwise.

1.2 Logic based judgment aggregation

In order to find a solution to the problem of inconsistent collective judgments that may be generated by majority voting, we have to work within a more general framework which abstracts from the specific characteristics of the decision problem in question, as well as endeavor to come up with other aggregation rules that are liberated from such inconsistencies.

In the logic based framework approach, which is the classical approach to judgment aggregation (see e.g. [15]) we start with a propositional language $\mathcal{L}$. A *judgment aggregation problem* for $\mathcal{L}$ is a tuple $\mathcal{J} = \langle N, A \rangle$ where:

- N is a finite non-empty set;
- $A \subseteq \mathcal{L}$, such that $A = \{\phi | \phi \in I\} \cup \{\neg\phi | \phi \in I\}$ for some finite $I \subseteq \mathcal{L}$ which contains only positive contingent formulas.

We denote by N the set of individuals (or *agents* or *voters*), and by I the set of *issues* to be decided; I is also called the *pre-agenda* of A. In the *Boolean framework*[2], the individuals can either accept or reject an issue in I, hence the set A -called the *agenda*- contains all possible positions toward the issues in I. The agenda A is, by definition,

[2]This model can be generalized so that instead of propositional logic, any logic satisfying some basic properties, like many-valued or modal logics, can be used.

a set of formulas closed under negation, i.e., $\forall\phi : \phi \in A$ if and only if $\neg\phi \in A$, where double negations are eliminated, i.e., $\forall\phi : \neg(\neg\phi) = \phi$.

For example, the doctrinal paradox agenda $\{p, q, p \wedge q\} \cup \{\neg p, \neg q, \neg(p \wedge q)\}$ expresses all the acceptance/rejection positions that one individual can adopt over the set of issues $\{p, q, p \wedge q\}$.

Given a judgment aggregation problem, the individuals express their opinions on the issues of the agenda. These opinions are called *judgment sets* and are defined as follows:

Definition 1 (***Judgment set***)**.** *Let $\mathcal{J} = \langle N, A\rangle$ be a judgment aggregation problem. A* judgment set *for $\mathcal{J}$ is a set of formulas $J \subseteq A$ such that:*

- *J is consistent;*
- *J is complete, i.e.,$\forall\phi \in A$, either $\phi \in J$ or $\neg\phi \in J$.*

The formulas that belong to the judgment set are the ones accepted by the individual and those that do not are the ones rejected. The demand for consistency excludes self-contradictory judgments; the demand for completeness assures that all individuals express their views on all issues posed by the agenda, and together reflect the desired "rationality" for the positions that might be adopted by an individual. In other words, these two conditions determine which subsets of A are rational positions that could potentially be held by a voter toward the issues.

The set of all judgment sets is denoted by $\mathbf{J} \subseteq \mathcal{P}(A)$, where $\mathcal{P}(\cdot)$ denotes the power-set function. A *judgment profile* $P = \langle J_i\rangle_{i\in N} \in \mathbf{J}^{|N|}$ is an $|N|$-tuple of judgment sets. By P_i we denote the i^{th} entry of P, i.e., the judgment set of agent i in P, and by $\mathbf{P}$ the set of all judgment profiles.

As we have already discussed, an *aggregation rule* should aggregate the opinions of the individuals, i.e., a profile P into a "rational" collective judgment.

Definition 2. *Let $\mathcal{J} = \langle N, A\rangle$ be a judgment aggregation problem. An* aggregation rule *for $\mathcal{J}$ is a function $f : \mathbf{P} \longrightarrow \mathcal{P}(A)$. The output set $f(P)$, where $P = \langle J_i\rangle_{i\in N}$, is sometimes denoted by J. The set J is then called a* collective set*. A collective set J which is a judgment set is called a* collective judgment set.

So, an aggregation rule takes as input any N-tuple of judgment sets (i.e. consistent and complete judgments) and outputs a subset of the agenda. As we have already seen, e.g. in the doctrinal paradox, such a subset is not necessarily consistent. In general the "rationality" of the individual judgment sets is not always transferred to the collective outcome. However, only aggregation rules that do preserve this type of rationality, i.e., rules of the form $f : \mathbf{P} \longrightarrow \mathbf{J}$, are of interest. These aggregation rules, whose output is a collective judgment set, are called *rational*.

A judgment set provides information about which propositions/issues of the agenda are rejected and which are accepted. Hence, a judgment set can be seen as a no/yes or 0/1 ballot, where the values 0 and 1 stand for rejection and acceptance, respectively. From this point of view, $\mathbf{J}$, the set of all judgment sets, is a subset of $\{0, 1\}^m$, where m is the number of issues the agenda is built on. This leads to a much simpler version of the logic based framework, where we can ignore the nature of the agenda.

More precisely, instead of the agenda, we have an m-sequence of propositional formulas, one for each issue, $\bar{\phi} = (\phi_1, \ldots, \phi_m)$. For notational convenience, if ϕ is a formula of propositional logic and $x \in \{0, 1\}$, let

$$\phi^x := \begin{cases} \phi & \text{if } x = 1, \\ \neg\phi & \text{if } x = 0. \end{cases}.$$

As the domain of an aggregation rule, we then consider the set

$$X_{\bar{\phi}} := \left\{ \bar{x} = (x_1, \ldots, x_m) \in \{0,1\}^m \mid \bigwedge_{j=1}^{m} \phi_j^{x_j} \text{ is satisfiable} \right\}.$$

Observe that, by definition, the consistency and completeness criteria are satisfied and $X_{\bar{\phi}}$ is comprised (exactly) by the "rational" ballots. Clearly, this formulation of the logic based framework has many practical advantages. In what follows, we use this formulation whenever we refer to the logic based framework.

This idea naturally leads us to the next section, where the framework presented does not rely on the context in which the judgments were collected and is focused on the judgments per se.

1.3 Abstract judgment aggregation.

In the abstract framework approach, introduced by Dokow and Holzman [10], we consider a nonempty finite set of *issues* $I = \{1, \ldots, m\}$ and a nonempty finite set of of n individuals/voters $N = \{1, \ldots, n\}$. Each voter expresses their position on every issue in I. In the Boolean framework, these positions may be of two types and are denoted by 0 and 1.

So, the set of all conceivable evaluations is $\{0,1\}^m$, where the j^{th} coordinate represents the position taken on issue j. In view of our previous discussion, one would justifiably expect some of these evaluations to be excluded. In this approach, this is determined by an *exogenously* given set $X \subseteq \{0,1\}^m$, which is called the set of *feasible voting patterns*, or simply, the *domain*. The evaluations that belong to X are called *feasible*; the rest are *infeasible*. Feasibility is an abstract property and its interpretation may vary according to the characteristics of the decision problem or the context in which it is presented .

For example, given a sequence of propositional formulas $\bar{\phi} = (\phi_1, \ldots, \phi_m)$, we can view $\phi_1, \ldots, \phi_m$ as the issues, and define $X = X_{\bar{\phi}}$, so that feasibility means logical consistency. Thus, every aggregation problem posed in the logic based framework can be expressed in the abstract framework.

Returning to the abstract approach, the domain X is taken to be an arbitrary nonempty subset of $\{0,1\}^m$. As a non-degeneracy condition, the projection of X on each issue is assumed to be $\{0,1\}$. This requirement is easily justified, as there is no reason to include an issue in a voting process, and then allow only a single position over it. Hence, this assumption is typically accepted in the field of social choice theory.

Let $n \geq 2$ be an integer representing the number of voters. The elements of $(\{0,1\}^m)^n$ can be viewed as $n \times m$ matrices whose rows correspond to voters and whose columns correspond to issues. We write x_j^i to denote the entry of the matrix in row i and column j, i.e., x_j^i represents the vote of voter i on issue j. The row vectors will be denoted as $x^1, \ldots, x^n$, and the column vectors as $x_1, \ldots, x_m$. The social vote in this framework is assumed to be extracted issue-by-issue, i.e., the social outcome on each issue does not depend on voting data on other issues. This property, though useful and desirable slightly complicates the definition of an aggregator for an abstract domain:

Definition 3. *Let $f = (f_1, \ldots, f_m)$ be an m-tuple of n-ary functions, where $f_j : \{0,1\}^n \longrightarrow \{0,1\}$, for $j \in \{1, \ldots, m\}$.*

1. *We say that f is* supportive (or conservative) *if for all $j \in \{1, \dots, m\}$ we have that*

$$\text{if } x_j = (x_j^1, \dots, x_j^n) \in \{0,1\}^n, \text{ then } f_j(x_j) = f_j(x_j^1, \dots, x_j^n) \in \{x_j^1, \dots, x_j^n\}.$$

2. *We say that f is an* aggregator *for X if it is supportive and, for all $j = \{1, \dots, m\}$ and for all $x_j \in \{0,1\}^n$, satisfies the condition bellow:*

$$\text{If } (x^1, \dots, x^n) \in X^n, \text{ then } (f_1(x_1), \dots, f_m(x_m)) \in X.$$

3. *An aggregator f is* dictatorial *on X if there is a number $d \in \{1, \dots, n\}$ such that $(f_1, \dots, f_m) = (pr_d^n, \dots, pr_d^n)$, where pr_d^n is the n-ary projection on the d^{th} coordinate. In this case we say that voter d is a* dictator.

Supportiveness reflects the justified demand for the social position over each issue to be the choice of at least one voter for this issue. An aggregator, apart from being supportive, assigns to every possible n-tuple of feasible voting patterns a social position, which is also feasible.

Let $b_1 = \ldots = b_n \in \{0,1\}$. Observe that $f = (f_1, \dots, f_m)$ is supportive if and only if each f_j is *unanimous*, i.e. $f_j(b_1, \dots, b_n) = b_1 = \ldots = b_n$, for all $j \in \{1, \dots, m\}$.

Definition 4. *A set X of feasible voting patterns is called a* possibility domain *if it has a non-dictatorial aggregator of some arity, and an* impossibility domain *otherwise.*

As we have already mentioned, a judgment aggregation problem posed in the logic based framework can be expressed in the abstract framework. The following theorem states that the reverse is also true and amounts to the *equivalence* of abstract judgment aggregation with logic based aggregation .

Theorem 1. *(Dokow and Holzman [10]) For every non-empty subset X of $\{0,1\}^m$ there exists an m-sequence of propositional formulas $\bar{\phi} = (\phi_1, \dots, \phi_m)$, so that $X_{\bar{\phi}} = X$.*

Proof. We will prove the result inductively on m, the dimension of X. In case $m = 1$ we have that $X \subseteq \{0,1\}$. Let $\bar{\phi} = \phi_1$ be the formula

$$\phi_1 := \begin{cases} y_1, & \text{if } X = \{0,1\} \\ y_1 \wedge \neg y_1, & \text{if } X = \{0\} \\ y_1 \vee \neg y_1, & \text{if } X = \{1\} \end{cases},$$

where y_1 is a Boolean variable. It can be easily established that for each of the three cases above, $\phi_1^{x_1}$ is satisfiable if and only if $x_1 \in X$, which means that $X_{\bar{\phi}} = X$. We shall thus prove the result for $m > 1$ assuming it holds for $m - 1$.

Let X be an arbitrary subset of $\{0,1\}^m$ and let X^- be its projection onto the first $m-1$ coordinates. By the inductive hypothesis applied on $X^- \subseteq \{0,1\}^{m-1}$, we get an $m-1$ sequence of propositional formulas $\bar{\psi} = (\phi_1, \dots, \phi_{m-1})$, so that $X_{\bar{\psi}} = X^-$, i.e., $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$ is satisfiable if and only if $(x_1, \dots, x_{m-1}) \in X^-$.

For notational convenience, if $\bar{x} \in \{0,1\}^{m-1}$, we denote by $\bar{x}0$ ($\bar{x}1$, respectively) the concatenation of $\bar{x}$ with (0) (with (1), respectively). Now consider the subsets of X^-:

$$\begin{aligned} X_0^- &:= \{\bar{x} \in X^- \mid \bar{x}1 \notin X\}, \\ X_1^- &:= \{\bar{x} \in X^- \mid \bar{x}0 \notin X\} \text{ and} \\ X_{01}^- &:= \{\bar{x} \in X^- \mid \bar{x}0 \in X_\phi \text{ and } \bar{x}1 \in X_\phi\}. \end{aligned}$$

Remark 1. *X_0^-, X_1^- and X_{01}^- are three (possibly empty) pairwise disjoint subsets of X^-, such that $X^- = X_0^- \cup X_1^- \cup X_{01}^-$. Moreover, $X = (X_0^- \times \{0\}) \cup (X_1^- \times \{1\}) \cup (X_{01}^- \times \{0,1\})$.*

Now, we define

$$\phi_m := \left(Z \wedge \neg \left(\bigvee_{(x_1,\ldots,x_{m-1}) \in X_0^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right) \right) \right) \vee \left(\bigvee_{(x_1,\ldots,x_{m-1}) \in X_1^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right) \right),$$

with the provision that if any of the setsX_0^-, X_1^- is empty then the corresponding part of ϕ_m is dropped, and where Z is a *new* Boolean variable. What remains to be shown is that, for an $\bar{x} = (x_1, \ldots, x_m) \in \{0,1\}^m$, $\bigwedge_{j=1}^m \phi_j^{x_j}$ is satisfiable if and only if $\bar{x} \in X$. We prove this result assuming that both X_0^-, X_1^- are non empty, since this case is the most complicated; the remaining cases easily follow as an immediate consequence of the definition of ϕ_m.

For the forward direction, let an arbitrary $\bar{x} = (x_1, \ldots, x_m) \in \{0,1\}^m$ such that $\bigwedge_{j=1}^m \phi_j^{x_j}$ is satisfiable. Then $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$ is satisfiable which, using the inductive hypothesis means that $(x_1, \ldots, x_{m-1}) \in X^-$. Let a be a truth assignment that satisfies $\bigwedge_{j=1}^m \phi_j^{x_j} = \bigwedge_{j=1}^{m-1} \phi_j^{x_j} \wedge \phi_m^{x_m}$. We will show that $(x_1, \ldots, x_m) \in X$. Keeping in mind Remark 1, we distinguish the following cases:

1. $(x_1, \ldots, x_{m-1}) \in X_0^-$. Then it suffices to show that $x_m = 0$. Assume towards a contradiction that $x_m = 1$, i.e., a satisfies ϕ_m. Observe that a does not satisfy the formula
$$\neg \left(\bigvee_{(x_1,\ldots,x_{m-1}) \in X_0^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right) \right),$$
thus, by definition of ϕ_m, there is a vector $(y_1, \ldots, y_{m-1}) \in X_1^-$ such that a satisfies $\bigwedge_{j=1}^{m-1} \phi_j^{y_j}$. Since $X_0^- \cap X_1^- = \emptyset$, there is an index i such that $y_i \neq x_i$, i.e., $\phi_i^{y_i} = \neg\phi_i^{x_i}$. Consequently, a does not satisfy $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$, which contradicts the selection of a. So, $x_m = 0$.

2. $(x_1, \ldots, x_{m-1}) \in X_1^-$. We show that $x_m = 1$ must hold. Indeed, a satisfies the formula
$$\bigvee_{(x_1,\ldots,x_{m-1}) \in X_1^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right),$$
which, by definition of ϕ_m, means that $x_m = 1$.

3. $(x_1, \ldots, x_{m-1}) \in X_{01}^-$. In this case we have to show that both $x_m = 0$ and $x_m = 1$ may hold. Recall that a satisfies $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$ and that $X_{01}^- \cap (X_0^- \cup X_1^-) = \emptyset$. So, for all $(y_1, \ldots, y_m) \in X_0^- \cup X_1^-$ there exists an index i such that $y_i \neq x_i$, i.e., $\phi_i^{y_i} = \neg\phi_i^{x_i}$. Consequently a does not satisfy either of the formulas:
$$\bigvee_{(x_1,\ldots,x_{m-1}) \in X_0^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right) \text{ and } \bigvee_{(x_1,\ldots,x_{m-1}) \in X_1^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right).$$
Thus, by taking the new variable Z to be false, we get that a does not satisfy ϕ_m, or equivalently, $x_m = 0$. On the other hand, if we take the new variable Z to be true, we have that a satisfies ϕ_m, which implies that $x_m = 1$.

For the inverse direction, we start with an $\bar{x} = (x_1, \ldots, x_m) \in X$ and we shall prove that the formula $\bigwedge_{j=1}^{m} \phi_j^{x_j}$ is satisfiable. Since $(x_1 \ldots, x_m) \in X$, we have that $(x_1 \ldots, x_{m-1}) \in X^-$ which, by the inductive hypothesis, means that $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$ is satisfiable. Let a be such a truth assignment. We will show that a also satisfies $\phi_m^{x_m}$. We distinguish the following cases:

1. If $x_m = 0$, then $(x_1, \ldots, x_{m-1}) \in X_0^- \cup X_{01}^-$. Since $(X_0^- \cup X_{01}^-) \cap X_1^- = \emptyset$, for all $(y_1, \ldots, y_m) \in X_1^-$ there exists an index i such that $y_i \neq x_i$, i.e., $\phi_i^{y_i} = \neg\phi_i^{x_i}$. Recall that a satisfies $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$, which implies that a does not satisfy the formula

$$\bigvee_{(x_1,\ldots,x_{m-1})\in X_1^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right)$$

   or, equivalently, a satisfies the formula

$$\neg \left(\bigvee_{(x_1,\ldots,x_{m-1})\in X_1^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right) \right).$$

   Thus, taking the new variable Z to be false ensures that a satisfies $\neg\phi_m = \phi_m^{x_m}$.

2. If $x_m = 1$, then $(x_1, \ldots, x_{m-1}) \in X_1^- \cup X_{01}^-$. Using the argument above and the fact that $(X_1^- \cup X_{01}^-) \cap X_0^- = \emptyset$, as well as that a satisfies $\bigwedge_{j=1}^{m-1} \phi_j^{x_j}$, we can infer that a satisfies the formula

$$\neg \left(\bigvee_{(x_1,\ldots,x_{m-1})\in X_0^-} \left(\bigwedge_{j=1}^{m-1} \phi_j^{x_j} \right) \right).$$

   In addition, taking the new variable Z to be true, we get that a satisfies $\phi_m = \phi_m^{x_m}$.

So, we have established that if $(x_1, \ldots, x_m) \in X$, then $\bigwedge_{j=1}^{m} \phi_j^{x_j}$ is satisfiable, which completes the proof of this direction. □

At this point, it should be mentioned that List and Puppe in [23, Section 2.3] give the following remark:

> There is a loss of information by moving from the logic-based framework to the abstract one.

They base their remark on the fact that a single $X \subseteq \{0,1\}^m$ may correspond to different sets of issues. These differences may concern the interpretation as well as the syntax of the formulas that comprise the issues. Indeed, the formula $\bar{\phi}$ that Theorem 1 asserts to exist is not uniquely defined, therefore the abstract framework does not offer information as to which formulas comprise the issues in the corresponding logic-based formulation.

1.4 Property based judgment aggregation.

This approach was introduced by Nehring and Puppe [25]. We are given a non-empty finite set E and a set of *properties* (unary relations) $\mathcal{H} = \{H_1, H_1^{\mathsf{C}}, \ldots, H_m, H_m^{\mathsf{C}}\}$

defined over E, i.e., each $H \in \mathcal{H}$ is a subset of E. The elements of E are called *evaluations* and a pair $(H_j, H_j^{\complement})$ is thought of as an *issue*. At a first glance, this framework is rather detached from the classic interpretation of the concepts issues, votes and voters. In order to obtain a deeper understanding of the nature of the set E, one can think of an evaluation as an individual's vote, while the evaluations that comprise a property coincide with the positions of the individuals that accept it. Since each property is a subset of evaluations in E, the issues are logically interrelated. This means that some combinations of evaluations are inconsistent. The domain of an aggregation problem is the *property space* $(E, \mathcal{H})$ and is defined as follows:

Definition 5. *A property space is a pair $(E, \mathcal{H})$, where E is a non-empty and finite set of objects ("evaluations"), and $\mathcal{H}$ is a collection of subsets of E satisfying*

1. $H \in \mathcal{H} \implies H \neq \emptyset$,
2. $H \in \mathcal{H} \implies H^{\complement} \in \mathcal{H}$,
3. *for all evaluations $x, y \in E$, if $x \neq y$ there exists $H \in \mathcal{H}$ such that $x \in H$ and $y \notin H$.*

Condition 1 is a non-degeneracy condition. Intuitively, if an evaluation $x \in H_j$, then issue j is accepted; on the other hand $x \in H_j^{\complement}$ means that issue j is rejected. Condition 2 therefore ensures that a property space is closed under negation. Condition 3 is a separation condition, in the sense that different evaluations are distinguished by at least one property. From another point of view, this implies that an evaluation is characterized by a unique set of properties in the sense that for all evaluations $x \in E$, $\{x\} = \bigcap\{H \in \mathcal{H} \mid x \in H\}$.

Definition 6. *Let $N = \{1, \ldots, n\}$ be a set of individuals with $n \geq 2$. An aggregator is a mapping $f : E^n \longrightarrow E$.*

So, given the individuals' evaluations, an *aggregator* outputs the "social" outcome that is also an evaluation. As Nehring and Puppe pointed out in [25], property spaces can be identified with particular subsets of $\{0,1\}^m$, for a suitable m, as illustrated in the next theorem. First, some notation:
If H is a property and $x \in \{0,1\}$, let

$$H^x := \begin{cases} H, & \text{if } x = 1 \\ H^{\complement}, & \text{if } x = 0 \end{cases}.$$

We then define the set

$$X_{\mathcal{H}} := \left\{ \bar{x} = (x_1, \ldots, x_m) \in \{0,1\}^m \mid \bigcap_{j=1}^{m} H_j^{x_j} \neq \emptyset \right\}.$$

Theorem 2 (Nehring and Puppe [25]). *Let $(E, \mathcal{H})$ be a property space. There is an one-to one and onto correspondence between E and $X_{\mathcal{H}}$.*

Proof. Let E be a finite non-empty set and let $\mathcal{H} = \{H_1, H_1^{\complement}, \ldots, H_m, H_m^{\complement}\}$ be a set of unary relations over E, satisfying conditions 1 to 3. For $1 \leq j \leq m$ we define the function $f_j : E \to \{0,1\}$ as follows:

$$f_j := \begin{cases} 1, & \text{if } x \in H_j \\ 0, & \text{if } x \in H_j^{\complement}. \end{cases}$$

Now, let $f : E \to \{0,1\}^m$, where $f(x) = (f_1(x), \ldots, f_m(x))$. Observe that for all $j \in \{1, \ldots, m\}$, by definition of f_j, $x \in H_j^{f_j(x)}$, i.e., $\bigcap_{j=1}^m H_j^{f_j(x)} \neq \emptyset$. Thus $f : E \to X_{\mathcal{H}}$. We will prove that f is an one-to-one and onto correspondence.

Let $x, y \in E$, with $x \neq y$. By condition 3, there exists $j \in \{1, \ldots, m\}$ such that either (i) $x \in H_j$ and $y \in H_j^{\complement}$ or (ii) $x \in H_j^{\complement}$ and $y \in H_j$. In both cases we have, by definition, that $f_j(x) \neq f_j(y)$, which implies that $f(x) \neq f(y)$. Hence, f is indeed one-to-one.

Now, we show that f is onto. Let $\bar{x} = (x_1, \ldots, x_m) \in X_{\mathcal{H}}$. By the definition of $X_{\mathcal{H}}$, we have that $\bigcap_{j=1}^m H_j^{x_j} \neq \emptyset$. So, there exists an element $y \in \bigcap_{j=1}^m H_j^{x_j} \subseteq E$. We claim that $f(y) = \bar{x} = (x_1, \ldots, x_m)$, i.e., $f_j(y) = x_j$ for all $j \in \{1, \ldots, m\}$. Assume, towards a contradiction, that there exists $j \in \{1, \ldots, m\}$ such that $f_j(y) \neq x_j$. Without loss of generality, suppose that $x_j = 1$ and $f_j(y) = 0$. By definition, we have that $y \in H_j^{\complement} = (H_j^{x_j})^{\complement}$ or, equivalently, $y \notin H_j^{x_j}$, which contradicts the fact that $y \in \bigcap_{j=1}^m H_j^{x_j} \subseteq H_j^{x_j}$. □

This allows us to identify $(E, \mathcal{H})$ with $X_{\mathcal{H}} \subseteq \{0,1\}^m$ and, therefore, an aggregation problem in the property based framework can always be expressed in the abstract framework. The next theorem proves that the property based framework is as general as the abstract one.

Theorem 3 (Nehring and Puppe [25]). *For every non-empty $X \subseteq \{0,1\}^m$, there exists a non-empty set E and a class $\mathcal{H}$ of unary relations over E, so that $X = X_{\mathcal{H}}$.*

Proof. Let $\emptyset \neq X \subseteq \{0,1\}^m$. For $1 \leq j \leq m$, we define the following unary relations over X:

$$H_j := \{\bar{x} = (x_1, \ldots, x_m) \in X \mid x_j = 1\}.$$

Note that $H_j^{\complement} = X \setminus H_j = \{\bar{x} = (x_1, \ldots, x_m) \in X \mid x_j = 0\}$.

Let $\mathcal{H} = \{H_1, H_1^{\complement}, \ldots, H_m, H_m^{\complement}\}$ be the class of these unary relations. Then $X = X_{\mathcal{H}}$, i.e.,

$$X = \left\{ \bar{x} = (x_1, \ldots, x_m) \in \{0,1\}^m \mid \bigcap_{j=1}^{m} H_j^{x_j} \neq \emptyset \right\}.$$

To establish this, let $\bar{x} = (x_1, \ldots, x_m) \in X$ be arbitrary. We claim that $\bar{x} \in H_j^{x_j}$ for all $j \in \{1, \ldots, m\}$. Indeed, by the definition of H_j, we have that if $x_j = 1$, then $\bar{x} \in H_j = H_j^{x_j}$ and if $x_j = 0$, then $\bar{x} \in H_j^{\complement} = H_j^{x_j}$. Therefore, $\bar{x} \in \bigcap_{j=1}^m H_j^{x_j}$, i.e., $\bigcap_{j=1}^m H_j^{x_j} \neq \emptyset$. So, $\bar{x} \in X_{\mathcal{H}}$. Hence, $X \subseteq X_{\mathcal{H}}$.

For the reverse inclusion, let an arbitrary $\bar{x} = (x_1, \ldots, x_m) \in X_{\mathcal{H}}$. By definition, $\bigcap_{j=1}^m H_j^{x_j} \neq \emptyset$, i.e., there exists an element $\bar{y} = (y_1, \ldots, y_m)$ such that $\bar{y} \in \bigcap_{j=1}^m H_j^{x_j} \subseteq X$. Hence, $\bar{y} \in X$ and for all $j \in \{1, \ldots, m\}$ $\bar{y} \in H_j^{x_j}$. Moreover, we have already proved that $\bar{y} \in H_j^{y_j}$ for all $j \in \{1, \ldots, m\}$. Thus, $\bar{y} \in X$ and $\bar{y} \in H_j^{x_j} \cap H_j^{y_j}$ for all $j \in \{1, \ldots, m\}$, i.e.,$\bar{y} \in X$ and $x_j = y_j$ for all $j \in \{1, \ldots, m\}$, or equivalently, $\bar{x} = \bar{y} \in X$. So, $X_{\mathcal{H}} \subseteq X$. □

Remark 2. *If we additionally assume that $X \subseteq \{0,1\}^m$ is non-degenerate, i.e., its projection on each coordinate is the set $\{0,1\}$, then for all $j \in \{1, \ldots, m\}$ we have that $H_j \neq \emptyset$ and $H_j^{\complement} \neq \emptyset$. Hence, $\mathcal{H}$ is non-degenerate as well. This ensures that Conditions 1 and 2 are satisfied. The separation Condition 3 can be trivially verified using the definition of the H_j's.*

Thus, we have established the *equivalence* between abstract and property based aggregation.

1.5 Integrity constraint based judgment aggregation

In this approach, introduced by Grandi and Endriss [14], an aggregation problem is characterized by a formula ϕ of propositional calculus on m variables $\{x_1, \ldots, x_m\}$, one for each of the *issues* in $I = \{1, \ldots, m\}$. A group of individuals $N = \{1, \ldots, n\}$ has to reach a collective decision based on the no/yes choices each individual makes regarding each issue. An individual's choice is a *ballot* in $\{0,1\}^m$ and it is considered *rational* if and only if it satisfies ϕ. Therefore, an individual's choice is identified with a truth assignment to the variables of ϕ. The truth value of a formula for an assignment is computed by the usual rules that apply to logical connectives. In what follows, the set of satisfying (returning the value 1) truth assignments, or models, of a formula ϕ, is denoted by $\text{Mod}(\phi)$. This means that the requisite rationality depends exclusively on ϕ. Hence, the formula ϕ is called an *integrity constraint*.

Definition 7. *The domain of an aggregation problem with integrity constraint the formula ϕ is $Mod(\phi)$.*

A *profile* P is a vector of rational ballots, one for each individual in N. An *aggregation procedure* is a function $f : \text{Mod}(\phi)^n \longrightarrow \{0,1\}^m$ that maps each profile to an element of $\{0,1\}^m$, considered to be the collective outcome. Clearly, the rationality of the individual choices may not be preserved in the collective level.

Definition 8. *An aggregation procedure $f : Mod(\phi)^n \longrightarrow \{0,1\}^m$ is called* collectively rational *or simply an* aggregator *for $Mod(\phi)$ if, for all profiles $P \in Mod(\phi)^n$, we have that $f(P) \in Mod(\phi)$.*

It is worth mentioning that since $\text{Mod}(\phi)$ is a subset of $\{0,1\}^m$, it can be viewed as a set of feasible evaluations X in the sense described in section 1.3. In other words, every integrity constraint aggregation problem can be expressed in the abstract framework. The next theorem is an immediate corollary of well known results (see e.g. Enderton [12, Theorem 15B]).

Theorem 4. *For any $X \subseteq \{0,1\}^m$ there is a formula ϕ of m variables such that $X = Mod(\phi)$.*

Proof. If $X = \emptyset$, let ϕ be the formula $\phi = y \wedge \neg y$ that is comprised of a single variable. Then, ϕ is not satisfiable, i.e. $\text{Mod}(\phi) = \emptyset = X$.

If $X \neq \emptyset$, let $X = \{\bar{x}_1, \ldots, \bar{x}_k\} \subseteq \{0,1\}^m$ be a numeration of its elements where, for $i \in \{1, \ldots, k\}$, we have that $\bar{x}_i = (x_{i1}, \ldots, \bar{x}_{im})$.
For $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, m\}$, let us set

$$\psi_{ij} := \begin{cases} y_j & \text{if } x_{ij} = 1 \\ \neg y_j & \text{if } x_{ij} = 0 \end{cases},$$
$$\chi_i := \psi_{i1} \wedge \ldots \wedge \psi_{im} \text{ and}$$
$$\phi := \chi_1 \vee \ldots \vee \chi_k.$$

Note that ϕ is a formula of m variables, so $\text{Mod}(\phi) \subseteq \{0,1\}^m$. We claim that $X = \text{Mod}(\phi)$.

To see this, let $\bar{x}_i = (x_{i1}, \ldots, x_{im}) \in X$. By definition, $\bar{x}_i = (x_{i1}, \ldots, x_{im})$ satisfies χ_i and consequently $\bar{x}_i$ satisfies ϕ, i.e. $\bar{x}_i \in \text{Mod}(\phi)$. Hence, $X \subseteq \text{Mod}(\phi)$.

We shall prove that $\text{Mod}(\phi) \subseteq X$ also holds. Let $a = (a_1, \ldots, a_m) \in \text{Mod}(\phi)$ be a truth assignment. Since a satisfies $\phi = \chi_1 \vee \ldots \vee \chi_k$, there is an index $i \in \{1, \ldots, k\}$ such that a satisfies $\chi_i = \psi_{i1} \wedge \ldots \wedge \psi_{im}$. Therefore, for all $j \in \{1, \ldots, m\}$, a satisfies

ψ_{ij}. This means that for all $j \in \{1, \ldots, m\}$, if $\psi_{ij} = y_j$ then $a_j = 1$, whereas if $\psi_{ij} = \neg y_j$ then $a_j = 0$. By definition of ψ_{ij}, for all $j \in \{1, \ldots, m\}$ we get that $a_j = x_{ij}$, i.e. $a = \bar{x}_i \in X$. So, $\mathrm{Mod}(\phi) \subseteq X$. □

Thus, any set of feasible voting patterns can be seen as the set of satisfying truth assignments of an integrity constraint ϕ. This result completes the *equivalence* of the integrity constraint approach with the abstract one.

Note that an integrity constraint ϕ gives rise to a possibility domain if $\mathrm{Mod}(\phi)$ admits a non-dictatorial aggregator in the sense of Definition 3.

Chapter 2

Characterization of possibility domains via integrity constraints

2.1 Possibility domains: a characterization

In this section we focus on the abstract based framework. As we have already mentioned, a set of feasible voting patterns $X \subseteq \{0,1\}^m$ is a possibility domain if it accepts a non-dictatorial aggregator of some arity. A question that arises here is the following: Given a set X of feasible voting patterns, are there any necessary and/or sufficient conditions on X that allow us to determine whether or not it is a possibility domain? In other words, is there a way to determine if X is a possibility domain by solely taking into account its form? There is a vast literature studying this question. Before illustrating some of the results, let us first present the terminology, as well as some basic remarks and examples.

Let $X \subseteq \{0,1\}^m$ be a possibility domain and let $f = (f_1, \ldots, f_m)$ be an aggregator for X. We say that f is of arity n if for all $j \in \{1, \ldots, m\}$, $f_j : \{0,1\}^n \longrightarrow \{0,1\}$, i.e. the arity of the aggregator is the arity of its component functions. As we will establish bellow, aggregators of arity two and three, called binary and ternary aggregators respectively, have a special role in the characterization of possibility domains.

Let $f = (f_1, \ldots, f_m)$ be a binary aggregator for X. Since f is by definition supportive, there are 2^2 possible binary operations for each f_j. Namely, for all $j \in \{1, \ldots, m\}$, f_j must be one of the binary operations $\vee, \wedge, \mathrm{pr}_1^2, \mathrm{pr}_2^2$, where $\vee$ and $\wedge$ are the standard binary logical operations and pr_i^2 is the binary projection on the i-th coordinate.

On the other hand, for a (supportive) ternary aggregator $f = (f_1, \ldots, f_m)$, there are 2^6 possible ternary operations for each f_j. So, we will be content to present only those that appear in the characterizations that follow, as well as those that appear in the next chapters.

Let x, y, z be three elements of $\{0,1\}$. We define the following ternary operations:

$$\vee^3(x,y,z) := x \vee y \vee z \text{ and } \wedge^3(x,y,z) := x \wedge y \wedge z,$$

$$\mathrm{maj}(x,y,z) := \begin{cases} x & \text{if } x = y \text{ or } x = z \\ y & \text{if } y = z \end{cases},$$

$$\min(x,y,z) := \begin{cases} x & \text{if } y = z \\ y & \text{if } x = z \\ z & \text{if } x = y \end{cases} \text{ and}$$

pr_i^3, the ternary projection on the i-th coordinate where $i = 1, 2, 3$.

The operations maj and min are called majority and minority operations, respectively. A minority operation coincides with the ternary direct sum mod 2, which from now on will be denoted as $\oplus$. A ternary aggregator $f = (f_1, \ldots, f_m)$ is called a *majority* aggregator if $f_j = \text{maj}$ for all $j \in \{1, \ldots, m\}$ and a *minority* aggregator if $f_j = \oplus$ for all $j \in \{1, \ldots, m\}$. An n-ary aggregator $f = (f_1, \ldots, f_m)$ is called a *projection* aggregator if for all $j \in \{1, \ldots, m\}$, $f_j = \text{pr}_i^n$ for some $i \in \{1, \ldots, n\}$. Using the well known results of Schaefer's work [30] in the field of constraint satisfaction problems and the fact that a set X of feasible voting patterns in the Boolean framework is a subset of $\{0,1\}^m$, we have that:

- X admits a majority aggregator if and only if X is a *bijunctive* logical relation, i.e. the set of satisfying assignments of a 2CNF-formula.
- X admits a minority aggregator, or equivalently X is component-wise closed under the direct sum mod 2, if and only if X is an *affine* logical relation, i.e. the set of solutions of linear equations over the two-element field.

Example 1. *The set* $X = \{(0,1,1),(1,0,0),(1,1,0)\}$ *admits the binary aggregator* $f = (\vee, \vee, \wedge)$, *as well as the majority aggregator* $g = (maj, maj, maj)$.

Since the binary aggregator f is by definition supportive, it suffices to confirm that X is closed under f. Indeed, one can easily check that for all $x, y \in X$, we have that $f(x,x) = x \in X$ and $f(x,y) = f(y,x) = (1,1,0) \in X$.

The majority aggregator g is also supportive and for the pairwise distinct $x, y, z \in X$ it holds that

$$\begin{aligned} g(x,x,x) &= x \in X, \\ g(x,x,y) &= x \in X \text{ and} \\ g(x,y,z) &= (1,1,0) \in X. \end{aligned}$$

Since permutations of the input do not affect g, we have that g is an aggregator for X.

Example 2. *The set* $X = \{(0,1,1),(1,0,0),(1,1,0),(0,0,1)\}$ *admits the minority aggregator* $f = (\oplus, \oplus, \oplus)$.

First, observe that f is supportive and that for all the pairwise distinct $x, y, z \in X$ we have that

$$\begin{aligned} f(x,x,x) &= x \in X, \\ f(x,x,y) &= y \in X \text{ and} \\ f(x,y,z) &\in X. \end{aligned}$$

Indeed,

$$f((0,1,1),(1,0,0),(1,1,0)) = (\oplus(0,1,1), \oplus(1,0,1), \oplus(1,0,0)) = (0,0,1) \in X.$$

Similarly, it holds that

$$\begin{aligned} f((0,1,1),(1,0,0),(0,0,1)) &= (1,1,0) \in X, \\ f((0,1,1),(1,1,0),(0,0,1)) &= (1,0,0) \in X \text{ and} \\ f((1,0,0),(1,1,0),(0,0,1)) &= (0,1,1) \in X. \end{aligned}$$

Since permutations of the input do not affect f, we have that f is an aggregator for X.

Example 3. *Let $X = X_1 \times X_2$ be a set of feasible voting patterns, where $X_1 \subseteq \{0,1\}^k$ and $X_2 \subseteq \{0,1\}^{m-k}$ for some $1 \leq k \leq m-1$. Then X admits a non-dictatorial projection aggregator.*

Indeed, for $n \geq 2$ and for some $d \neq d'$ so that $d, d' \in \{1, \ldots, n\}$, the n-ary projection aggregator $f = (f_1, \ldots, f_k, f_{k+1}, \ldots, f_m)$ where

$$f_j = \begin{cases} pr_d^n & \text{for } 1 \leq j \leq k \\ pr_{d'}^n & \text{for } k+1 \leq j \leq m \end{cases}$$

is an aggregator for X. To establish this, let $x_i = (x_i^1, \ldots, x_i^k, x_i^{k+1}, \ldots, x_i^m) \in X$, for all $i \in \{1, \ldots, n\}$, be arbitrary. Then $f(x_1, \ldots, x_n) = (x_d^1, \ldots, x_d^k, x_{d'}^{k+1}, \ldots, x_{d'}^m) \in X_1 \times X_2 = X$. Also, observe that since $d \neq d'$, f is not dictatorial. So, X admits a non-dictatorial projection aggregator, which in turn means that every set of feasible voting patterns that is a Cartesian product is a possibility domain.

Before delving into Dokow and Holtzman's characterization of possibility domains for the Boolean framework, there is one more notion we need to get acquainted with, namely the notion of *total blockedness*. Generally speaking, a set $X \subseteq \{0,1\}^m$ is called *totally blocked* if a certain directed graph G_X, associated with X, is strongly connected. Following Dokow and Holzman's [10] symbolism, the vertex set of G_X is the set $\{0_1, 1_1, 0_2, 1_2, \ldots, 0_m, 1_m\}$, where the vertex u_j stands for holding position u on issue j. We say that G_X is strongly connected if and only if every two distinct vertices of G_X are connected by a directed path. An alternative interpretation for this is that every two possible positions u_i, u_j on any issue are connected by a directed path denoted $u_i \longrightarrow u_j$. Intuitively, rephrasing Dokow and Holzman [10], "a set X is totally blocked if it is possible to deduce any position on any issue from any position on any issue, via a chain of deductions". This informal description of the notion of total blockedness is adequate for the purpose of this paper, thus the formal definition will be omitted.

We are now ready to state some relatively new results that provide necessary and sufficient conditions for a set X of feasible voting patterns to be a possibility domain. We remind the reader that throughout this paper X is considered to be non-degenerate.

Theorem 5. *(Dokow and Holzman [10, Theorem 2.2]) Let $X \subseteq \{0,1\}^m$ be a set of feasible voting patterns. Then, X is a possibility domain if and only if*

- *X is affine, or*
- *X is not totally blocked.*

Theorem 6. *(Dokow and Holzman [10, Claim 3.6]) Let $X \subseteq \{0,1\}^m$ be a set of feasible voting patterns. If X is not totally blocked then it is a possibility domain. In fact, for every $n \geq 2$ X admits an n-ary non-dictatorial aggregator.*

The following result provides a characterization of Boolean possibility domains that does not refer to the notion of total blockedness[1]. This result, although not explicitly stated previously, is easily derivable from Theorems 5 and 6.

Theorem 7. *(Dokow and Holzman [10]) Let $X \subseteq \{0,1\}^m$ be a set of feasible voting patterns. Then, X is a possibility domain if and only if*

[1]For a corresponding characterization of possibility domains for the non-Boolean framework, see Kirousis et al. [18].

- *X is affine, or*
- *X admits a binary non-dictatorial aggregator.*

Proof. Let $X \subseteq \{0,1\}^m$ be a set of feasible voting patterns. If X is affine or admits a binary non-dictatorial aggregator, then X is a possibility domain, by definition. Conversely, if X is a possibility domain then, by Theorem 5, X is affine or X is not totally blocked. This means that X is affine or, using Theorem 6, X admits a binary non-dictatorial aggregator. □

Theorem 7 constitutes a characterization of Boolean possibility domains. As we have already seen, a given subset of $\{0,1\}^m$ is a possibility domain if and only if it accepts a non-dictatorial aggregator of some arity. So, if we tried to determine whether a set $X \subseteq \{0,1\}^m$ is a possibility domain without using Theorem 7, the number of aggregators that should be taken into account would be exceedingly large. In fact, for each n, there are $2^{(2^n-2)\cdot m} - n$ potential non-dictatorial aggregators of arity n.

Indeed, an n-ary supportive operator $f : \{0,1\}^{m\times n} \to \{0,1\}^m$, is of the form $f = (f_1,\ldots,f_m)$, where for $j \in \{1,\ldots,m\}$, $f_j : \{0,1\}^n \to \{0,1\}$, i.e., f_j maps each element of $\{0,1\}^n$ to 0 or 1. Since we have assumed that f is supportive, it holds that $f_j(0,\ldots,0) = 0$ and $f_j(1,\ldots,1) = 1$, for all $j \in \{1,\ldots,m\}$. Thus, there are two possible selections for each element of $\{0,1\}^n \setminus \{(0,\ldots,0),(1,\ldots,1)\}$ and by extent

$$2^{(2^n-2)}$$

different selections for each f_j, for all $j \in \{1,\ldots,m\}$. So, there are

$$(2^{(2^n-2)})^m$$

possible supportive operations $f : \{0,1\}^{m\times n} \to \{0,1\}^m$. Of those, exactly n are dictatorial (one for each $d \in \{1,\ldots,n\}$, in case $f_j = \mathrm{pr}^n_d$ for all $j \in \{1,\ldots,m\}$). Hence, there exist

$$(2^{(2^n-2)})^m - n = 2^{(2^n-2)\cdot m} - n$$

potential n-ary supportive, non-dictatorial aggregators for a subset X of $\{0,1\}^m$.

Theorem 7 restricts this check to aggregators of a specific form, namely minority aggregators and non-dictatorial aggregators of arity two, instead of aggregators of any arity. Despite this improvement, deciding whether a set of feasible voting patterns is a possibility domain by a *direct* check of the aggregators indicated by Theorem 7 remains an exponential-time procedure in m, the number of issues.

Indeed, for $n = 2$, the cardinality of all potential binary non-dictatorial (supportive) aggregators is

$$2^{(2^2-2)\cdot m} - 2 = 4^m - 2.$$

However, Kirousis et al. [17, Theorem 2], using the characterization above, proved that given a set $X \subseteq \{0,1\}^m$ of feasible voting patterns, it can be determined in *time polynomial* in the size of X whether it is a possibility domain.

Theorem 8. *(Kirousis et al. [17, Theorem 2]) There is a polynomial-time algorithm for the following decision problem: Given a set $X \subseteq \{0,1\}^m$ of feasible voting patterns in the Boolean framework, determine whether or not it is a possibility domain and, if it is, produce a binary non-dictatorial aggregator for X or a minority aggregator for X.*

Therefore, the decision problem of whether a set of feasible voting patterns, in the Boolean framework[2], is a possibility domain is *tractable*. This result concerns the case where the domain $X \subseteq \{0,1\}^m$ is explicitly given.

In Section 1.5, we established that the abstract framework is equivalent with the integrity constraint based framework, in the sense that every aggregation problem posed in the abstract framework can be expressed in the integrity constraint based framework and vise versa. This brings into the picture a new question: Is there an analogous characterization for integrity constraints? Specifically, are there any sufficient and necessary conditions concerning the syntactical form of an integrity constraint ϕ, so that $\text{Mod}(\phi)$ is a possibility domain?

The purpose of sections 2.2 and 2.3 is to examine this question in detail and to progressively lead us to the answer.

2.2 Syntactic characterization of integrity constraints: Conjectures and counterexamples

In this section, we work in the integrity constraint based framework and we try to detect necessary and sufficient syntactic characteristics of a propositional formula ϕ that yields a possibility domain. This question was answered by Díaz, Kirousis, Kokonezi and Livieratos [9] in 2019. Following Díaz et al., we call this problem "syntactic characterization of integrity constraints". The aim of the present section is to analyze this problem in detail and present the conjectures that preceded the final result which will, in turn, be presented is section 2.3. Even though these conjectures have been disproved, their contribution was crucial to the uncovering of the final result and a close examination will provide a fruitful insight.

Let $D \subseteq \{0,1\}^m$ be the domain of an aggregation problem with integrity constraint the formula ϕ. Then $D = \text{Mod}(\phi)$, i.e. D is the set of all truth assignments that satisfy the formula ϕ. As we have seen (Theorem 7), a subset $D \subseteq \{0,1\}^m$ is a possibility domain if and only if it is closed under certain operators, namely minority aggregators and binary non-dictatorial aggregators. It is well known from Propositional Logic that closure properties of $D = \text{Mod}(\phi)$ may correspond to syntactic characteristics of ϕ. Our goal is to prove that D is a possibility domain if and only if it is the set of satisfying truth assignments of a formula ϕ of a specific syntactic type that is to be determined.

From now on, we focus on the integrity constraint based framework and, by extent, on formulas of propositional logic. As we have seen in Section 1.5, given an integrity constraint ϕ, we can easily acquire an aggregation problem whose domain is $\text{Mod}(\phi)$ and vise versa. Bearing on this view, we will detach ourselves from the aggregation theoretic interpretation and will approach the problem in terms of closure properties of logical relations. In other words, we no longer need to keep in mind the number of issues nor the number of voters; while the question of whether a domain D is a possibility domain will be tantamount to checking whether D is a logical relation closed under certain operators.

Let us first introduce the necessary definitions along with the notation that we will use.

We start with set of Boolean variables $V = \{x_1, \dots, x_m\}$. A *literal* is a variable $x \in V$ (positive literal) or its negation $\neg x$ (negative literal). A disjunction $(l_{i_1} \vee \cdots \vee l_{i_r})$ of literals from different variables is a *clause*. We say that a formula is in Conjunctive

[2] Kirousis et al. proved in [17, Theorem 3] that this result is valid for the non-Boolean framework as well.

Normal Form (CNF) if it is a conjunction of clauses. A formula is called 2-CNF if every clause of it contains exactly 2 literals. A clause is called *Horn* if it is a clause with at most one positive literal and *dual Horn* if it is a clause with at most one negative literal.

Definition 9. *Let ϕ be a formula in CNF.*

1. *The formula ϕ is called Horn if every clause of ϕ is a Horn clause.*
2. *The formula ϕ is called dual Horn if every clause of ϕ is a dual Horn clause.*
3. *The formula ϕ is called bijunctive if every clause of ϕ contains at most two literals.*

Example 4. *Consider the set $V = \{x_1, x_2, x_3, x_4\}$ of variables and the formulas*

$$\phi_1 = (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4),$$
$$\phi_2 = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_4), \text{ and}$$
$$\phi_3 = (x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4)$$

defined over V. It can be easily verified that the formula ϕ_1 is Horn and ϕ_2 is dual Horn, whereas the formula ϕ_3 is neither a Horn nor a dual Horn formula, since it is comprised of both Horn and dual Horn clauses. The formula ϕ_3 is bijunctive as every clause of it contains no more than two literals.

Definition 10. *A formula is called affine if it is a conjunction of sub-formulas each of which is an exclusive OR formula, i.e. a set of literals connected with $\oplus$, the exclusive OR operation.*

Generalizing the notion of a clause, exclusive OR formulas will be called clauses as well.

Example 5. *The formula $\phi_4 = (x_1 \oplus \neg x_2 \oplus x_3) \wedge (\neg x_1 \oplus \neg x_3 \oplus x_4) \wedge (x_2 \oplus x_4)$ defined over $V = \{x_1, x_2, x_3, x_4\}$ is an affine formula comprised of three (generalized) clauses.*

Bellow, we present a list of well known results of syntactic characterization problems for classes of relations with given closure properties, which are relevant to our problem.

Proposition 1. *(see Dechter and Pearl [7]) Let D be a subset of $\{0,1\}^m$. Then,*

1. *D is component-wise closed under $\wedge$ if and only if $D = Mod(\phi)$, where ϕ is a Horn formula.*
2. *D is component-wise closed under $\vee$ if and only if $D = Mod(\phi)$, where ϕ is a Dual Horn formula.*

Proposition 2 (Schaefer [30]). *Let D be a subset of $\{0,1\}^m$. Then,*

1. *D is component-wise closed under $\oplus$, the direct sum mod 2, if and only if $D = Mod(\phi)$, where ϕ is an affine formula.*
2. *D is component-wise closed under maj if and only if $D = Mod(\phi)$, where ϕ is a bijunctive formula.*

To illustrate a concrete application of the above, consider the formulas ϕ_1, ϕ_2, ϕ_3 and ϕ_4 of examples 4 and 5. The corresponding domains are

$$D_1 = \{0,1\}^4 \setminus \{(0,0,0,1),(1,0,0,1),(0,1,0,1),(1,1,0,1),(1,1,1,0),(1,1,1,1)\},$$
$$D_2 = \{(0,1,0,0),(1,1,0,0),(1,0,0,1),(1,0,1,1),(1,1,1,0),(1,1,1,1)\},$$
$$D_3 = \{(0,1,0,0),(1,0,0,1),(0,1,0,1),(1,0,1,1),(1,1,0,1)\} \text{ and}$$
$$D_4 = \{(0,0,0,1),(1,1,0,0),(0,1,1,0),(1,0,1,1)\},$$

respectively. All of the above sets are possibility domains. In particular, D_1 accepts the aggregator $f_1 = (\wedge,\wedge,\wedge,\wedge)$, as ϕ_1 is a Horn formula; D_2 accepts the aggregator $f_2 = (\vee,\vee,\vee,\vee)$, as ϕ_2 is a dual Horn formula; D_3 accepts the majority aggregator $f_3 = (maj, maj, maj, maj)$, as ϕ_3 is a bijunctive formula; and D_4 accepts the minority aggregator $f_4 = (\oplus,\oplus,\oplus,\oplus)$, as ϕ_4 is an affine formula.

In what follows, a domain $D \subseteq \{0,1\}^m$ is called Horn, dual Horn, affine or bijunctive if there is a Horn, dual Horn, affine or bijunctive formula ϕ of m variables such that $\text{Mod}(\phi) = D$, respectively. We furthermore assume, except if specifically noted, that m denotes the number of variables of a CNF formula ϕ and k the number of its clauses. Since every propositional formula can be converted into an equivalent formula that is in CNF, all formulas bellow are assumed to be in CNF, unless explicitly stated otherwise.

We are now ready to endeavor to approach the problem of syntactic characterization of integrity constraints. The starting point is Theorem 7, according to which a domain $D \subseteq \{0,1\}^m$ is a possibility domain if and only if D admits an aggregator $f = (f_1, \ldots, f_m)$, where

(i) f is a minority aggregator, or

(ii) f is a non-dictatorial binary projection aggregator, or

(iii) f is a non-projection binary aggregator.

To obtain the characterization, we work separately for each of the cases above. Recall that domains of type (i) have already been identified as models of affine formulas (Schaefer [30]). Characterizing domains of the second type is rather "easy". In fact, we show that D admits an aggregator of type (ii) if and only if there exists a formula ϕ with specific syntactic characteristics such that $D = \text{Mod}(\phi)$. We call those formulas separable. The remaining case is the most challenging and will concern us the most. In order to deal with it, we will distinguish some sub-cases depending on the type of the component functions that may comprise a non-projection binary aggregator and then (Section 2.3) evolve those results into a unified result.

We will need some additional notation. Let D be a subset of $\{0,1\}^m$. For a set of indices $I \subseteq \{1,\ldots,m\}$, let $D_I := \{(a_i)_{i \in I} \mid a \in D\}$ be the projection of D to the indices of I, and $D_{-I} := D_{\{1,\ldots,m\}\setminus I}$. Also, for two (partial) vectors $a = (a_1,\ldots,a_k) \in D_{\{1,\ldots,k\}}$, $k < m$ and $b = (b_1,\ldots,b_{m-k}) \in D_{\{k+1,\ldots,m\}}$, we define their *concatenation* to be the vector $ab = (a_1,\ldots,a_k,b_1,\ldots,b_{m-k})$. Finally, given two subsets $D, D' \subseteq \{0,1\}^m$, we write $D \approx D'$ if we can obtain D by *permuting* the coordinates of D', i.e. if $D = \{(d_{j_1},\ldots,d_{j_m}) \mid (d_1,\ldots,d_m) \in D'\}$, where $\{j_1,\ldots,j_m\} = \{1,\ldots,m\}$. Observe that in this case, if D' accepts the aggregator $f = (f_1,\ldots,f_m)$ then D accepts the aggregator $g = (f_{j_1},\ldots,f_{j_m})$, which emerges from the corresponding permutation of the component functions of f.

We begin with characterizing the domains of type (ii), i.e. domains closed under a non-dictatorial binary projection aggregator.

Definition 11. *A formula is called* separable *if its variables can be partitioned into two non-empty disjoint subsets so that no clause of it contains literals from both subsets.*

Example 6. *The formula* $\phi_5 = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_4 \vee x_5)$ *defined over* $V = \{x_1, x_2, x_3, x_4, x_5\}$ *is separable. Indeed, for the partition* $V_1 = \{x_1, x_2, x_3\}$, $V_2 = \{x_4, x_5\}$ *of* V*, we have that no clause of* ϕ_5 *contains variables from both subsets of the partition. On the other hand, there is no such partition of* $\{x_1, x_2, x_3, x_4\}$ *for neither of the formulas* ϕ_1, ϕ_2, ϕ_3 *and* ϕ_4 *of the previous examples.*

Recall that for a non-dictatorial binary projection aggregator $f = (f_1, \dots, f_m)$, we have that for $j = 1, \dots, m$, $f_j \in \{pr_1^2, pr_2^2\}$ and there exist indices $i \neq j$ such that $f_i \neq f_j$. The following lemma is an immediate consequence of the fact that an aggregator is, by definition, supportive.

Lemma 1. *A domain* D *is closed under a binary non-dictatorial projection aggregator if and only if there exists a partition* (I, J) *of* $\{1, \dots, m\}$ *such that* $D \approx D_I \times D_J$.

Proof. ($\Rightarrow$) Let $f = (f_1, \dots, f_m)$ be a binary non-dictatorial projection aggregator for D. We may assume, without loss of generality, that $f_i = \mathrm{pr}_1^2$, $i = 1, \dots, k$ and $f_j = \mathrm{pr}_2^2$, $j = k+1, \dots, m$. Let also $I := \{1, \dots, k\}$ and $J := \{k+1, \dots, m\}$. Since f is non-dictatorial we have that $k < m$, which in turn means that (I, J) is a partition of $\{1, \dots, m\}$. We will prove that $D = D_I \times D_J$. First observe that, by definition, $D \subseteq D_I \times D_J$.

For the reverse inclusion, let $a \in D_I$ and $b \in D_J$. It holds that there exist partial vectors $a' \in D_I$ and a $b' \in D_J$ such that both $ab', a'b \in D$. Since $f = (f_1, \dots, f_m)$ is an aggregator for D, $f_i = \mathrm{pr}_1^2$, $i \in I$ and $f_j = \mathrm{pr}_2^2$, $j \in J$, we have that:

$$f(ab', a'b) = ab \in D,$$

which implies that $D_I \times D_J \subseteq D$ is also true.

($\Leftarrow$) Suppose that $D \approx D_I \times D_J$, where (I, J) is a partition of $\{1, \dots, m\}$. Assume, without loss of generality, that $I = \{1, \dots, k\}$, $k < m$ and $J = \{k+1, \dots, m\}$ (thus $D = D_I \times D_J$). Let also $ab', a'b \in D$, where $a, a' \in D_I$ and $b, b' \in D_J$.

Obviously, if $f = (f_1, \dots, f_m)$ is an m-tuple of projections such that $f_i = \mathrm{pr}_1^2$, $i \in I$ and $f_j = \mathrm{pr}_2^2$, $j \in J$, then $f(ab', a'b) = ab \in D$, since $a \in D_I$ and $b \in D_J$. Thus, $f = (f_1, \dots, f_m)$ is a non-dictatorial projection aggregator for D. □

The proposition bellow realizes the syntactic characterization of possibility domains that accept a binary non-dictatorial projection aggregator.

Proposition 3. *A domain* D *admits a binary non-dictatorial projection aggregator* $(f_1, \dots, f_m)$ *if and only if there exists a separable formula* ϕ *whose set of models equals* D.

Proof. ($\Rightarrow$) Since D admits a binary non-dictatorial projection aggregator $(f_1, \dots, f_m)$, by Lemma 1, $D \approx D_I \times D_J$, where (I, J) is a partition of $\{1, \dots, m\}$ such that $I = \{i \mid f_i = \mathrm{pr}_1^2\}$ and $J = \{j \mid f_j = \mathrm{pr}_2^2\}$. Let ϕ_1 and ϕ_2 be formulas defined on $\{x_i \mid i \in I\}$ and $\{x_j \mid j \in J\}$, respectively, such that $\mathrm{Mod}(\phi_1) = D_I$ and $\mathrm{Mod}(\phi_2) = D_J$. Let also $\phi = \phi_1 \wedge \phi_2$. It is straightforward to observe that, since ϕ_1 and ϕ_2 contain no common variables,

$$\mathrm{Mod}(\phi) \approx \mathrm{Mod}(\phi_1) \times \mathrm{Mod}(\phi_2) = D_I \times D_J \approx D.$$

($\Leftarrow$) Assume that ϕ is separable and that $\text{Mod}(\phi) = D$. Since ϕ is separable, we can find a partition (I, J) of $\{1, \ldots, m\}$, a formula ϕ_1 defined on $\{x_i \mid i \in I\}$ and a formula ϕ_2 defined on $\{x_j \mid j \in J\}$, such that $\phi = \phi_1 \wedge \phi_2$. Easily, it holds that

$$D = \text{Mod}(\phi) \approx \text{Mod}(\phi_1) \times \text{Mod}(\phi_2) = D_I \times D_J.$$

The required now follows by Lemma 1. □

We now turn our attention to domains of type (iii), i.e. domains that accept a non-projection binary aggregator. Since this case is the most arduous, we will first examine certain specific sub-cases where the type of operations that may comprise a binary non projection aggregator is restricted. Recall that a binary non-projection aggregator $f = (f_1, \ldots, f_m)$ may be comprised of any combination of the operators $\wedge, \vee, \text{pr}_1^2, \text{pr}_2^2$, as long as at least one f_j is the operator $\wedge$ or $\vee$. As we have already mentioned (Proposition 1), domains closed under an aggregator $f = (f_1, \ldots, f_m)$, where $\{f_j \mid 1 \leq j \leq m\} = \{\wedge\}$ or $\{\vee\}$, have been syntactically characterized as models of Horn and dual Horn formulas, respectively. The next step is to syntactically characterize domains closed under an aggregator $f = (f_1, \ldots, f_m)$, where $\{f_j \mid 1 \leq j \leq m\} = \{\wedge, \vee\}$. We will then examine what happens if we allow some of the f_j's to be projection operations.

Definition 12. *A formula ϕ defined over the set of variables $V = \{x_1, \ldots, x_m\}$ is called renamable Horn if there is a subset $V_0 \subseteq V$ so that, if we replace every appearance of every negated literal from V_0 with the corresponding positive one and vice versa, ϕ is transformed to a Horn formula.*

The process of replacing the literals of some variables with their logical opposite ones is called a *renaming* of the variables of ϕ.

Example 7. *Consider the formulas*

$$\phi_6 = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_5)$$
$$\phi_7 = (\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_4 \vee x_5)$$

defined over $V = \{x_1, x_2, x_3, x_4, x_5\}$.

The formula ϕ_6 is renamable Horn. To see this, let $V_0 = \{x_1, x_2, x_3, x_4\}$. By renaming these variables, we get the Horn formula

$$\phi_6^* = (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_5).$$

On the other hand, it is easy to check that ϕ_7 cannot be transformed into a Horn formula for any subset of V since, for the first clause to become Horn, at least two variables from $\{x_2, x_3, x_4\}$ have to be renamed, making the second clause not Horn.

Notice that Horn and dual Horn formulas are, trivially, renamable Horn. Indeed, it suffices to take as $V_0 = \emptyset$ the empty set for Horn formulas, and $V_0 = V$ for dual Horn formulas. Thus, both formulas ϕ_1 and ϕ_2 of example 4 are renamable Horn. The formula

$$\phi_3 = (x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4)$$

defined over $V = \{x_1, x_2, x_3, x_4\}$ of the aforementioned example is renamable Horn as well. Indeed, by renaming the variables in $V_0 = \{x_1, x_4\}$ we get the Horn formula

$$\phi_3^* = (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_4).$$

Observe also, that the variables the renaming of which transforms a formula into a Horn one are not uniquely defined. For instance, for the formula ϕ_3 above, we could have chosen as V_0 the singleton $\{x_2\}$ and thus obtain the Horn formula

$$\phi_3^{*\prime} = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4).$$

As it will be established bellow (Proposition 4), renamable Horn formulas syntactically characterize domains closed under aggregators of the form $f = (f_1, \ldots, f_m)$, where $\{f_j \mid 1 \leq j \leq m\} \subseteq \{\wedge, \vee\}$. Let us mention here that a binary Boolean function $f_j : \{0,1\}^2 \to \{0,1\}$ is called *symmetric* if for all pairs of bits b_1, b_2, we have that $f_j(b_1, b_2) = f_j(b_2, b_1)$. A binary aggregator comprised exclusively of symmetric components is called symmetric. We have already seen that the only binary functions that may comprise a supportive binary aggregator are $\wedge, \vee$ and the two projection functions $\mathrm{pr}_1^2, \mathrm{pr}_2^2$. Of those four, only the first two are symmetric. Thus, the set of binary aggregators $f = (f_1, \ldots, f_m)$, where $\{f_j \mid 1 \leq j \leq m\} \subseteq \{\wedge, \vee\}$, is exactly the set of symmetric binary aggregators.

Proposition 4. *A domain D admits a symmetric binary aggregator if and only if there exists a renamable Horn formula ϕ whose set of models equals D.*

Proof. $(\Rightarrow)$ Suppose D admits the symmetric binary aggregator $f = (f_1, \ldots, f_m)$. Let ψ be a formula defined over $V = \{x_1, \ldots, x_m\}$ so that $\mathrm{Mod}(\psi) = D$. Let also $V_0 = \{x_j \in V \mid f_j = \vee\}$ and ψ^* be the formula obtained by renaming the variables of V_0 in ψ. The first step is to prove that $\mathrm{Mod}(\psi^*)$ is Horn.

For each $d = (d_1, \ldots d_m) \in D$, let $d^* = (d_1^*, \ldots, d_m^*)$ be such that

$$d_j^* = \begin{cases} 1 - d_j & \text{if } x_j \in V_0, \\ d_j & \text{if } x_j \in V \setminus V_0, \end{cases}$$

for $j = 1, \ldots, m$, and set $D^* = \{d^* \mid d \in D\}$. Observe that by renaming x_j in ψ we cause all of its literals to be satisfied by the opposite value, which implies that d^* satisfies ψ^* if and only if d satisfies ψ or, equivalently, $\mathrm{Mod}(\psi^*) = D^*$. Thus, it suffices to show that D^* admits the aggregator $g = (\wedge, \ldots, \wedge)$. The latter is an immediate consequence of the fact that D admits the aggregator $f = (f_1, \ldots, f_m)$, where

$$f_j = \begin{cases} \vee, & \text{if } x_j \in V_0, \\ \wedge, & \text{if } x_j \in V \setminus V_0, \end{cases}$$

and that $\wedge(1 - d_j, 1 - d_j') = 1 - \vee(d_j, d_j')$, for any $d, d' \in D$. Since D^* is Horn, there exists a Horn formula ϕ^* so that $\mathrm{Mod}(\phi^*) = D^*$. Let $\phi := (\phi^*)^*$ be the formula that occurs by renaming the variables of V_0 in ϕ^*. Clearly, ϕ is a renamable Horn formula. To complete the proof of this direction, it suffices to show that $D = \mathrm{Mod}(\phi)$. This follows from the observation that by renaming x_j in ϕ we cause all of its literals to be satisfied by the opposite value, which implies that d satisfies ϕ if and only if d^* satisfies ϕ^*, which in turn implies that

$$\mathrm{Mod}(\phi) = \{d \mid d^* \in \mathrm{Mod}(\phi^*)\} = \{d \mid d^* \in D^*\} = D.$$

$(\Leftarrow)$ Let ϕ be a renamable Horn formula defined over $V = \{x_1, \ldots, x_m\}$ so that $\mathrm{Mod}(\phi) = D$. By definition, there exists a subset $V_0 \subseteq V$ so that by renaming its variables in ϕ, we get the Horn formula ϕ^*. We will prove that D admits the symmetric aggregator $f = (f_1, \ldots, f_m)$, where

$$f_j = \begin{cases} \vee, & \text{if } x_j \in V_0, \\ \wedge, & \text{if } x_j \in V \setminus V_0. \end{cases}$$

Let again D^* be defined as above. Then, using the same argument, we have that $D^* = \mathrm{Mod}(\phi^*)$. The required now follows from the fact that $1 - \vee(d_j, d_j') = \wedge(1 - d_j, 1 - d_j')$, for any $d, d' \in D$ and that D^* admits the aggregator $g = (\wedge, \ldots, \wedge)$, since ϕ^* is Horn. □

It should be stressed that whenever a domain D admits a symmetric binary aggregator, following the process described in the proof of Proposition 4, we can always acquire a domain D^* that admits an aggregator $g = (g_1, \ldots, g_m)$ with $g_j = \wedge$, for all $j = 1, \ldots, m$. Reversely, given a Horn domain $D' = \text{Mod}(\phi)$, where ϕ is a Horn formula, by renaming some of the variables in ϕ, we can acquire a (renamable Horn) formula ϕ^* and in turn a domain $(D')^* = \text{Mod}(\phi^*)$ that admits a symmetric binary aggregator $f = (f_1, \ldots, f_m)$. The components of f that are $\vee$ correspond to the variables of ϕ we choose to rename. Intuitively, the process of renaming provides a way to interchange the symmetric components of an aggregator, provided that some necessary but very specific alternations of the related domain are taken into account[3].

Bearing on this view, we may temporarily set aside aggregators with component functions $\vee$, and focus on aggregators of the form $f = (f_1, \ldots, f_m)$, where $f_j \in \{\wedge, \text{pr}_1^2, \text{pr}_2^2\}$. Following the course of thought of Díaz et al., we illustrate bellow the main attempts towards the syntactical characterization of integrity constraints for domains that admit aggregators of this type. These early attempts are presented bellow as conjectures. Even though they were disproved, a close examination of the corresponding counterexamples allows us to comprehend the complexity of possibility domains and pinpoint the deviant cases. To this purpose, we first need to take a step back and try to identify, if possible, the structural features of such possibility domains as subsets of $\{0,1\}^m$.

Let $d \in \{1,2\}$ and an aggregator $f = (f_1, \ldots, f_m)$ with $\{f_j \mid 1 \leq j \leq m\} = \{\wedge, \text{pr}_d^2\}$. Naturally, the first thing tested was whether a domain D that admits f is a Cartesian product.

Conjecture 1. *[To be disproved] The following statements are equivalent:*

1. *The domain D is closed under $f = (f_1, \ldots, f_m)$ with $|\{j \mid f_j = \wedge\}| = k$ and $|\{j \mid f_j = \text{pr}_d^2\}| = m - k$.*
2. *There is a partition (I, J) of $\{1, \ldots, m\}$ where $|I| = k$, so that D_I is Horn and $D \approx D_I \times D_J$.*

Direction 2 ⇒ 1 is straightforward: For notational convenience, we assume that $I = \{1, \ldots, k\}$ and $J = \{k+1, \ldots, m\}$ and, by extent, that $D = D_I \times D_J$. We show that D admits $f = (f_1, \ldots, f_m)$ with $f_j = \wedge$ for $j \in I$ and $f_j = \text{pr}_d^2$ for $j \in J$. Let $a = (a_1, \ldots, a_m)$ and $b = (b_1, \ldots, b_m)$ be two arbitrary elements of D, with $(a_1, \ldots, a_k)$, $(b_1, \ldots, b_k) \in D_I$ and $(a_{k+1}, \ldots, a_m), (b_{k+1}, \ldots, b_m) \in D_J$. Since D_I is Horn, we have that $(a_1 \wedge b_1, \ldots, a_k \wedge b_k) \in D_I$. Moreover, $(\text{pr}_d^2(a_{k+1}, b_{k+1}), \ldots, \text{pr}_d^2(a_m, b_m)) \in D_J$ and thus $f(a,b) = (a_1 \wedge b_1, \ldots, a_k \wedge b_k, \text{pr}_d^2(a_{k+1}, b_{k+1}), \ldots, \text{pr}_d^2(a_m, b_m)) \in D_I \times D_J = D$. □

This means that whenever a domain D satisfies the conditions described in 2, it admits an aggregator of the form described in 1. However, the reverse direction does not hold, since there exist possibility domains that admit aggregators of this type that are not Cartesian products. We can establish this using the following example.

Counterexample 1. *Consider the set $D^1 = \{(0,0,0,1), (0,0,1,0), (1,1,0,1)\}$. Let $x = (0,0,0,1)$, $y = (0,0,1,0)$, $z = (1,1,0,1)$ and the aggregator $f^1 = (\wedge, \wedge, \text{pr}_1^2, \text{pr}_1^2)$. We*

[3] A formal proof of this statement is provided in Section 2.3. This intuitive approach suffices for now.

can easily establish that

$$f^1(x,x) = x, f^1(y,y) = y, f^1(z,z) = z \in D^1,$$
$$f^1(x,y) = f^1(x,z) = f^1(z,x) = f^1(z,y) = (0,0,0,1) \in D^1 \text{ and}$$
$$f^1(y,x) = f^1(y,z) = (0,0,1,0) \in D^1,$$

which means that D^1 is closed under f^1. Nonetheless, D^1 is not a Cartesian product (not even after permutation of its coordinates) as its cardinality ($|D^1| = 3$) is a prime number, and its projection on each $j \in \{1,2,3,4\}$ is the two-element set $D^1_j = \{0,1\}$.

Hence, Conjecture 1 does not hold.

In view of our discussion above, it should be clear that the structure of Cartesian products as described in Conjecture 1 is too restrictive, as it fails to characterize the domains at hand in total. Therefore, the next attempt of Díaz et al. towards the characterization of those domains proposes a less constrained condition, as illustrated in Conjecture 2.

The idea behind this conjecture is to correlate the component functions of the aggregator that are the logical operation $\wedge$ with propositional variables that carry the same indices, and then check whether the notion of Horn formulas can be somehow generalized so that it properly describes the domains at issue. Intuitively, the aim is to obtain a formula ϕ with a "Horn part" comprised of the variables of ϕ that correspond to the $\wedge$ components of the aggregator, so that $\text{Mod}(\phi) = D$. Therefore, we introduce the following notion.

Definition 13. *Let ϕ be a formula defined over the variable set V. We say that ϕ is Horn ignoring a set of variables $V' \subseteq V$, if it is Horn ignoring the variables from V' (i.e. when we delete all literals whose variable is in V').*

Observe that a formula is Horn ignoring a set of variables V' if and only if it is Horn ignoring the positive occurrences of variables from V' (i.e. when we delete all positive literals whose variable is in V').

Conjecture 2. *[To be disproved] The following statements are equivalent:*

1. *The domain D is closed under $f = (f_1, \ldots, f_m)$ with $\{f_j \mid 1 \leq j \leq m\} = \{\wedge, \text{pr}^2_d\}$ and $|\{j \mid f_j = \text{pr}^2_d\}| = k$.*
2. *There is a formula ϕ defined over $V = \{x_1, \ldots, x_m\}$ and a subset $V' \subseteq V$ of cardinality k, so that ϕ is Horn ignoring V' and $D = Mod(\phi)$.*

As it will be established bellow, the second condition of Conjecture 2 is too generic, in the sense that there are domains adherent to it while they do not admit aggregators of the form described in the first condition.

Counterexample 2. *Consider the formula $\psi = x_1 \vee x_2 \vee x_3$ and take as D^2 the set of its satisfying truth assignments, i.e. $D^2 = Mod(\psi)$. Note that the formula ψ is Horn ignoring the variables of the set $V' = \{x_2, x_3\}$.*

Now, observe that D^2 is not closed under the aggregator $f^2 = (\wedge, \text{pr}^2_1, \text{pr}^2_1)$. Indeed, we have that both $(1,0,0)$ and $(0,1,0) \in D^2$ but $f^2((1,0,0),(0,1,0)) = (0,0,0) \notin D^2$.

Due to the fact that ψ is symmetric, D^2 is not closed under any aggregator with two projection components. Therefore, Conjecture 2 does not hold.

This simple counterexample indicates that whenever variables that correspond to $\wedge$ components of the aggregator appear in the same clause with variables that correspond to projection components, things tend to become more complex. One possible way out of this problem is to find a way so that the simultaneous appearance of these two types of variables in a clause of ϕ does not result in such dysfunctional outcomes. This could be achieved by demanding that the formula ϕ satisfies some additional structural restrictions. However, every adjustment ventured at that time fell in vein, as similar counterexamples were constructed.

Then, the observations illustrated bellow gave birth to the third main conjecture.

Observation 1. *The domain $D^1 = \{(0,0,0,1),(0,0,1,0),(1,1,0,1)\}$ of Counterexample 1, apart from the aggregator $f^1 = (\wedge, \wedge, \mathrm{pr}_1^2, \mathrm{pr}_1^2)$, also admits the aggregator $g^1 = (\wedge, \wedge, \wedge, \vee)$. Indeed, for all $u \neq v \in D^1$ we have that $g^1(u,u) = u \in D^1$ and $g^1(u,v) = (0,0,0,1) \in D^1$.*

Observation 2. *Consider the formula $\psi = x_1 \vee x_2 \vee x_3$ of Counterexample 2. We established above that $D^2 = Mod(\psi)$ does not admit the aggregator $f^2 = (\wedge, \mathrm{pr}_1^2, \mathrm{pr}_1^2)$. However, $D^2 = Mod(\psi)$ is a possibility domain as it admits the symmetric aggregator $g^2 = (\vee, \vee, \vee)$. The latter holds due to the fact that ψ is a dual Horn formula.*

In other words, both D^1 and D^2 admit aggregators comprised solely of symmetric components. The question that arises here is whether this common feature could lead to useful results. So, the general idea behind Conjecture 3 is to characterize domains that admit a *binary* non-dictatorial aggregator, as models of separable formulas that satisfy an additional condition: the set of variables that correspond to the symmetric components can not be extended. Intuitively, given a domain $D \subseteq \{0,1\}^m$ that admits a binary non-dictatorial aggregator, we could consider a maximal, with respect to set inclusion, set of $\wedge/\vee$-variables V_0 and then check whether D is the Cartesian product $D \approx D_{I_0} \times D_{-I_0}$, where $I_0 = \{i \in \{1,\ldots,m\} \mid x_i \in V_0\}$. Specifically:

Conjecture 3. *[To be disproved] Let $D \subseteq \{0,1\}^m$ be a domain and $I = \{1,\ldots,m\}$ be a set of indices. The following statements are equivalent.*

1. *The domain D admits a binary non-dictatorial aggregator.*
2. *There exist pairwise disjoint subsets $I_0, I_1, I_2 \subseteq I$, some of which might be empty, whose union is I, such that:*
   - *If $I_0 = \emptyset$ then both I_1 and I_2 are non empty, and*
   - *D_{I_0} admits a binary aggregator all the components of which are symmetric functions, and*
   - *$D \approx D_{I_0} \times D_{I_1} \times D_{I_2}$.*

It should be noted here that we, intentionally, do not state the form of the aggregator explicitly. The reason we do not specify which component functions of the aggregator are $\wedge, \vee, \mathrm{pr}_1^2$ or pr_2^2 is that the structure of D (described in the second condition) is realized, if at all, only for certain aggregators, i.e. the aggregators $f = (f_1, \ldots, f_m)$ for which the set $I_0^f = \{i \in I \mid f_i \in \{\wedge, \vee\}\}$ is a maximal (with respect to set inclusion) element of the partially ordered set $\mathcal{I} = \{I_0^f \mid f \text{ is an aggregator for } D\}$. Of course, the aforementioned pertain to the argumentation for direction $1 \Rightarrow 2$, where it was intended to take as I_0 any such maximal set; the reverse direction is rather obvious.

By the discussion above, the fact that Conjecture 3 does not conflict with any of the Counterexamples 1 or 2 should not come as a surprise. Indeed, for the former we can take $I_0 = \{1,2,3,4\}$ and $I_1, I_2 = \emptyset$; then, $D^1 = D_{I_0}$ which admits the symmetric aggregator $g^1 = (\wedge, \wedge, \wedge, \vee)$. For the latter, we take $I_0 = \{1,2,3\}$ and again $I_1, I_2 = \emptyset$. Then, $D^2 = D_{I_0}$ and we have that D^2 admits the symmetric aggregator $g^2 = (\vee, \vee, \vee)$.

However, Conjecture 3 does not hold, since there exist domains that admit binary non-dictatorial aggregators while at the same time they do not satisfy the requirements of the second condition of Conjecture 3.

Counterexample 3. *Let $D^3 \subseteq \{0,1\}^5$ be the following domain, over the set of indices $I = \{1,2,3,4,5\}$:*

$$D^3 = \{(0,1,0,0,1), (1,0,0,0,1), (0,0,0,0,1), (0,0,0,1,0), (0,0,1,0,0)\}.$$

We can easily establish that D^3 is a possibility domain, as it admits the binary non-dictatorial aggregator $f^3 = (\wedge, \wedge, \mathrm{pr}_1^2, \mathrm{pr}_1^2, \mathrm{pr}_1^2)$. Indeed, for any two distinct $x = (x_1, x_2, x_3, x_4, x_5)$, $y = (y_1, y_2, y_3, y_4, y_5) \in D^3$, we have that $f^3(x, y) = (0, 0, x_3, x_4, x_5)$, which is also an element of D^3.

Now, observe that the cardinality of D^3 is a prime number ($|D^3| = 5$) and its projection to each $j \in I$ is the two element set $D^3_j = \{0,1\}$. In order to obtain a partition of I that satisfies the limitations of the second condition of Conjecture 3, we are obligated to set $I_0 = I$ and both $I_1, I_2 = \emptyset$. In this case we have that $D^3 = D_{I_0}$. The latter set, though, in opposition to Conjecture 3, does not admit a symmetric aggregator. Indeed, we have that $D^3_{\{3,4,5\}} = \{(0,0,1), (0,1,0), (1,0,0)\}$ which is closed only under dictatorial aggregators.

At this point we should be convinced that the structure of Cartesian products and, by extent, the syntactical type of separable[4] formulas, is unfit to describe the entirety of possibility domains that admit a binary non-dictatorial aggregator. This brings us back to the concept of the second conjecture. Recall that the aim is, given a domain $D \subseteq \{0,1\}^m$ that admits a binary non-dictatorial aggregator, to find a syntactic type of a formula ϕ defined over the set of variables $V = \{x_1, \dots, x_m\}$, so that $D = \mathrm{Mod}(\phi)$. And, reversely, given such a formula ϕ we want $\mathrm{Mod}(\phi)$ to admit a binary non-dictatorial aggregator. What we have established so far is that this search should be narrowed down to formulas whose clauses may simultaneously contain variables that correspond to symmetric components and variables that correspond to projection components of an aggregator. So, the efforts of Díaz et al. were thereafter concentrated to the identification of this syntactic type of formulas, being aware of this extra piece of information.

The next conjecture was also generated by Observations 1 and 2 and is in fact an improved version of Conjecture 2 in the following sense: The syntactic type proposed is an analogous generalization of renamable Horn formulas where after the renaming we acquire a formula with a generalized *maximal* "Horn part". Intuitively, the variables that constitute the "Horn part" are supposed to correspond to a *maximal* set of symmetric components of a binary non-dictatorial aggregator and the rest to projection components.

In order to be able to express this notion of maximality in terms of syntactic properties of a formula, we need the following.

Definition 14. *Let ϕ be a propositional formula defined over the set of variables V. We say that a variable $x \in V$ is positively (respectively negatively) pure if x has only positive*

[4]The notion of separable formulas here is used in a broader sense, where the different types of variables (apart from not appearing in the same clauses) may satisfy some additional conditions.

(respectively negative) appearances in ϕ. In general, positively or negatively pure variables are simply called pure.

We are now ready to state the fourth conjecture.

Conjecture 4. *[To be disproved] Let $D \subseteq \{0,1\}^m$ be a domain defined over the set of issues $I = \{1,\ldots,m\}$ and $V = \{x_1,\ldots,x_m\}$ be a set of Boolean variables. The following are equivalent.*

1. *The domain D admits a binary non-dictatorial aggregator*
2. *There exists a CNF formula ϕ on V whose set of models equals D and for which there exist pairwise disjoint subsets $V_0, V_1, V_2 \subseteq V$, with union V such that all the following conditions are satisfied:*
   (a) *For some renaming V_0^*, of the variables in V_0, every clause of the formula ϕ^* obtained from ϕ by this renaming contains at most one positive literal of a variable in V_0^*.*
   (b) *No clause of ϕ contains literals from both V_1 and V_2.*
   (c) *The formula ϕ contains no pure variables from either V_1 or V_2.*
   (d) *If $V_0 = \emptyset$ then both $V_1, V_2 \neq \emptyset$.*

Observe that we do not demand every variable of V_0 to be renamed.

Proof. (Only direction $1 \Rightarrow 2$). Assume that D admits a binary non-dictatorial aggregator. If f is any such, let I_0^f be the set of issues where the components of f are symmetric. Consider an f where I_0^f is maximal, meaning that there is no binary non-dictatorial f' with $I_0^f \subsetneq I_0^{f'}$. Let also I_1^f be the subset of $I \setminus I_0^f$ where the components of f are pr_2^1 and similarly for I_2^f. In what follows, we drop the superscript f because f remains fixed. Assume, without loss of generality, that I_0 is comprised of the issues $\{1,\ldots,k\}$, with $k \geq 0$, I_1 of the issues $\{k+1,\ldots,k+l\}$, with $l \geq 0$ and I_2 of the issues $\{k+l+1,\ldots,m\}$. Also, let V_0, V_1, V_2 be the subsets of V corresponding to I_0, I_1, I_2, respectively. Observe that $k = 0$ and $l = 0$ correspond to $V_0 = \emptyset$ and $V_1 = \emptyset$, respectively, whereas $V_2 = \emptyset$ corresponds to $k + l = m$. Since f was a non-dictatorial aggregator, we have that Condition (d) is satisfied.

For notational convenience, let D_0, D_1, D_2 be the projections of D on I_0, I_1, I_2, respectively. Obviously, D_0 admits the symmetric aggregator $g = (f_1,\ldots,f_k)$ hence, there is a formula ψ_0 on the variables V_0, such that ψ_0 can be renamed into a Horn ψ_0^* and $\mathrm{Mod}(\psi_0) = D_0$. Let $V_0^* := \{x_1^*,\ldots,x_k^*\}$ be the renaming of V_0, where

$$x_j^* = \begin{cases} \neg x_j & \text{if } f_j = \vee, \\ x_j & \text{if } f_j = \wedge, \end{cases}$$

for $1 \leq j \leq k$ and let $V^* := V_0^* \cup V_1 \cup V_2$. Let D^*, D_0^* be the corresponding to D, D_0 flipped domains, and f^* be the non-dictatorial aggregator on D^* obtained by flipping the components of f that are $\vee$ into $\wedge$.

Henceforth, if a, b are sequences of bits, then ab denotes their concatenation. Also, when no confusion could arise, we denote by the same symbol a bit and a sequence that has this bit as its only term. Finally, for two equal-length sequences of bits a, b, let $a \wedge b$ denote the sequence obtained by taking the $\wedge$ of a bit of a and a bit of b, component-wise.

The proof will be complete if we construct a formula $\psi^*_{1,2}$ on V^*, such that none of its clauses contains more than one positive literal from the variables in V^*_0, and none contains literals from both V_1 and V_2 and, moreover, the conjunction of $\psi^*_{1,2}$ with ψ^*_0 gives a formula ϕ^* such that $\text{Mod}(\phi^*) = D^*$. Observe that the Condition (c) about no pure literals of ϕ from $V_1 \cup V_2$ will be satisfied because of the maximality of I_0 (ϕ is the formula obtained from ϕ^* by giving to the variables their former name).

First, a claim:

Claim 1. *Let $u = u_0u_1u_2, u' = u'_0u'_1u'_2 \in D^*$, where the indices denote the projections of u, u' on the equally indexed, respectively, sets of issues, respectively for u, u'. Then, it holds that $(u_0 \wedge u'_0)v_1v_2 \in D^*$, where v_1 is either u_1 or u'_1 and v_2 is either u'_2 or u_2.*

The proof of the claim can be obtained by repeatedly applying f^* to u and u' in various orders.

We now continue with the proof of this direction. For each element $u \notin D^*$ whose projection on I_0 is in D^*_0, we will construct a clause c on V^* which, if conjuncted with ψ^*_0, will give a formula whose models do not contain u (for linguistic convenience, we say that c "excludes" u), but any truth assignment $u' \in D^*$ is a model of $\psi^*_0 \wedge c$ (we say that c does not exclude anything that should not be excluded). The clause c will contain at most one positive literal from V^*_0, and either will not contain literals from V_2 or c will not contain literals from V_1. The conjunction of all these clauses c will give $\psi^*_{1,2}$.

First, some notation: If a, b are sequences of bits of the same length, then we write $a \preceq b$ if at every coordinate where a is 1, so is b. If, furthermore, $a \neq b$, we write $a \prec b$.

Assume now that $u = u_0u_1u_2 \notin D^*$ with $u_0 \in D^*_0$. We start by observing that by the claim above, one of the following is true:

(i) For all truth assignments u'_2 to the variables in V_2, $u_0u_1u'_2 \notin D^*$.

(ii) For all truth assignments u'_1 to the variables in V_1, $u_0u'_1u_2 \notin D^*$.

Without loss of generality, we assume bellow that the fist is true. The following is crucial in constructing the clause c.

Claim 2. *Let $u = u_0u_1u_2 \notin D^*$ with $u_0 \in D^*_0$. If (i) holds, there exists no $u' \in D^*$ of the form $u'_0u_1u'_2$, with $u_0 \preceq u'_0$ and u'_2 an arbitrary sequence of bits that is a truth assignment of the variables in V_2.*

To establish this, assume that there exists $u' \in D^*$ of the above form. Since $u_0 \in D^*_0$, there is $w = u_0w_1w_2 \in D^*$. Then, $f^*(u', w) = u_0u_1w_2 \in D^*$, which contradicts assumption (i).

Now, we construct the clause c that excludes u but not anything that should not be excluded. Let a_0 be the conjunction of positive literals from V^*_0 corresponding to the variables where u_0 takes the value 1 and a_1 be the conjunction of literals from V_1 that is satisfied by exactly u_1. Let l_0 be a single positive literal of a variable from V^*_0, where u_0 takes the value 0 (if there is none, instead of l_0 take the empty disjunction of literals). Define c to be $a_0 \wedge a_1 \rightarrow l_0$[5]. Obviously, c excludes from D^* only sequences of the form $u'_0u_1u'_2$, for some $u'_0 \succeq u_0$, where u'_2 is some truth assignment to the variables in V_2. So, by Claim 2, no sequence that should not be excluded is excluded by c. Moreover, u is excluded, as it should. □

[5]Note that for any Boolean variables x, y, z the formula $x \wedge y \rightarrow z$ is logically equivalent to (has the same models as) the clause $\neg x \vee \neg y \vee z$.

So, we proved that whenever D admits a binary non-dictatorial aggregator, there is a formula ϕ of the form described in condition 2 so that $\text{Mod}(\phi) = D$. For the reverse direction, given a formula ϕ with $\text{Mod}(\phi) = D$ and sets V_0, V_1, V_2, so that all conditions (a) to (d) are satisfied, we want to construct a binary non-dictatorial aggregator for D. We concisely describe the purpose each of these conditions was hoped (and in some cases actually managed) to serve.

The validity of condition (a) ensures that the formula ϕ^* is Horn ignoring the set of variables $V \setminus V_0^*$, which in turn implies that the symmetric components of the (under construction) aggregator should be in a one-to-one correspondence with the variables of V_0. Note that some of the variables in V_0 may not be renamed; in fact, from the variables of V_0, those that are renamed would correspond to $\vee$ and those that are not to $\wedge$ components of the aggregator. The sets V_1 and V_2 were destined to include the variables that correspond to pr_1^2 and pr_2^2 components of the aggregator respectively. Therefore, Condition (b), imitating the definition of separable formulas that characterize domains closed under non-dictatorial projection aggregators, was expected to ensure the closure of D under the projection components. Condition (c), as we have seen, is in line with the requirement of maximality of the symmetric components of the aggregator that is adumbrated. Lastly, the purpose of Condition (d) is to ensure that the aggregator is non-dictatorial, even if it has no symmetric components.

Nonetheless, Conditions (a) to (d) failed to meet the expectations of Díaz et al., as a counterexample for direction $2 \Rightarrow 1$ was constructed.

Counterexample 4. *Let $D^4 = \{(1,0,0),(0,1,0),(0,0,1)\} \subseteq \{0,1\}^3$ be a domain defined over the set of issues $I = \{1,2,3\}$. We have already mentioned that this set is an impossibility domain, i.e it only admits dictatorial aggregators. Also, consider the formula*

$$\phi = (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_3),$$

defined over $V = \{x_1, x_2, x_3\}$ and let $V_0 = \{x_1\}$, $V_1 = \{x_2, x_3\}$ and $V_2 = \emptyset$. Since $V_2 = \emptyset$ and $V_0 \neq \emptyset$, we have that Conditions (b) and (d) are trivially satisfied. Moreover, ϕ contains no pure literals from $V_1 \cup V_2$ thus Condition (c) is satisfied as well. Now, observe that taking $x_1^* = x_1$, i.e. $V_0^* = V_0 = \{x_1\}$, every clause of the formula

$$\phi^* = \phi = (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

contains at most one positive literal from V_0^*, hence Condition (a) is also satisfied. Moreover, we can easily establish that $\text{Mod}(\phi) = D^4$, in contrast to Conjecture 4.

Let us get a more detailed review of this example in the light of the discussion that precedes it. Since $V_0 = \{x_1\}, V_1 = \{x_2, x_3\}$ and $V_2 = \emptyset$, the prospective aggregator implicitly indicated by Conjecture 4 is $f^4 = (\wedge, \text{pr}_1^2, \text{pr}_1^2)$. We have already argued that D^4 is not closed under f^4, but a closer investigation has a lot to reveal.

Consider the elements $u = (1,0,0)$ and $v = (0,1,0)$ of D^4. Since $D^4 = \text{Mod}(\phi)$, both u, v satisfy every clause of ϕ. Applying f^4 to u and v with this order, we get the element $w = (0,0,0) \notin D^4$, which satisfies every clause of ϕ except for the clause $c = (x_1 \vee x_2 \vee x_3)$. Now, observe that the variable $x_1 \in V_0$ appears negatively in every clause of ϕ that contains literals from both V_0, V_1, aside from the clause c where it appears positively. This, of course, could be just a coincidence, however if -hypothetically- x_1 appeared negatively in c as well, then ϕ would be satisfied by w.

This observation triggered the next attempt towards the characterization of possibility domains. The main idea behind it is very well captured by the following lemma.

Lemma 2. *Let ϕ be a formula in CNF defined over the variable set $V = \{x_1, \ldots, x_m\}$. If there exists a non-empty subset $V_0 \subseteq V$ so that*

- *every clause of ϕ that contains only variables from V_0 is Horn and*
- *every appearance of a variable from V_0 to the clauses that contain variables from $V \setminus V_0$ is negative,*

then $Mod(\phi)$ admits the aggregator $f = (f_1, \ldots, f_m)$, where $f_j = \begin{cases} \wedge & \text{if } x_j \in V_0, \\ pr_1^2 & \text{if } x_j \in V \setminus V_0. \end{cases}$

Proof. To enable notation, we may assume without loss of generality, that the set of indices of variables in V_0 is $I_0 = \{1, \ldots, k\}$ and we also set $D := \text{Mod}(\phi)$. In order to establish the claim above, we have to show that for two arbitrary elements $u = u_0u_1$, $v = v_0v_1 \in D$, where $u_0, v_0 \in D_{I_0}$ and $u_1, v_1 \in D_{-I_0}$ we get that $f(u, v) = (u_0 \wedge v_0)u_1 \in D$. This amounts to showing that every clause of ϕ is satisfied by $(u_0 \wedge v_0)u_1$. Let c be an arbitrary clause of ϕ. We distinguish the following cases according to the claim above:

- The clause c contains only variables from V_0. Then, by the hypothesis, c is Horn and since both u_0, v_0 satisfy c, so does $u_0 \wedge v_0$ and, by extent, $(u_0 \wedge v_0)u_1$.
- The clause c contains variables from $V \setminus V_0$. If any literal of c that corresponds to a variable of $V \setminus V_0$ is satisfied by u_1, we have nothing to prove. If there is no such literal, since u_0u_1 satisfies c, it must hold that a negative literal $\neg x_i, i \in I_0$ is satisfied by u_0. Thus, $u_0(x_i) = 0$, which means that $(u_0 \wedge v_0)(x_i) = 0$ as well. Consequently, c is satisfied by $(u_0 \wedge v_0)u_1$.

Since the clause c was arbitrary, we have that $D = \text{Mod}(\phi)$ is indeed closed under f. □

So, roughly speaking, the syntactic type we are looking for ought to include formulas in CNF that (possibly after some renaming) have a "Horn part" so that the variables that comprise it appear only negatively (if at all) in the rest of the clauses. Furthermore, this syntactic type should be liberated from the prohibition of the appearance of pure literals of variables that correspond to projection components; this is because this requirement was added in view of the maximality requirement, which appears to fail ruling out impossibility domains.

2.3 Syntactic characterization of integrity constraints

In this section, we present the answer to the problem of the syntactic characterization of possibility domains, proposed by Díaz et al. [9]. In order to address this problem, by Theorem 7, it suffices to characterize domains that admit (i) a minority aggregator, or (ii) a non-dictatorial binary projection aggregator or (iii) a binary non-projection aggregator. So far we have established that domains of types (i) and (ii) are characterized as the sets of models of *affine* formulas and *separable* formulas, respectively. We have also acquired a rough sketch of the syntactic type of formulas that characterize the domains of type (iii). Following Díaz et al. we call these formulas *renamable partially Horn*. In short, a renamable partially Horn formula, is a formula such that if we change the logical sign of some of its variables, we get a formula that has a Horn part and whose remaining clauses contain only negative occurrences of the variables in the Horn part.

We first introduce the necessary definitions and examples.

Definition 15. *A formula ϕ defined over the variable set V is called partially Horn if there is a nonempty subset $V_0 \subseteq V$ such that*

(i) the clauses containing only variables from V_0 are Horn and

(ii) the variables of V_0 appear only negatively (if at all) in a clause containing also variables not in V_0.

If a formula ϕ is partially Horn, then any non-empty subset $V_0 \subseteq V$ that satisfies the requirements of Definition 15 will be called an *admissible set of variables*. Also the Horn clauses that contain variables only from V_0 will be called *admissible clauses*. It should be stressed out that the set of admissible clauses might be empty. A Horn clause with a variable in $V \setminus V_0$ will be called *inadmissible*. The reason for the possible existence of such clauses will be made clear in the following example, but let as briefly state that this discrimination between the Horn clauses is derived from the requirement for only negative occurrences of the admissible variables in the non-admissible clauses.

Observe that a Horn formula is, trivially, partially Horn too, as the set of its variables is itself an admissible set of variables. Also, a formula that contains at least one negative pure literal, $\neg x_i$, is partially Horn. Indeed, by setting V_0 to be the singleton $\{x_i\}$ the requirements of Definition 15 are trivially satisfied.

Example 8. *We first examine the formulas of the previous examples. Both formulas*

$$\begin{aligned}\phi_6 &= (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_5) \text{ and}\\ \phi_6^* &= (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_5)\end{aligned}$$

are partially Horn. Indeed, ϕ_6 contains the negative pure literal $\neg x_5$ and ϕ_6^* is Horn. On the other hand, the formulas

$$\begin{aligned}\phi_5 &= (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_4 \vee x_5) \text{ and}\\ \phi_7 &= (\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_4 \vee x_5)\end{aligned}$$

are not, since for every possible $V_0 \subseteq \{x_1, x_2, x_3, x_4, x_5\}$, we either get non-Horn clauses containing variables only from V_0, or variables of V_0 that appear positively in inadmissible clauses.

The formula

$$\phi_8 = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$$

is partially Horn. Its first three clauses are Horn, yet the third has to be put in every inadmissible set, since x_3 appears positively in the fourth clause which is not Horn. The first two clauses though constitute an admissible set of Horn clauses. Finally, the formula

$$\phi_9 = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$$

is not partially Horn. Indeed, since all its variables appear positively in some clause, we need at least one clause to be admissible. The first two clauses of ϕ_9 are Horn, but we will show that they both have to be included in an inadmissible set. Indeed, the second has to belong to every inadmissible set since x_3 appears positively in the third, not Horn, clause. Furthermore, x_2 appears positively in the second clause, which we just showed to belong to every inadmissible set. Thus, the first clause also has to be included in every inadmissible set, and therefore ϕ_9 is not partially Horn.

Accordingly to the case of renamable Horn formulas, we define:

Definition 16. *A formula is called renamable partially Horn if some of its variables can be renamed (in the sense of Definition 12) so that it becomes partially Horn.*

Observe that any Horn, renamable horn or partially Horn formula is trivially renamable partially Horn. Also, a formula with at least one pure positive literal is renamable partially Horn, since by renaming the corresponding variable, we get a formula with a pure negative literal.

Example 9. *All formulas of the previous example are renamable partially Horn: ϕ_6^*, ϕ_6 and ϕ_8 correspond to the trivial cases we discussed above, whereas ϕ_5, ϕ_7 and ϕ_9 all contain the pure positive literal x_4.*

Lastly, we examine two more formulas:

$$\phi_{10} = (\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \neg x_4 \text{ and}$$
$$\phi_{11} = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3).$$

Easily, the formula ϕ_{10} is not partially Horn, but by renaming x_1 and x_4, we obtain the partially Horn formula

$$\phi_{10}^* = (x_1 \vee x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge x_4,$$

where $V_0 = \{x_4\}$ is the set of admissible variables. One the other hand, the formula ϕ_{11} is not renamable partially Horn. Indeed, whichever variables we rename, we end up with one Horn and one non-Horn clause, with at least one variable of the Horn clause appearing positively in the non-Horn clause.

Observe that the definition of renamable partially Horn formulas provides no information about the relation between the set of the renamed variables and that of the admissible ones. The following remark clarifies any ambiguity regarding this issue.

Remark 3. *Let ϕ be a renamable partially Horn formula, and let ϕ^* be a partially Horn formula obtained by renaming some of the variables of ϕ, with V_0 being the admissible set of variables. Let also $\mathcal{C}_0$ be an admissible set of Horn clauses in ϕ^*. We can assume that only variables of V_0 have been renamed, since the other variables are not involved in the definition of being partially Horn. Also, we can assume that a Horn clause of ϕ^* all the variables of which are in V_0 belongs to $\mathcal{C}_0$. Indeed, if not, we can add it to $\mathcal{C}_0$.*

Example 10. *Consider the renamable partially Horn formula of the previous example, defined over the variable set $V = \{x_1, x_2, x_3, x_4\}$,*

$$\phi_{10} = (\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \neg x_4.$$

Recall that by renaming the variables x_1, x_4 we obtain the partially Horn formula

$$\phi_{10}^* = (x_1 \vee x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge x_4,$$

where the set of admissible variables is $V_0 = \{x_4\}$. In addition, we can trivially take the empty set as a set of admissible clauses, $\mathcal{C}_0 = \emptyset$. Observe that while establishing whether ϕ_{10}^* is in fact a partially Horn formula, we need not to take the variable x_1 into account. In other words, the logical sign of the literals of x_1 is insignificant, which in turn implies that there was no reason to rename this variable in the first place.

Indeed, by renaming in ϕ_{10} only the variable x_4 we obtain the partially Horn formula

$$\phi_{10}^{*\prime} = (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge x_4,$$

where the set of admissible variables is again the set $V_0 = \{x_4\}$. Moreover, we can assume that the Horn clause, x_4 which contains variables only from V_0, belongs to the set of admissible clauses, i.e., $\mathcal{C}_0' = \{x_4\}$.

Definition 17. *A formula is called a* possibility integrity constraint *if it is either affine, or separable, or renamable partially Horn.*

We will prove that possibility integrity constraints are the answer to the problem of the syntactic characterization of possibility domains. To this purpose, we first prove that renamable partially Horn formulas characterize domains closed under binary non projection aggregators. We will first need two lemmas. Lemma 3, allows us to work - if necessary- with binary non-projection aggregators whose components contain at most one of the projection operations pr_1^2 and pr_2^2. Lemma 4, is a generalization of Proposition 4 and provides us the ability to interchange between the symmetric components of a binary aggregator, according to desideratum, at the small cost of working with a slightly altered domain.

Lemma 3. *Suppose D admits a binary aggregator $f = (f_1, \ldots, f_m)$ such that there exists a partition (H, I, J) of $\{1, \ldots, m\}$ where f_h is symmetric for all $h \in H$, $f_i = \mathrm{pr}_s^2$, for all $i \in I$ and $f_j = \mathrm{pr}_t^2$, with $t \neq s$, for all $j \in J$. Then, D also admits a binary aggregator $g = (g_1, \ldots, g_m)$, such that $g_h = f_h$, for all $h \in H$ and $g_i = \mathrm{pr}_s^2$, for all $i \in I \cup J$.*

Proof. Without loss of generality, assume that there exist $1 \leq k < l < m$ such that $H = \{1, \ldots, k\}$, $I = \{k+1, \ldots, l\}$ and $J = \{l+1, \ldots, m\}$ and that $s = 1$ (and thus $t = 2$). It suffices to prove that, for two arbitrary vectors $a, b \in D$, $g(a, b) \in D$, where $(g_1, \ldots, g_m)$ is defined as in the statement of the lemma.

Assume that for all $i \in H$, $f_i(a_i, b_i) = c_i$. Since f is an aggregator for D, it holds that $f(a, b)$ and $f(b, a)$ are both vectors in D. By the same token, so is $f(f(a, b), f(b, a))$. The result is now obtained by noticing that:

$$
\begin{aligned}
f(a,b) &= (c_1, \ldots, c_k, a_{k+1}, \ldots, a_l, b_{l+1}, \ldots, b_m), \\
f(b,a) &= (c_1, \ldots, c_k, b_{k+1}, \ldots, b_l, a_{l+1}, \ldots, a_m),
\end{aligned}
$$

and thus: $f(f(a, b), f(b, a)) = (c_1, \ldots, c_k, a_{k+1}, \ldots, a_m) = g(a, b)$. □

Lemma 4. *Suppose D admits a binary aggregator $(f_1, \ldots, f_m)$ such that, for some $J \subseteq \{1, \ldots, m\}$, f_j is symmetric for all $j \in J$. For each $d = (d_1, \ldots, d_m) \in D$, let $d^* = (d_1^*, \ldots, d_m^*)$ be such that:*

$$
d_j^* = \begin{cases} 1 - d_j & \text{if } j \in J, \\ d_j & \text{else,} \end{cases}
$$

or $j = 1, \ldots, m$ and set $D^* = \{d^* \mid d \in D\}$. Then D^* admits the binary aggregator $(g_1, \ldots, g_m)$, where: (i) $g_j = \wedge$ or all $j \in J$ such that $f_j = \vee$, (ii) $g_j = \vee$ for all $j \in J$ such that $f_j = \wedge$ and (iii) $g_j = f_j$ for the rest.

Note that we do not assume that the set $J \subseteq \{1, \ldots, m\}$ includes every coordinate j such that f_j is symmetric.

Proof. The proof immediately follows from the fact that $\wedge(1 - d_j, 1 - d_j') = 1 - \vee(d_j, d_j')$ (resp. $\vee(1 - d_j, 1 - d_j') = 1 - \wedge(d_j, d_j')$), for any $d, d' \in D$; and that by renaming x_j, $j \in J$, in ϕ, we cause all of its literals to be satisfied by the opposite value. Thus, d satisfies ϕ if and only if d^* satisfies ϕ^*, where ϕ^* is the formula obtained from ϕ by this renaming. □

Theorem 9 (Dìaz et al. [9]). *D admits a binary aggregator $f = (f_1, \ldots, f_m)$ which is not a projection aggregator if and only if there exists a renamable partially Horn formula ϕ whose set of models equals D.*

Recall that for two vectors $a, b \in D$, we define $a \preceq b$ to mean that if $a_i = 1$ then $b_i = 1$, for all $i \in \{1, \ldots, m\}$ and $a \preceq b$ when $a \preceq b$ and $a \neq b$.

Proof. ($\Rightarrow$) We will work with the corresponding domain D^* of Lemma 4 that admits an aggregator $(g_1, \ldots, g_m)$ whose symmetric components, corresponding to the symmetric components of $(f_1, \ldots, f_m)$, are all equal to $\wedge$. Suppose that $V_0 = \{x_i \mid g_i = \wedge\}$. For D^*, we compute a formula $\phi = \phi_0 \wedge \phi_1$, where ϕ_0 is defined on the variables of V_0 and is Horn and where ϕ_1 has only negative appearances of variables of V_0. The result is then derived by renaming all the variables x_j, where j is such that $f_j = \vee$.

Let $I := \{i \mid f_i \text{ is symmetric}\}$ (by the hypothesis, $I \neq \emptyset$). Let also $J := \{j \mid f_j = \vee\}$ (J might be empty). Obviously $J \subseteq I$. For each $d = (d_1, \ldots, d_m) \in D$, let $d^* = (d_1^*, \ldots, d_m^*)$, where $d_j^* = 1 - d_j$ if $j \in J$ and $d_i^* = d_i$ else. Easily, if $D^* = \{d^* \mid d \in D\}$, by Lemmas 3 and 4 it admits an aggregator $(g_1, \ldots, g_m)$ such that $g_i = \wedge$, for all $i \in I$ and $g_j = \mathrm{pr}_1^2$, $j \notin I$. Thus, there is a Horn formula ϕ_0 on $\{x_i \mid i \in I\} := V_0$, such that $\mathrm{Mod}(\phi_0) = D_I^*$.

If $I = \{1, \ldots, m\}$, we have nothing to prove. Thus, suppose, without loss of generality, that $I = \{1, \ldots, k\}$, $k < m$. For each $a = (a_1, \ldots, a_k) \in D_I^*$, let $B_a := \{b \in D_{-I}^* \mid ab \in D^*\}$ be the set containing all partial vectors that can extend a. For each $a \in D_I^*$, let ψ_a be a formula on $\{x_j \mid j \notin I\}$, such that $\mathrm{Mod}(\psi_a) = B_a$. Finally, let $I_a := \{i \in I \mid a_i = 1\}$ and define:

$$\phi_a := \left(\bigwedge_{i \in I_a} x_i \right) \rightarrow \psi_a,$$

for all $a \in D_I^*$.

Consider the formula:

$$\phi = \phi_0 \wedge \left(\bigwedge_{a \in D_I^*} \phi_a \right).$$

We will prove that ϕ is partially Horn and that $\mathrm{Mod}(\phi) = D^*$. By Lemma 4, the renamable partially Horn formula for D can be obtained by renaming in ϕ the variables x_i such that $i \in J$.

We have already argued that ϕ_0 is Horn. Also, since ϕ_a is *logically equivalent* to (has exactly the same models as):

$$\left(\bigvee_{i \in I_a} \neg x_i \right) \vee \psi_a,$$

any variable of V_0 that appears in the clauses of some ϕ_a, does so negatively. It follows that ϕ is partially Horn.

Next we show that $D^* \subseteq \mathrm{Mod}(\phi)$ and that $\mathrm{Mod}(\phi) \subseteq D^*$. For the former inclusion, let $ab \in D^*$, where $a \in D_I^*$ and $b \in B_a$. Then, it holds that a satisfies ϕ_0 and b satisfies ψ_a. Thus ab satisfies ϕ_a.

Now, let $a' \in D_I^* : a \not\succeq a'$. Then, a does not satisfy $\bigwedge_{i \in I_{a'}} x_i$, since there exists some coordinate $i \in I_{a'}$ such that $a_i = 0$ and $a'_i = 1$. Thus, ab satisfies $\phi_{a'}$. Finally, let $a'' \in D_I^* : a'' \prec a$. Then, a satisfies $\bigwedge_{i \in I_{a''}} x_i$ and thus we must prove that b satisfies $\psi_{a''}$.

Since $a'' \in D_I^*$, there exists a $c \in D_{-I}^*$ such that $a''c \in D^*$. Then, since $(g_1, \dots, g_m)$ is an aggregator for D^*:

$$(g_1, \dots, g_m)(ab, a''c) = \\ (\wedge(a_1, a_1''), \dots, \wedge(a_k, a_k''), \mathrm{pr}_1^2(b_1, c_1), \dots, \mathrm{pr}_1^2(b_{m-k}, c_{m-k})) = a''b \in D^*,$$

since $a'' \prec a$. Thus, $b \in B(a'')$ and, consequently, it satisfies $\psi_{a''}$.

We will prove the opposite inclusion by showing that an assignment not in D^* cannot satisfy ϕ. Let $ab \notin D^*$. If $a \notin D_I^*$, we have nothing to prove, since a does not satisfy ϕ_0 and thus $ab \notin \mathrm{Mod}(\phi)$. So, let $a \in D_I^*$. Then, $b \notin B_a$, lest $ab \in D^*$. But then, b does not satisfy ψ_a and thus ab does not satisfy ϕ_a. Consequently, $ab \notin \mathrm{Mod}(\phi)$.

Thus, by renaming the variables x_i, $i \in J$, we produce a renamable partially Horn formula, call it ψ, such that $\mathrm{Mod}(\psi) = D$.

($\Leftarrow$) Let ψ be a renamable partially Horn formula with $\mathrm{Mod}(\psi) = D$. Let $J \subseteq \{1, \dots, m\}$ such that, by renaming all the x_i, $i \in J$, in ψ, we obtain a partially Horn formula ϕ. Let V_0 be the set of variables such that any clause containing only variables from V_0 is Horn, and that appear only negatively in clauses that contain variables from $V \setminus V_0$. By Remark 3, we can assume that $\{x_i \mid i \in J\} \subseteq V_0$. Let also $\mathcal{C}_0$ be the set of admissible Horn clauses of ϕ.

Let again $D^* = \{d^* \mid d \in D\}$, where $d_j^* = 1 - d_j$ if $j \in J$ and $d_i^* = d_i$ else, for all $d \in D$. By Lemma 4, $\mathrm{Mod}(\phi) = D^*$. By the same Lemma, and by noticing that whichever the choice of $J \subseteq \{1, \dots, n\}$, $(D^*)^* = D$, it suffices to prove that D^* is closed under a binary aggregator $(f_1, \dots, f_m)$, where $f_i = \wedge$ for all i such that $x_i \in V_0$ and $f_j = \mathrm{pr}_1^2$ for the rest.

Without loss of generality, let $I = \{1, \dots, k\}$, $k < m$ (lest we have nothing to show) be the set of indices of the variables in V_0. We need to show that if $ab, a'b' \in D$, where $a, a' \in D_I^*$ and $b, b' \in D_{-I}^*$, then $(a \wedge a')b \in D$, where $a \wedge a' = (a_1 \wedge a_1', \dots, a_k \wedge a_k')$.

Let $\phi = \phi_0 \wedge \phi_1$, where ϕ_0 is the conjunction of the clauses in $\mathcal{C}_0$ and ϕ_1 the conjunction of the rest of the clauses of ϕ. By the hypothesis, ϕ_0 is Horn and thus, since a, a' satisfy ϕ_0, so does $a \wedge a'$. Now, let C_r be a clause of ϕ_1. If any literal of C_r that corresponds to a variable not in ϕ_0 is satisfied by b, we have nothing to prove. If there is no such literal, since ab satisfies C_r, it must hold that a negative literal $\neg x_i$, $i \in I$, is satisfied by a. Thus, $a_i = 0$, which means that $a_i \wedge a_i' = 0$ too. Consequently, C_r is satisfied by $(a \wedge a')b$. Since C_r was arbitrary, the proof is complete. □

Finally, we are ready to prove that possibility integrity constraints are the answer to the problem of the syntactic characterization of possibility domains.

Theorem 10 (Díaz et al. [9]). *D is a possibility domain if and only if there exists a possibility integrity constraint ϕ whose set of models equals D.*

Proof. ($\Rightarrow$) If D is a possibility domain, then, by Theorem 7, it either admits a ternary aggregator all components of which are the binary addition mod 2, i.e., a minority aggregator, or a non-dictatorial binary projection aggregator, or a non-projection binary aggregator. In the first case, by Proposition 2, D is the model set of an affine formula. In the second, by Proposition 3, it is the model set of a separable formula and in the third, by Theorem 9, that of a renamable partially Horn formula. Thus, in all cases, D is the model set of a possibility integrity constraint.

($\Leftarrow$) Let ϕ be a possibility integrity constraint such that $\mathrm{Mod}(\phi) = D$. If ϕ is affine then, by Proposition 2, D admits a ternary aggregator all components of which are the binary addition mod 2, i.e., a minority aggregator. If ϕ is separable then, by

Proposition 3, D admits a non-dictatorial binary projection aggregator. Finally, by Theorem 9, if ϕ is renamable partially Horn then D admits a non-projection binary aggregator. In every case, D is a possibility domain. □

Remark 4. *At this point we should note that in case a domain is a singleton, $D = \{\bar{x}\}$, then for every (supportive) aggregator it holds that $f[D^n] = \{\bar{x}\}$, which implies that any aggregator, restricted on D, degenerates into a dictatorial one. An analogous issue is raised whenever the projection D_j of D on issue $j \in \{1, \ldots, m\}$ is a singleton, as in this case the corresponding component of any aggregator degenerates to a projection operation.*

To avoid such degeneracies, in Computational Social Choice domains are usually assumed to have at least two elements and that their projection on each coordinate is $\{0, 1\}$. In this context, we assume that the class of possibility integrity constraints is restricted to contain only those whose set of satisfying truth assignments is also non-degenerate.

Chapter 3

Other forms of non-dictatorial aggregation- Characterizations of the corresponding domains

In this chapter, we study several different sub-classes of non-dictatorial aggregators that have been introduced in the field of judgment aggregation. Namely, locally non-dictatorial aggregators (which have no projection components), aggregators that are not *generalized dictatorships* (whose output is not always one of the vectors of their input), *anonymous* aggregators (which are not affected by any permutation of their input), *monotone* aggregators (whose output does not change if more voters agree with it) and *StrongDem* aggregators (where the votes of any $k-1$ voters can be fixed in a way such that the k-th voter cannot change the result). Furthermore, we syntactically characterize domains that admit each of the above five kinds of aggregators. Then, we consider the property of systematicity and examine how our results change if the aggregators are required to satisfy this property.

3.1 Local possibility domains

In this section, we consider a subclass of non-dictatorial aggregators, that consists of aggregators which are not a projection function, even when restricted to any given issue. This type of aggregators, called *locally non-dictatorial aggregators* was introduced by Nehring and Puppe [25]. In what follows, we present the syntactic characterization of Díaz et al. [9], for *local possibility domains*, i.e., Boolean domains that admit a locally non-dictatorial aggregator.

The stimulus for studying this class of aggregators was that, in some cases, a non-dictatorial aggregator may not comply with our sense of social fairness. As Nehring and Puppe state:

> Non-dictatorial aggregation rules can still be rather degenerate, e.g. by giving almost all decision power to one agent (voter), or by specifying different "local" dictators for different issues. [25, p.478]

Since this type of authoritarianism may be unfit for certain decision making processes, locally non-dictatorial aggregators were introduced.

Definition 18. *A n-ary aggregator* $(f_1, \ldots, f_m)$ *is locally non-dictatorial if* $f_j \neq \mathrm{pr}_d^n$*, for all* $d \in \{1, \ldots, n\}$ *and* $j = 1, \ldots, m$.

Definition 19. *D is a* local possibility domain (lpd) *if it admits a locally non-dictatorial aggregator.*

It is straightforward to establish that the class of lpd's is a proper subclass of that of possibility domains. Indeed, every locally non-dictatorial aggregator is, clearly, non-dictatorial, hence every lpd is a possibility domain. On the other hand, any non-dictatorial aggregator with at least one projection component is locally dictatorial. The following is an example of a possibility domain that is not an lpd.

Example 11. *Let D be the Cartesian product $D = \{(1,0,0),(0,1,0),(0,0,1)\} \times \{0,1\}$. We can easily establish that D is a possibility domain, as it is closed under the non-dictatorial projection aggregator $f = (\mathrm{pr}_1^2, \mathrm{pr}_1^2, \mathrm{pr}_1^2, \mathrm{pr}_2^2)$. However, the set $\{(1,0,0),(0,1,0),(0,0,1)\}$ only admits dictatorial aggregators, therefore D admits no locally non-dictatorial aggregators.*

Local possibility domains were also introduced by Kirousis et al. in [18], as *uniform* non-dictatorial domains[1], where the following characterization[2] was proven:

Theorem 11 (Kirousis et al. [18])**.** *$D \subseteq \{0,1\}^m$ is a local possibility domain if and only if it admits a ternary aggregator $(f_1,\dots,f_m)$ such that $f_j \in \{\wedge^{(3)}, \vee^{(3)}, maj, \oplus\}$, for $j = 1,\dots,m$.*

Theorem 11 consists a characterization of local possibility domains in terms of the aggregators they admit. As in the case of possibility domains, we want to verify how these closure properties are reflected on the syntactic properties of the corresponding integrity constraints. In other words, we seek a certain syntactic type of formulas - that is to be determined- whose sets of models describe local possibility domains in total. Following Díaz et al. [9] we call this type of formulas *local possibility integrity constraints*.

We will first address a technical issue. Let V, V' be two disjoint sets of variables. By further generalizing the notion of a clause of a CNF formula, we say that a (V,V')-*generalized clause* is a clause of the form:

$$(l_1 \vee \cdots \vee l_s \vee (l_{s+1} \oplus \cdots \oplus l_t)),$$

where the literal l_j corresponds to variable v_j, $j = 1,\dots,t$, $v_1,\dots,v_s \in V$, $v_{s+1},\dots,v_t \in V'$ and $0 \le s < t$. Such a clause is falsified by exactly those assignments that falsify every literal l_i, $i = 1,\dots,s$ and satisfy an even number of literals l_j, $j = s+1,\dots,t$. An affine clause is trivially a (V,V')-generalized clause, where all its literals correspond to variables from V'.

Consider now the following definition, which is analogous to Definition 17.

Definition 20. *A formula ϕ is a local possibility integrity constraint (lpic) if there are three pairwise disjoint subsets $V_0, V_1, V_2 \subseteq V$, with $V_0 \cup V_1 \cup V_2 = V$, where no clause contains variables both from V_1 and V_2 and such that:*

1. *by renaming some variables of V_0, we obtain a partially Horn formula ϕ^*, whose set of admissible variables is V_0,*
2. *any clause contains at most two variables from V_1 and*
3. *the clauses containing variables from V_2 are (V_0, V_2)-generalized clauses.*

[1] The notion of uniform non-dictatorial domains is broader than this of lpd's in the sense that in their work Kirousis et al. in [18], refer to both Boolean and non-Boolean frameworks.

[2] For a corresponding characterization for the non-Boolean case see Kirousis et al. [18]

Example 12. *Easily, every (renamable) Horn, bijunctive or affine formula is an lpic. On the other hand, consider the following possibility integrity constraint:*

$$\phi = (\neg x_1 \lor x_2 \lor x_3 \lor x_4) \land (\neg x_2 \lor \neg x_3 \lor \neg x_4).$$

ϕ is partially Horn, since it has the pure negative literal $\neg x_1$ and thus a possibility integrity constraint. But, it is not an lpic, since however we define V_0, V_1, either there will be a variable of V_0 with a positive appearance in a non-admissible clause (even after any possible renaming of the variables of V_0) and/or there will be a clause with more than two literals from V_1.

We will prove that local possibility integrity constraints syntactically characterize local possibility domains. Our approach is entirely analogous to this of possibility domains, only this time the starting point is Theorem 11. Intuitively, we want the different types of coordinate functions of an aggregator to correspond to different types of variables of an lpic. In fact, the set V_0 is destined to contain variables that correspond to $\land^3$ and $\lor^3$ components, V_1 variables that correspond to maj components, and V_2 those that correspond to $\oplus$ components of an aggregator. From this perspective, Definition 20 becomes less cumbersome and we can easily acquire the following corollary:

Corollary 1. *If ϕ is a local possibility integrity constraint, then it is also a possibility integrity constraint.*

Proof. Let V_0, V_1 and V_2 be as in Definition 20. If $V_0 \neq \emptyset$, ϕ is renamable partially Horn. Else, if $V_0 = V_1 = \emptyset$, then ϕ is affine. On the other hand, if $V_0 = \emptyset$ and V_1 and V_2 are not, ϕ is separable. Finally, if $V_1 = V$, then ϕ is bijunctive and equivalently, 2-SAT. The result now follows from the fact that any 2-SAT formula is renamable Horn. Indeed, let α be an assignment satisfying ϕ and rename all the variables $x \in V$ such that $\alpha(x) = 1$. Then, every clause of ϕ either has a positive literal that is renamed, or a negative one that is not renamed. □

Before we syntactically characterize lpd's as the sets of models of lpic's, we first need two lemmas.

Lemma 5. *Let $D \subseteq \{0,1\}^m$ and $I = \{j_1, \ldots, j_t\} \subseteq \{1, \ldots, m\}$. If $f = (f_1, \ldots, f_m)$ is a k-ary aggregator for D, then $(f_{j_1}, \ldots, f_{j_t})$ is a k-ary aggregator for D_I.*

Proof. Without loss of generality, assume $I = \{1, \ldots, s\}$, $s \leq m$, and that $a^1, \ldots, a^k \in D_I$. It follows that there exist $b^1, \ldots, b^k \in D_{-I}$ such that $c^1, \ldots, c^k \in D$, where $c^i = a^i b^i$, $i = 1, \ldots k$. Since f is an aggregator for D:

$$f(c^1, \ldots, c^k) := (f_1(c^1_1, \ldots, c^k_1), \ldots, f_m(c^1_m, \ldots c^k_m)) \in D.$$

Thus, $(f_1(c^1_1, \ldots, c^k_1), \ldots, f_s(c^1_s, \ldots, c^k_s)) \in D_I$. □

Lemma 6. *Suppose that D admits a ternary aggregator $f = (f_1, \ldots, f_m)$, where $f_j \in \{\land^{(3)}, maj, \oplus\}$, $j = 1, \ldots, m$. Then D admits a binary aggregator $g = (g_1, \ldots, g_m)$ such that $g_i = \land$, for all i such that $f_i = \land^{(3)}$, $g_j = \mathrm{pr}^2_1$, for all j such that $f_j = maj$ and $g_k = \mathrm{pr}^2_2$, for all k such that $f_k = \oplus$.*

Proof. The result is immediate, by defining $g = (g_1, \ldots, g_m)$ such that:

$$g_j(x, y) = f_j(x, x, y),$$

for $j = 1, \ldots, m$. □

Theorem 12 (Dìaz et al. [9]). *A domain $D \subseteq \{0,1\}^m$ is a local possibility domain if and only if there is a local possibility integrity constraint ϕ such that $Mod(\phi) = D$.*

Proof. ($\Rightarrow$) In all that follows, we assume that an empty domain is described by the empty formula, which is trivially Horn, bijunctive and affine. The proof will closely follow that of Theorem 9.

Since D is an lpd, by Theorem 11, there is a ternary aggregator $f = (f_1, \dots, f_m)$ such that every component $f_j \in \{\wedge^{(3)}, \vee^{(3)}, \text{maj}, \oplus\}$, $j = 1, \dots, m$. Again, let $D^* = \{d^* \mid d \in D\}$, where $d^*_j = 1 - d_j$ if j is such that $f_j = \vee^{(3)}$, and $d^*_j = d_j$ in any other case. Thus, by Lemma 4, D^* admits a ternary aggregator $g = (g_1, \dots, g_m)$ such that $g_j \in \{\wedge^{(3)}, \text{maj}, \oplus\}$, for $j = 1, \dots, m$. Thus, by showing that D^* is described by a lpic ϕ, we will obtain the same result for D by renaming all the variables x_j, where j is such that $f_j = \vee^{(3)}$.

Without loss of generality, assume that $I := \{i \mid g_i = \wedge^{(3)}\} = \{1, \dots, s\}$, $J := \{j \mid g_j = \text{maj}\} = \{s+1, \dots, t\}$ and $K := \{k \mid g_k = \oplus\} = \{t+1, \dots, m\}$, where $0 \leq s \leq t \leq m$. Since D^*_I is Horn, there is a Horn formula ϕ_0 such that $\text{Mod}(D^*_I) = \phi_0$.

If $s = t = m$, we have nothing to prove. Thus, suppose $s < t \leq m$. For each $a = (a_1, \dots, a_s) \in D^*_I$, let $B^1_a := \{b \in D^*_J \mid ab \in D^*_{I \cup J}\}$ and $B^2_a := \{c \in D^*_J \mid ac \in D^*_{I \cup K}\}$ be the sets of partial vectors extending a to the indices of J and K respectively.

Claim 3. *For each $a \in D^*_I$, B^1_a and B^2_a are bijunctive and affine respectively.*

Proof. We will prove the claim for B^1_a. The proof for B^2_a is the same.

Let $b^1, b^2, b^3 \in B^1_a$. Then $ab^1, ab^2, ab^3 \in D^*_{I \cup J}$. By Lemma 5, $(g_1, \dots, g_t)$ is an aggregator for $D^*_{I \cup J}$ and by the definition of g, it holds that $ab \in D^*_{I \cup J}$, where $b = \text{maj}(b^1, b^2, b^3)$. Thus, $b \in B^1_a$ and the result follows. □

Thus, for each $a \in D^*_I$, there is a bijunctive formula ψ_a and an affine χ_a, such that $\text{Mod}(\psi_a) = B^1_a$ and $\text{Mod}(\chi_a) = B^2_a$. Let $I_a := \{i \in I \mid a_i = 1\}$ and define:

$$\phi^1_a := \left(\bigwedge_{i \in I_a} x_i \right) \to \psi_a$$

and

$$\phi^2_a := \left(\bigwedge_{i \in I_a} x_i \right) \to \chi_a,$$

for all $a \in D^*_I$.

Consider the formula:

$$\phi = \phi_0 \wedge \left(\bigwedge_{a \in D^*_I} \phi^1_a \right) \wedge \left(\bigwedge_{a \in D^*_I} \phi^2_a \right).$$

Let $V_0 = \{x_i \mid i \in I\}$, $V_1 = \{x_j \mid j \in J\}$ and $V_2 = \{x_k \mid k \in K\}$. The fact that ϕ is partially Horn with admissible set V_0, can be seen in the same way as in Theorem 9. Now, consider ψ_a, for some $a \in D^*_I$. Since it is bijunctive, it is of the form:

$$\psi_a = \bigwedge_{j=1}^{r} \left(l_{j_1} \vee l_{j_2} \right),$$

where l_{j_i} are literals of variables from V_1. Thus, ϕ_a^1 is equivalent to:

$$\bigwedge_{j=1}^{r}\left(\left(\bigvee_{i\in I_a}\neg x_i\right)\vee l_{j_1}\vee l_{j_2}\right).$$

Thus, the clauses of ϕ_a^1 contain at most two literals from V_1.

In the analogous way, we can see that the clauses of ϕ_a^2 are (V_0, V_1)-generalized clauses. Finally, by construction, there is no clause in ϕ that contains variables both from V_1 and V_2. It follows that ϕ is an lpic. What remains now is to show that $\text{Mod}(\phi) = D^*$.

Observe that by Lemmas 6 and 3, D^* admits a binary aggregator $h = (h_1, \ldots, h_m)$ such that $h_i = \wedge$, for all $i \in I$ and $h_j = \text{pr}_1^2$, for all $j \in J \cup K$. The proof now is exactly like the one of Theorem 9, by letting $B_a = \{bc \mid b \in B_a^1 \text{ and } c \in B_a^2\}$ and

$$\phi_a = \phi_a^1 \wedge \phi_a^2.$$

($\Leftarrow$) Let ψ be an lpic, with $\text{Mod}(\psi) = D$. Let V_0, V_1 and V_2 be subsets of V as in Definition 20. Let also ϕ be the partially Horn formula obtained by ψ by renaming the variables of a subset $V^* \subseteq V_0$. Again, assume $D^* = \{d^* \mid d \in D\}$, where $d_j^* = 1 - d_j$ if $x_j \in V^*$ and $d_i^* = d_i$ else, for all $d \in D$. By Lemma 4, $\text{Mod}(\phi) = D^*$. Thus, by Theorem 11, it suffices to prove that D^* is closed under a ternary aggregator $(f_1, \ldots, f_m)$, where $f_i \in \{\wedge^{(3)}, \text{maj}, \oplus\}$ for $i = 1, \ldots, m$.

Without loss of generality, let $I = \{1, \ldots, s\}$, be the set of indices of the variables in V_0, $J = \{s+1 \ldots, t\}$ be that of the indices of variables in V_1 and $K = \{t+1, \ldots, m\}$ that of the indices of variables in V_2. We need to show that if $abc, a'b'c', a''b''c'' \in D^*$, where $a, a', a'' \in D_I^*$, $b, b', b'' \in D_J^*$ and $c, c', c'' \in D_K^*$, then

$$d := (\wedge^{(3)}(a, a', a''), \text{maj}(b, b', b''), \oplus(c, c', c'')) \in D^*.$$

Let $\phi = \phi_0 \wedge \phi_1 \wedge \phi_2$, where ϕ_0 is the conjunction of the clauses containing only variables from V_0, ϕ_1 the conjunction of clauses containing variables from V_1 and where ϕ_2 contains the rest of the clauses of ϕ. Observe that by the hypothesis, there is no variable appearing both in a clause of ϕ_1 and ϕ_2.

By the hypothesis, ϕ_0 is Horn and thus, since a, a', a'' satisfy ϕ_0, so does $a \wedge a' \wedge a''$.

Now, let C_r be a clause of ϕ_1. Suppose that there is a literal of a variable $x_i \in V_0$ in C_r that is satisfied by a. Since ϕ is partially Horn with respect to V_0, it must hold that this literal was $\neg x_i$. This means that $a_i = 0$ and thus $\wedge^{(3)}(a_i, a_i', a_i'') = 0$. The same holds if $\neg x_i$ is satisfied by a' or a''. Thus, C_r is satisfied.

Now, suppose there is no such literal and that the literals of C_r corresponding to variables of V_1 are l_i, l_j. Since $abc, a'b'c', a''b''c''$ satisfy ϕ, it holds that (b_i, b_j), (b_i', b_j') and (b_i'', b_j'') satisfy $l_i \vee l_j$. Without loss of generality, Assume that $\text{maj}(b_i, b_i', b_i'') = b_i$ and that b_i does not satisfy l_i (lest we have nothing to prove). Then, $b_i = b_i'$ or $b_i = b_i''$. Assume the former (again without loss of generality). Then, it must be the case that b_j, b_j' satisfy l_j. Thus $b_j = b_j'$ and $\text{maj}(b_j, b_j', b_j'') = b_j$, which satisfies l_j. In every case, C_r is satisfied by d.

Now, let C_q be a clause of ϕ_2. Again, if there there is a literal of a variable $x_i \in V_0$ in C_q that is satisfied by a, we obtain the required result as in the case of C_r. Thus, suppose there is no such literal and that the sub-clause of C_q obtained by deleting the variables of V_0 is:

$$C_q' = (l_1 \oplus \cdots \oplus l_z).$$

Since $abc, a'b'c', a''b''c''$ satisfy ϕ, it holds that c, c' and c'' satisfy C'_q. Since C'_q is affine, it holds that $\oplus(c, c', c'')$, and satisfies it.

In all cases, we proved that d satisfies ϕ and thus the proof is complete. □

Remark 5. *Note that, due to the non-degeneracy conditions, we have assumed that all the formulas we consider have non-degenerate domains (see Remark 4).*

3.2 Generalized Dictatorships

We now turn to the aggregation procedures called *generalized dictatorships*. In our framework, a n-ary aggregator is a generalized dictatorship if, on input any n vectors from a domain D, always returns one of those vectors as its output. These aggregators are a natural generalization of the notion of dictatorial aggregators, in the sense that they select a possibly different "dictator" for each set of n feasible voting patters, instead of a single global one.

However, these aggregators were originally introduced by Cariani et al. [5] as *rolling dictatorships*, under the stronger requirement that the above property holds for any n vectors of $\{0,1\}^m$. Under this requirement, a generalized dictatorship selects for each n-tuple of vectors from $\{0,1\}^m$ one of these vectors as its outcome (though not necessarily always the same one), hence the denomination "rolling" dictatorships. In that framework, Grandi and Endriss [13] showed that the class of generalized dictatorships coincides with the class of operators that are aggregators for every Boolean domain $D \subseteq \{0,1\}^m$.

The difference with our framework is that generalized dictatorships -and all aggregators in general- even though they are defined over $\{0,1\}^{n \times m}$, they are studied within the "rationality restrictions" of a given domain, in the sense that we are not interested in how they behave on irrational inputs, i.e. inputs outside of the domain. In this section, we show that domains admitting aggregators which are not generalized dictatorships are exactly the possibility domains (apart from some trivial cases), and are thus described by possibility integrity constraints, a result proved by Díaz et al. [9].

Definition 21. *Let $f = (f_1, \ldots, f_m)$ be an m-tuple of n-ary conservative functions. We say that f is a* generalized dictatorship *for a domain $D \subseteq \{0,1\}^m$, if, for any $x^1, \ldots, x^n \in D$, it holds that:*

$$f(x^1, \ldots, x^n) := (f_1(x_1), \ldots, f_m(x_m)) \in \{x^1, \ldots, x^n\}. \tag{3.1}$$

Much like dictatorial functions, it is straightforward to observe that if f is a generalized dictatorship for D, then it is also an aggregator for D.

The following example shows that the result of Grandi and Endriss [13, Theorem 16] does not hold in our setting, as it illustrates an aggregator that is a generalized dictatorship for one domain and not for another.

Let us first introduce some additional notation. In what follows, we will denote by $\bar{f}$ an n-ary aggregator $(f_1, \ldots, f_m)$, where $f_1 = \ldots = f_m := f$. Such aggregators are called systematic (see Section 3.4).

Example 13. *Consider the binary aggregator $\bar{\wedge} = (\wedge, \wedge, \wedge)$ and the the Horn formulas:*

$$\phi_{12} = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3),$$

and

$$\phi_{13} = (\neg x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3).$$

The set of satisfying assignments of ϕ_{12} is:

$$Mod(\phi_{12}) = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\}.$$

By definition, $Mod(\phi_{12})$ is a Horn domain and it thus admits the binary symmetric aggregator $\bar{\wedge} = (\wedge, \wedge, \wedge)$. Furthermore, $\bar{\wedge}$ is not a generalized dictatorship for $Mod(\phi_{12})$, since $\bar{\wedge}((0,0,1),(0,1,0)) = (0,0,0) \notin \{(0,0,1),(0,1,0)\}$.

On the other hand, $\bar{\wedge}$ is again an aggregator for the Horn domain:

$$Mod(\phi_{13}) = \{(0,0,0), (0,1,0), (0,1,1), (1,1,1)\},$$

but, contrary to the previous case, $\bar{\wedge}$ is a generalized dictatorship for $Mod(\phi_{13})$, since it is easy to verify that for any $x, y \in Mod(\phi_{13})$, $\bar{\wedge}(x,y) \in \{x,y\}$.

Finally, observe that $(\wedge, \vee, \vee)$ is an aggregator for $Mod(\phi_{13})$ that is not a generalized dictatorship. The latter claim follows from the fact that:

$$(\wedge, \vee, \vee)((0,1,0),(1,1,1)) = (0,1,1) \notin \{(0,1,0),(1,1,1)\},$$

while the former is left to the reader. Thus, interestingly enough, ϕ_{13} describes a domain admitting an aggregator that is not a generalized dictatorship, although it is not the aggregator that "corresponds" to the formula.

This means that deciding whether an aggregator is a generalized dictatorship depends on the domain in question. Therefore, generalized dictatorship as a feature is not necessarily an inbuilt property of an aggregator.

Despite the above, it is easy to see that a dictatorial aggregator $(\mathrm{pr}_i^n, \ldots, \mathrm{pr}_i^n)$ is a generalized dictatorship for any $D \subseteq \{0,1\}^m$, for all $n \geq 1$ and for all $i \in \{1, \ldots, n\}$. Thus, trivially, every domain admits aggregators which are generalized dictatorships. The following result is also straightforward.

Lemma 7. *Let $D \subseteq \{0,1\}^m$, where $|D| = 2$. Then, every aggregator f for D is a generalized dictatorship.*

Proof. Assume that $D = \{x, y\}$, where x and y are distinct and that f is a n-ary aggregator for D that is not a generalized dictatorship for D. Then, it must hold either that $f(x, \ldots, x) = y$ or $f(y, \ldots, y) = x$ (the output of f must always be x or y since f is an aggregator of $\{x, y\}$). Contradiction, since f is conservative. □

Recall that a domain cannot have strictly less than two elements since we have assumed that it is not degenerate.

Again, the aim is to syntactically characterize domains that admit aggregators which are not generalized dictatorships. The following result shows that these domains are all the possibility domains with at least three elements, and are thus characterized by possibility integrity constraints.

Theorem 13 (Dìaz et al. [9])**.** *A domain $D \subseteq \{0,1\}^m$, with at least three elements, admits an aggregator that is not a generalized dictatorship if and only if it is a possibility domain.*

Proof. The forward direction is obtained by the trivial fact that an aggregator that is not a generalized dictatorship is also non-dictatorial.

Now, suppose that D is a possibility domain. Then it is either affine or it admits a binary non-dictatorial aggregator. We begin with the affine case. It is a known result that $D \subseteq \{0,1\}^m$ is affine if and only if it is closed under $\oplus$, or, equivalently, if it admits the minority aggregator:

$$\bar{\oplus} = (\underbrace{\oplus, \ldots, \oplus}_{m\text{-times}})$$

Claim 4. *Let $D \subseteq \{0,1\}^m$ be an affine domain. Then, the* minority *aggregator:*

$$\bar{\oplus} = (\underbrace{\oplus, \ldots, \oplus}_{m\text{-times}})$$

is not a generalized dictatorship for D.

Proof. Let $x, y, z \in D$ be three pairwise distinct vectors. Since $y \neq z$, there exists a $j \in \{1, \ldots, m\}$ such that $y_j \neq z_j$. It follows that $y_j + z_j \equiv 1(\mod 2)$. This means that $\oplus(x_j, y_j, z_j) \neq x_j$ and thus that $\bar{\oplus}(x, y, z) \neq x$. In the same way we show that $\bar{\oplus}(x, y, z) \notin \{x, y, z\}$, which is a contradiction, since $\bar{\oplus}$ is an aggregator for D. □

Now, recall that if $f = (f_1, \ldots, f_m)$ is a binary non-dictatorial aggregator, then $f_j \in \{\wedge, \vee, \mathrm{pr}_1^2, \mathrm{pr}_2^2\}$, $j = 1, \ldots, m$. If $f_j \in \{\wedge, \vee\}$ for all $j \in \{1, \ldots, m\}$, we call f symmetric, whereas if $f_j \in \{\mathrm{pr}_1^2, \mathrm{pr}_2^2\}$ for all $j \in \{1, \ldots, m\}$, we call f a projection aggregator.

Claim 5. *Suppose $D \subseteq \{0,1\}^m$ admits a binary non-dictatorial non-symmetric aggregator $f = (f_1, \ldots, f_m)$. Then f is not a generalized dictatorship.*

Proof. Assume, to obtain a contradiction, that f is a generalized dictatorship for D and let $x, y \in D$. Then, $f(x, y) := z \in \{x, y\}$. Assume that $z = x$. The case where $z = y$ is analogous.

Let $J \subseteq \{1, \ldots, m\}$ such that f_j is symmetric, for all $j \in J$ and f_j is a projection otherwise. Note that $J \neq \{1, \ldots, m\}$. Let also $I \subseteq \{1, \ldots, m\} \setminus J$, such that $f_i = \mathrm{pr}_2^2$, for all $i \in I$ and $f_i = \mathrm{pr}_1^2$ otherwise. If $I \neq \emptyset$, then, for all $i \in I$, it holds that:

$$y_i = pr_2^2(x_i, y_i) = f_i(x_i, y_i) = z_i = x_i.$$

Since x, y were arbitrary, it follows that $D_i = \{x_i\}$, for all $i \in I$. Contradiction, since D is non-degenerate.

If $I = \emptyset$, then $f_j = \mathrm{pr}_1^2$, for all $j \notin J$. Note that in that case, $J \neq \emptyset$, lest f is dictatorial. Now, consider $f(y, x) := w \in \{x, y\}$ since f is a generalized dictatorship. By the definition of f, $w_j = z_j = x_j$, for all $j \in J$, and $w_i = y_i$, for all $i \notin J$. Thus, if $w = x$, D is degenerate on $\{1, \ldots, m\} \setminus J$, whereas if $w = y$, D is degenerate on J. In both cases, we obtain a contradiction. □

The only case left is when $D \subseteq \{0,1\}^m$ admits a binary symmetric aggregator. Contrary to the previous cases, where we showed that the respective non-dictatorial aggregators could not be generalized dictatorships, here we cannot argue this way, as Example 13 indicates. Interestingly enough, we show that as in Example 13, we can always find some symmetric aggregator for such a domain that is not a generalized dictatorship.

Claim 6. *Suppose $D \subseteq \{0,1\}^m$ admits a binary non-dictatorial symmetric aggregator $f = (f_1, \ldots, f_m)$. Then, there is a binary symmetric aggregator $g = (g_1, \ldots, g_m)$ for D (g can be different from f) that is not a generalized dictatorship for D.*

Proof. If f is not a generalized dictatorship for D, we have nothing to prove. Suppose it is and let $J \subseteq \{1, \ldots, m\}$, such that $f_j = \vee$, for all $j \in J$ and $f_i = \wedge$ for all $i \notin J$ (J can be both empty or $\{1, \ldots, m\}$).

Let $D^* = \{d^* = (d_1^*, \ldots, d_m^*) \mid d = (d_1, \ldots, d_m) \in D\}$, where:

$$d_j^* = \begin{cases} 1 - d_j & \text{if } j \in J \\ d_j & \text{else.} \end{cases}$$

By Lemma 4, $h = (h_1, \dots, h_m)$ is a symmetric aggregator for D if and only if $h^* = (h_1^*, \dots, h_m^*)$ is an aggregator for D^*, where $h_j^* = h_j$, for all $j \notin J$ and, for all $j \in J$, if $h_j = \vee$, then $h_j^* = \wedge$ and vice-versa. As expected, the property of being a generalized dictatorship carries on this transformation.

Claim 7. *The operator h is a generalized dictatorship for D if and only if h^* is a generalized dictatorship for D^*.*

Proof. Let $x = (x_1, \dots, x_m), y = (y_1, \dots, y_m) \in D$ and $z := h(x, y)$. Since $\vee(x_j, y_j) = 1 - \wedge(1 - x_j, 1 - y_j)$ and $\wedge(x_j, y_j) = 1 - \vee(1 - x_j, 1 - y_j)$, it holds that $z_j = h_j^*(x_j^*, y_j^*)$, for all $j \notin J$, and $1 - z_j = h_j^*(x_j^*, y_j^*)$, for all $j \in J$. Thus, $z^* = h^*(x^*, y^*)$. It follows that $z \in \{x, y\}$ if and only if $z^* \in \{x^*, y^*\}$. □

Now, since D admits the generalized dictatorship f, it follows that D^* admits the binary aggregator $\bar{\wedge} = (\underbrace{\wedge, \dots, \wedge}_{m\text{-times}})$, that is also a generalized dictatorship. Our aim is to show that D^* admits a symmetric aggregator that is not a generalized dictatorship. The result will then follow by Claim 7.

For two elements $x^*, y^* \in D^*$, we write $x^* \leq y^*$ if, for all $j \in \{1, \dots, m\}$ such that $x_j^* = 1$, it holds that $y_j^* = 1$.

Claim 8. *$\leq$ is a* total ordering *for D^*.*

Proof. To obtain a contradiction, let $x^*, y^* \in D^*$ such that neither $x^* \leq y^*$ nor $y^* \leq x^*$. Thus, there exist $i, j \in \{1, \dots, m\}$, such that $x_i^* = 1$, $y_i^* = 0$, $x_j^* = 0$ and $y_j^* = 1$. Thus:

$$\wedge(x_i^*, y_i^*) = \wedge(x_j^*, y_j^*) = 0.$$

Then, $\bar{\wedge}(x^*, y^*) \notin \{x^*, y^*\}$. Contradiction, since $\bar{\wedge}$ is a generalized dictatorship. □

Thus, we can write $D^* = \{d^1, \dots, d^N\}$, where $d^s \leq d^t$ if and only if $s \leq t$. Let $I \subseteq \{1, \dots, m\}$ be such that, for all $j \in I$: $d_j^s = 0$ for $s = 1, \dots, N-1$, and $d_j^N = 1$. Observe that I cannot be empty, lest $d^N = d^{N-1}$ and that $I \neq \{1, \dots, m\}$, since $|D| \geq 3$. Let now $g = (g_1, \dots, g_m)$ such that $g_j = \wedge$, for all $j \in I$ and $g_j = \vee$, for all $j \notin I$.

We show that g is an aggregator for D^*. Indeed, let $d^s, d^t \in D^*$ with $s \leq t \leq N-1$. Then, for all $j \notin I$:

$$g_j(d_j^s, d_j^t) = \vee(d_j^s, d_j^t) = d_j^t.$$

Also for all $j \in I$:

$$g_j(d_j^s, d_j^t) = \wedge(d_j^s, d_j^t) = 0 = d_j^t.$$

Thus, $g(d^s, d^t) = d^t \in D^*$. Finally, consider $g(d^s, d^N)$. Again, $g_j(d_j^s, d_j^N) = \wedge(d_j^s, d_j^N) = 0$ for all $j \in I$ and $g_j(d_j^s, d_j^N) = \vee(d_j^s, d_j^N) = d_j^N$, for all $j \notin I$. By definition of I, $g(d^s, d^N) = d^{N-1} \in D^*$. This, last point shows also that g is not a generalized dictatorship, since, for any $s \neq N-1$, $d^{N-1} \notin \{d^s, d^N\}$. □

This completes the proof of Theorem 13. □

By Theorems 10 and 13 we obtain the following characterization.

Corollary 2 (Dìaz et al. [9]). *A domain $D \subseteq \{0,1\}^m$, with at least three elements, admits an aggregator that is not a generalized dictatorship if and only if there exists a possibility integrity constraint whose set of models equals D.*

Remark 6. *Apart from the initial issue regarding the non-degeneracy assumptions we discussed in Remark 4, here we have to deal with the additional requirement for the domain to be comprised of three or more elements. Thus, for this case* only, *we implicitly assume that we only consider possibility integrity constraints whose sets of models are comprised of at least three elements. In Section 4.3, we argue that it is possible to distinguish such possibility integrity constraints from the rest (see Remark 9).*

3.3 Anonymous, Monotone and StrongDem Aggregators

In this section we turn to three more kinds of non-dictatorial aggregators that have been studied in the field of Judgement Aggregation, namely *anonymous*, *monotone* and *StrongDem* aggregators. Each of these types of aggregation procedures, bears an appealing property, inspired by the features of majority voting.

Definition 22. *Let $D \subseteq \{0,1\}^m$. A n-ary aggregator $f = (f_1, \dots, f_m)$ for D is:*

1. *Anonymous, if it holds that for all $j \in \{1, \dots, m\}$ and for any permutation $p : \{1, \dots, n\} \mapsto \{1, \dots, n\}$:*

$$f_j(a_1, \dots, a_n) = f_j(a_{p(1)}, \dots, a_{p(n)}),$$

for all $a_1, \dots, a_n \in \{0,1\}$.

2. *Monotone, if it holds that for all $j \in \{1, \dots, m\}$ and for all $i \in \{1, \dots, n\}$:*

$$f_j(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) = 1 \Rightarrow f_j(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n) = 1.$$

3. *StrongDem, if it holds that for all $j \in \{1, \dots, m\}$ and for all $i \in \{1, \dots, n\}$, there exist $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n \in \{0,1\}$:*

$$f_j(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) = f_j(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n).$$

Anonymous aggregators ensure that the outcome is not affected by permutations of the input and, by extent, that all voters are treated equally. Monotone aggregators certify that if some voters convert their positions into the aggregator's result, then the outcome does not change. At first sight, this interpretation is stronger than what property 2 asserts, however it can be easily established using the contrapositive and induction. StrongDem aggregators were introduced by Szegedy and Xu [33]. The idea here is that there is a way to fix the votes of any $n-1$ voters so that the remaining voter has no influence on the outcome of the aggregation procedure. As Szegedy and Xu [33] argue, "what makes the notion of StrongDem particularly attractive is that when viewing its minimalistic definition, it seems a necessary condition for democracy, but it also has equivalent formulations, that are strong enough to accept it as a sufficient condition". Apart from this, Szegedy and Xu [20, 33] show that StrongDem aggregators have strong algebraic properties, as they relate to a property of functions called *strong resilience*.

It should be noted here that since Definition 22 refers to each component of an aggregator, we will also call a *Boolean operation* $f : \{0,1\}^m \longrightarrow \{0,1\}$ anonymous or monotone if it satisfies property 1 or 2 of this definition, respectively. Following Kun and Szegedy [20], we call Boolean functions that satisfy property 3 of Definition 22 1-*immune*.

It is straightforward to observe that a majority aggregator is anonymous, monotone, as well as StrongDem. To establish the latter, simply set all votes except the vote of voter i on the j^{th} issue to some $a \in \{0,1\}$. Then the outcome will be a, regardless the position the position of the i^{th} voter. Another immediate consequence of Definition 22, is that an anonymous or a StrongDem aggregator is non-dictatorial. On the other hand, every dictatorial aggregator is monotone, as projection functions are monotone.

For the convenience of the reader, in Table 3.1 we provide a concise illustration of the Boolean operations that have appeared in our characterization results, in comparison to the properties of Definition 22.

	anonymous	monotone	1-immune
$\vee$, $\wedge$, maj, $\vee^3$, $\wedge^3$	yes	yes	yes
$\oplus$	yes	no	no
pr_i^n	no	yes	no

TABLE 3.1: A classification regarding anonymity, monotonicity and 1-immunity

Since projections are neither anonymous nor 1-immune, we easily obtain the following result.

Corollary 3. *Any domain D admitting an anonymous or a StrongDem aggregator is a local possibility domain.*

Also, since all four binary Boolean operators are monotone, Theorem 7 directly implies that:

Corollary 4. *Any possibility domain D either admits a monotone non-dictatorial aggregator or an anonymous one.*

Proof. A possibility domain either admits a binary non-dictatorial aggregator, which is necessarily monotone, or is affine and thus admits the minority aggregator, which is anonymous. □

As one would expect, there exist non-dictatorial aggregators that are neither anonymous, nor monotone, nor StrongDem. For example, any ternary aggregator with at least one component being pr_1^3 and another being $\oplus$, has none of the aforementioned properties, as pr_1^3 is not anonymous, $\oplus$ is not monotone and neither of the two is 1-immune. What Corollary 4 implies is that a domain admitting such an aggregator, also admits another that is monotone or anonymous. From another point of view, provided that non-dictatorial aggregation is possible for a given domain, Corollary 4 guarantees the existence of a non-dictatorial aggregator that reflects at least one property of the majority aggregator, for that domain.

We proceed now with some examples that highlight the various connections between these types of aggregators. In Example 14, we present some specific formulas, where the domains they describe admit an aggregator that is at the same time anonymous, monotone and StrongDem. In Example 15, an aggregator that is anonymous, but neither monotone nor StrongDem and in Example 16, a non-dictatorial aggregator that is monotone, but neither anonymous nor StrongDem. Interestingly, as we will see in the sequel, there is no domain admitting a StrongDem aggregator and not an anonymous or monotone one. This not to say that every StrongDem aggregator is anonymous or monotone.

Example 14. *Any renamable Horn or bijunctive formula describes a domain admitting a symmetric or majority aggregator respectively. Such aggregators are anonymous, monotone and StrongDem.*

For a more complex example, consider the formula

$$\phi_{14} = (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3 \vee x_4),$$

whose set of satisfying assignments is the local possibility domain:

$$Mod(\phi_{14}) = \{0,1\}^4 \setminus \{(1,0,0,0),(1,0,1,0),(1,0,0,1),(1,0,0,0),(0,0,0,0)\}.$$

It is straightforward to check that $Mod(\phi_{14})$ admits the anonymous, monotone and StrongDem aggregator $(\wedge^{(3)}, \vee^{(3)}, maj, maj)$.

Example 15. *Consider the affine formula*

$$\phi_{15} = x_1 \oplus x_2 \oplus x_3,$$

whose set of satisfying truth assignments is:

$$Mod(\phi_{15}) = \{(0,0,1),(0,1,0),(1,0,0),(1,1,1)\}.$$

Clearly, $Mod(\phi_{15})$ admits the anonymous minority aggregator $\bar{\oplus} = (\oplus, \oplus, \oplus)$, which is neither monotone nor StrongDem.

In fact, using a combination of results by Dokow and Holzman [11, Example 3] and Kirousis et al. [16, Example 4.5], that involve the notion of total blockedness, it can be proven that $Mod(\phi_{15})$ does not admit any (non-dictatorial) monotone or StrongDem aggregators.

Example 16. *Recall the formula*

$$\phi_{11} = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3)$$

of Example 9, whose set of satisfying assignments is

$$Mod(\phi_{11}) = \{0,1\}^3 \setminus \{(1,0,0),(0,1,1)\}.$$

By checking all 4^3 different triples of binary supportive operators and since $Mod(\phi_{11})$ is not affine, by Theorem 7, one can see that $Mod(\phi_{11})$ is an impossibility domain.

Now, consider the separable formula

$$\phi_{16} := (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_4 \vee x_5 \vee x_6) \wedge (x_4 \vee \neg x_5 \vee \neg x_6),$$

whose set of satisfying assignments is the Cartesian product:

$$Mod(\phi_{16}) = Mod(\phi_{11}) \times Mod(\phi_{11}).$$

It easily follows that $Mod(\phi_{16})$ is a possibility domain admitting the monotone aggregator

$$(\mathrm{pr}_1^2, \mathrm{pr}_1^2, \mathrm{pr}_1^2, \mathrm{pr}_2^2, \mathrm{pr}_2^2, \mathrm{pr}_2^2).$$

Obviously, this aggregator is neither anonymous nor StrongDem.

Furthermore, since $Mod(\phi_{16})$ is not a local possibility domain, by Corollary 3, we can infer that it admits no anonymous, nor StrongDem aggregators.

We now provide examples of StrongDem aggregators that are either not anonymous or not monotone. Nevertheless, it holds that domains admitting such aggregators, also admit aggregators that are anonymous and monotone (see Theorem 16 below).

Example 17. *Let* $\bar{f} = (\underbrace{f, \ldots, f}_{m\text{-times}})$*, where* f *is a ternary operation defined as follows:*

$$f(0,0,0) = f(0,0,1) = f(0,1,1) = f(1,0,1) = 0,$$
$$f(0,1,0) = f(1,0,0) = f(1,1,0) = f(1,1,1) = 1.$$

We can easily establish that $\bar{f}$ *is StrongDem as, for each component of* $\bar{f}$*, it holds that:*

$$f(x,0,1) = f(0,x,1) = f(0,0,x) = 0,$$

for all $x \in \{0,1\}$*. On the other hand,* $\bar{f}$ *is neither anonymous nor monotone, since e.g.* $f(0,0,1) \neq f(0,1,0)$ *and* $f(0,1,0) = 1$*, whereas* $f(0,1,1) = 0$*.*

Now, consider $\bar{g} = (\underbrace{g, \ldots, g}_{m\text{-times}})$ *where* g *is a ternary operation defined as follows:*

$$g(0,0,0) = g(0,0,1) = g(0,1,0) = g(1,0,0) = g(1,1,0) = 0,$$
$$g(0,1,1) = g(1,0,1) = g(1,1,1) = 1.$$

Again, $\bar{g}$ *is StrongDem, as each component of* $\bar{g}$ *is 1-immune. Indeed, for all* $x \in \{0,1\}$ *it holds that:*

$$g(x,0,0) = g(0,x,0) = g(0,0,x) = 0.$$

It is straightforward to establish that $\bar{g}$ *is also monotone. On the other hand,* $\bar{g}$ *is not anonymous, since* $g(1,1,0) \neq g(0,1,1)$*.*

Finally, let $\bar{h} = (\underbrace{h, \ldots, h}_{m\text{-times}})$*, where* h *is a 4-ary operation defined as follows:*

$$h(x,y,z,w) = 1 \text{ if and only if exactly two or all of } x,y,z,w \text{ are equal to } 1.$$

Since the output of h *does not depend on the positions of the input bits,* h *is anonymous. Also,* h *is 1-immune, since:*

$$h(x,0,0,0) = h(0,y,0,0) = h(0,0,z,0) = h(0,0,0,w) = 0,$$

for all $x,y,z,w \in \{0,1\}$*. On the other hand,* h *is not monotone, since* $h(0,0,1,1) = 1$ *and* $h(0,1,1,1) = 0$*. Hence, the aggregator* $\bar{h}$ *is StrongDem and anonymous but not monotone.*

The only combination of properties from Definition 22 we have not seen, is an anonymous and monotone aggregator that is not StrongDem. The reason is that such aggregators do not exist, as the following lemma indicates.

Lemma 8. *Let* f *be an* n*-ary anonymous and monotone Boolean function. Then,* f *is also* 1*-immune.*

Proof. For $n = 2$, the only anonymous functions are $\wedge$ and $\vee$, which are also 1-immune.

Let $n \geq 3$. Since f is anonymous and monotone, it is not difficult to observe that there is some $l \in \{0, \ldots, n\}$, such that the output of f is 0 if and only if there are at most l 1's in the input bits. If $l > 0$ then for all $x \in \{0,1\}$:

$$f(x,0,0\ldots,0,0) = f(0,x,0,\ldots,0,0) = \cdots = f(0,0,0,\ldots,0,x) = 0.$$

If $l = 0$ then:

$$f(x,1,1\ldots,1,1) = f(1,x,1,\ldots,1,1) = \cdots = f(1,1,1,\ldots,1,x) = 1,$$

for all $x \in \{0,1\}$. In both cases, f is 1-immune. □

3.3.1 Characterizations for domains admitting anonymous, monotone and StrongDem aggregators

In the previous section, we studied three forms of non-dictatorial aggregators, with appealing properties. Since these properties are of great importance, if not necessity, for democratic voting schemes, we proceed with the syntactic characterization of domains that admit (i) anonymous, (ii) monotone and (iii) StrongDem aggregators. In particular, we show that domains admitting anonymous aggregators are described by local possibility integrity constraints, while domains admitting non-dictatorial monotone aggregators by separable or renamable partially Horn formulas and, lastly, that domains admitting StrongDem aggregators are described by a subclass of local possibility integrity constraints.

We begin with the case of anonymous aggregators. The starting point is, again, a result of Nehring and Puppe:

Theorem 14 (Nehring and Puppe [25], Theorem 2)**.** *A domain $D \subseteq \{0,1\}^m$ admits a monotone locally non-dictatorial aggregator if and only if it admits a monotone anonymous one.*

Kirousis et al. [16] strengthened this result by dropping the monotonicity requirement and fixing the arity of the anonymous aggregator, as a direct consequence of Theorem 11. We first need the following:

Definition 23. *We say that a ternary Boolean operator g is commutative if and only if for all $x, y \in \{0,1\}$, it holds that*

$$g(x,x,y) = g(x,y,x) = g(y,x,x).$$

It is not difficult to see that a ternary operator g is commutative if and only if $g \in \{\wedge^{(3)}, \vee^{(3)}, \text{maj}, \oplus\}$ (see e.g. [16, Lemma 5.7]). Another straightforward observation is that, for the case of ternary Boolean operations, the notions of anonymity and commutativity coincide.

Corollary 5 (Kirousis et al. [16], Corollary 5.11)**.** *D is a local possibility domain if and only if it admits a ternary anonymous aggregator.*

Proof. Theorem 11 asserts that, D is a local possibility domain if and only if D admits a ternary aggregator $(f_1, \ldots, f_m)$ where for each $j \in \{1, \ldots, m\}$, $f_j \in \{\wedge^3, \vee^3, maj, \oplus\}$. The result now follows by simply observing that the latter set coincides with the set of all ternary (unanimous) commutative Boolean operations. □

We obtain the following characterization result as an immediate consequence of Corollaries 3 and 5.

Corollary 6 (Dìaz et al. [9])**.** *D admits an n-ary anonymous aggregator if and only if there exists a local possibility integrity constraint whose set of models equals D.*

To deal with monotone and StrongDem aggregators we will need some preliminary work. The first fact we will use is that the set of aggregators of a domain D is *closed under superposition*.

Lemma 9 (Superposition of aggregators)**.** *Let $f = (f_1, \ldots, f_m)$ be an n-ary aggregator for D. If $g^1, \ldots, g^n$ are n l-ary aggregators for D, where $g^i = (g^i_1, \ldots, g^i_m)$, $i = 1, \ldots, n$, then $h := f(g^1, \ldots, g^n)$, where $h = (h_1, \ldots, h_m)$ and:*

$$h_j(x_1, \ldots, x_l) = f_j(g^1_j(x_1, \ldots, x_l), \ldots, g^n_j(x_1, \ldots, x_l)),$$

for all $j = 1, \ldots, m$ and for all $x_1, \ldots, x_l \in \{0,1\}$, is an l-ary aggregator for D.

Proof. This result is straightforward and can be found in [16, Lemma 5.6]. □

This fact allows us to press into service a result from the field of Universal Algebra, known as *Post's lattice*. Post's lattice denotes the lattice of all clones on a two-element set $\{0,1\}$, ordered by inclusion. A *clone* on a finite set A is a set $\mathcal{C}$ of finitary operations on A (i.e., functions from a finite power of A to A) such that C contains all projection functions and is closed under superpositions (see e.g. Szendrei [34]). Post [27] provided a complete classification of all clones of *Boolean* operations.

Here, we take advantage of the fact that the set of aggregators of a domain D is closed under superposition and (clearly) contains all dictatorial aggregators, to obtain the following result.

Lemma 10. *For a Boolean domain $D \subseteq \{0,1\}^m$, let, for all $j \in \{1,\dots,m\}$:*

$$\mathcal{C}_j := \{f \mid \text{There exists an aggregator } (f_1,\dots,f_m) \text{ for } D \text{ s.t. } f_j = f\},$$

be the set of the j-th components of every aggregator for D. Then, $\mathcal{C}_j$ is a clone.

This result has already been effectively used by Kirousis et al. [16] in order to obtain the characterizations of possibility and local possibility domains in the Boolean and non-Boolean framework. Here, we will use it in order to obtain the analogous results for domains that admit non-dictatorial monotone and StrongDem aggregators.

One of the main results of Post's classification that we will use is the following (see [3] for an easy to follow presentation):

Lemma 11. *Let $\mathcal{C}$ be a clone containing only unanimous operations. Then, either at least one of $\wedge, \vee, maj, \oplus$ is in $\mathcal{C}$, or $\mathcal{C}$ contains only projections.*

Finally, we will need two more definitions.

Definition 24. *We say that an n-ary Boolean operation f is an essentially unary function, if there exists a unary Boolean function g and an $i \in \{1,\dots,n\}$ such that:*

$$f(x_1,\dots,x_n) = g(x_i),$$

for all $x_1,\dots,x_n \in \{0,1\}$.

Note that the only unanimous such functions are the projections.

Definition 25. *We say that an n-ary Boolean operation f is linear, if there exist constants $c_0,\dots,c_n \in \{0,1\}$ such that:*

$$f(x_1,\dots,x_n) = c_0 \oplus c_1x_1 \oplus \cdots \oplus c_nx_n,$$

where $\oplus$ again denotes binary addition $\mod 2$.

It should be noted here, that an n-ary unanimous linear function with exactly one $c_i \neq 0$, is an essentially unary function, and by extent the projection operation pr_i^n.

Now, we proceed with some basic features of linear functions that we will need.

Lemma 12. *Let $f : \{0,1\}^n \mapsto \{0,1\}$ be a linear function and let $c_0, c_1,\dots,c_n \in \{0,1\}$ such that:*

$$f(x_1,\dots,x_n) = c_0 \oplus c_1x_1 \oplus \cdots \oplus c_nx_n.$$

Then f is unanimous if and only if $c_0 = 0$ and $c_i = 1$ for an odd number of indices $i \in \{1,\dots,n\}$.

Proof. The inverse direction is straightforward. For the forward direction, set $x_1 = \cdots = x_n = 0$. Then, $f(0,\ldots,0) = c_0$ and since f is unanimous, we have that $c_0 = 0$. Finally, assume, to obtain a contradiction, that there is an even number of $c_1 \ldots, c_n$ that are equal to 1. Set $x_1 = \cdots = x_n = 1$. Then, it holds that $f(1,\ldots,1) = 0$ and f is not unanimous. Contradiction. □

Since we work only with unanimous functions, from now on we will assume that a linear function satisfies the conditions of Lemma 12. This implies also that any linear function has odd arity.

Lemma 13. *Let $f : \{0,1\}^n \mapsto \{0,1\}$ be a linear function, $n \geq 3$. Then, either f is an essentially unary function or, it is neither monotone nor 1-immune.*

Proof. Let $c_1,\ldots,c_n \in \{0,1\}$ such that:

$$f(x_1,\ldots,x_n) = c_1x_1 \oplus \cdots c_nx_n,$$

and assume that f is not an essentially unary function. Then, there exist at least three pairwise different indices $i \in \{1,\ldots,n\}$ such that $c_i = 1$. If there are exactly three then $f = \oplus$, which is neither monotone nor 1-immune.

Now, assume that there are at least five pairwise different indices $i \in \{1,\ldots,n\}$ such that $c_i = 1$. We need only four of these indices. Let $i_1, i_2, i_3, i_4 \in \{1,\ldots,n\}$ such that $c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4} = 1$.

Set $x_{i_1} = x_{i_2} = x_{i_3} = 1$ and $x_i = 0$, for all $i \in \{1,\ldots,n\} \setminus \{i_1, i_2, i_3\}$. Then, $f(x_1,\ldots,x_n) = 1$. By letting $x_{i_4} = 1$ too, we obtain $f(x_1,\ldots,x_n) = 0$, which shows that f is not monotone.

Finally, aiming to a contradiction, suppose that f is 1-immune. Then, there exist $d_2,\ldots,d_n \in \{0,1\}$ such that:

$$\begin{aligned} f(0,d_2,\ldots,d_n) =& (1,d_2,\ldots,d_n) \Leftrightarrow \\ c_2d_2 \oplus \cdots \oplus c_nd_n =& c_1 \oplus c_2d_2 \oplus \cdots c_nd_n \Leftrightarrow \\ c_1 =& 0. \end{aligned}$$

Continuing in the same way, we can prove that $c_i = 0$, for $i = 1,\ldots,n$, which is a contradiction. □

We are now ready to prove a characterization result concerning non-dictatorial monotone aggregators.

Theorem 15 (Dìaz et al. [9]). *A domain $D \subseteq \{0,1\}^m$ admits a monotone non-dictatorial aggregator of some arity if and only if it admits a binary non-dictatorial one.*

Proof. That a domain admitting a binary non-dictatorial aggregator, admits also a non-dictatorial monotone one is obvious, since all binary unanimous functions are monotone.

For the forward direction, since D admits a monotone non-dictatorial aggregator, it is a possibility domain. Now, to obtain a contradiction, suppose D does not admit a binary non-dictatorial aggregator. Kirousis et al. [16, Lemma 3.4] showed that in this case, every n-ary aggregator, $n \geq 2$ for D is systematic[3].

Now, since D contains no binary non-dictatorial aggregators, $\wedge, \vee \notin \mathcal{C}_j$, for all $j \in \{1,\ldots,m\}$. Thus, by Lemma 11 either *maj* or $\oplus$ are contained in $\mathcal{C}_j$, for all

[3]The notion of *local monomorphicity* Kirousis et al. use in [16], corresponds to systematicity in the Boolean framework.

$j \in \{1, \ldots, m\}$ (since the aggregators must be systematic), lest each $\mathcal{C}_j$ contains only projections.

Assume that $\overline{maj}$ is an aggregator for D. Then, by Kirousis et al. [16, Theorem 3.7], D admits also a binary non-dictatorial aggregator. Contradiction.

Thus, we also have that $maj \notin \mathcal{C}_j$, $j = 1, \ldots, n$. It follows that only $\oplus \in \mathcal{C}_j$, $j = 1, \ldots, n$. By Post [27], it follows that for all $j \in \{1, \ldots, m\}$, $\mathcal{C}_j$ contains only linear functions (see also [3]). Since $\oplus$ is not monotone, the contradiction is now obtained by Lemma 13. □

Thus, by Proposition 3 and Theorem 9, we obtain the following syntactic characterization.

Corollary 7 (Dìaz et al. [9])**.** *D admits an n-ary non-dictatorial monotone aggregator if and only if there exists a separable or renamable partially Horn integrity constraint whose set of models equals D.*

To complete this subsection, we now turn to StrongDem aggregators. As we have already seen (Corollary 3), any domain that admits a StrongDem aggregator, is a local possibility domain, i.e. domains admitting a StrongDem aggregator constitute a subclass of local possibility domains. We have also established that a domain D is a local possibility domain if and only if it admits a ternary aggregator whose components are necessarily among the operations: $\wedge^3, \vee^3, maj, \oplus$. Of those four operations, only $\oplus$ is not 1-immune. The question that arises here, is whether we can obtain an analogous characterization for domains admitting StrongDem aggregators by simply excluding the appearance of $\oplus$. Dìaz et al. [9] proved that the answer is positive.

To obtain a proof of the above, we will use two more operators: the "diamond" operator $\diamond$ of Kirousis et al. [16], which is used to combine ternary aggregators in order to obtain new ones whose components are commutative functions (see also Bulatov's [4, Section 4.3] “Three Operations Lemma”); and the "star" operator $\star$ of Dìaz et al. [9], which under certain circumstances generates $\oplus$-free aggregators.

Definition 26. *Let $f = (f_1, \ldots, f_m)$ and $g = (g_1, \ldots, g_m)$ be two m-tuples of ternary functions. Define $h := f \diamond g$ to be the m-tuple of ternary functions $h = (h_1, \ldots, h_m)$, where:*

$$h_j(x, y, z) = f_j(g_j(x, y, z), g_j(y, z, x), g_j(z, x, y)),$$

for all $x, y, z \in \{0, 1\}$.

It is not hard to observe that if f and g are aggregators for a domain D, then so is h, since it is produced by a superposition of f and g. Moreover, it is easy to notice that h_j is commutative if and only if either f_j or g_j are. Thus, we obtain the following result.

Lemma 14 (Kirousis et al. [16], Lemma 5.10)**.** *Let $f = (f_1, \ldots, f_m)$ and $g = (g_1, \ldots, g_m)$ be two m-tuples of ternary functions and $f \diamond g := h = (h_1, \ldots, h_m)$. Then, for each $j \in \{1, \ldots, m\}$:*

$$h_j \in \{\wedge^{(3)}, \vee^{(3)}, maj, \oplus\} \text{ if and only if either } f_j \text{ or } g_j \in \{\wedge^{(3)}, \vee^{(3)}, maj, \oplus\}.$$

Furthermore, if g_j is commutative, then $h_j = g_j$, $j = 1, \ldots, m$.

Definition 27. *Let $f = (f_1, \ldots, f_m)$ and $g = (g_1, \ldots, g_m)$ be two m-tuples of ternary functions. Define $h := f \star g$ to be the m-tuple of ternary functions $h = (h_1, \ldots, h_m)$, where:*

$$h_j(x, y, z) = f_j(f_j(x, y, z), f_j(x, y, z), g_j(x, y, z)),$$

for all $x, y, z \in \{0, 1\}$.

Easily, if f and g are aggregators for a domain D, then so is h, since it is produced by a superposition of f and g. Also, if f_j is commutative, then so is h_j. Another useful feature of this operator is that, even in case g_j is commutative, it does not necessarily dominate $\star$ (in opposition to the $\diamond$ operator). However, the most interesting feature of $\star$ is that given some assumptions for f and g, h has no components equal to $\oplus$, as the following lemma indicates.

Lemma 15 (Dìaz et al. [9]). *Let $f = (f_1, \ldots, f_m)$ be an m-tuple of ternary functions, such that $f_j \in \{\wedge^{(3)}, \vee^{(3)}, maj, \oplus\}$, $j = 1, \ldots, m$, and let $J = \{j \mid f_j = \oplus\}$. Let also $g = (g_1, \ldots, g_m)$ be an m-tuple of ternary functions, such that $g_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\}$, for all $j \in J$. Then, for the m-tuple of ternary functions $f \star g := h = (h_1, \ldots, h_m)$, it holds that:*

$$h_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\},$$

for $j = 1, \ldots, m$.

Proof. First, let $j \in \{1, \ldots, m\} \setminus J$. Then, $f_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\}$ and let $x, y, z \in \{0, 1\}$ that are not all equal (lest we have nothing to show since all f_j, g_j are unanimous). If $f_j = \wedge^{(3)}$, then easily:

$$h_j(x,y,z) = \wedge^{(3)}(\wedge^{(3)}(x,y,z), \wedge^{(3)}(x,y,z), g_j(x,y,z)) = \wedge^{(3)}(0, 0, g_j(x,y,z)) = 0,$$

which shows that $h_j = \wedge^{(3)}$. Analogously, we show that $f_j = \vee^{(3)}$ implies that $h_j = \vee^{(3)}$. Finally, let $f_j = maj$ and let $maj(x,y,z) := z \in \{0,1\}$. Then:

$$h_j(x,y,z) = maj(maj(x,y,z), maj(x,y,z), g_j(x,y,z)) = maj(z, z, g_j(x,y,z)) = z,$$

which shows that $h_j = maj$.

Thus, we can now assume that $J \neq \emptyset$. Let $j \in J$. Then, we have that $f_j = \oplus$ and $g_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\}$. Thus, we have that:

$$h_j(x,y,z) = \oplus(\oplus(x,y,z), \oplus(x,y,z), g_j(x,y,z)) = g_j(x,y,z),$$

from which it follows that $h_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\}$. The proof is now complete. □

We now proceed with the characterization results of Dìaz et al., for domains admitting StrongDem aggregators.

Theorem 16 (Dìaz et al. [9]). *A Boolean domain $D \subseteq \{0,1\}^m$ admits an n-ary StrongDem aggregator if and only if it admits a ternary aggregator $f = (f_1, \ldots, f_m)$ such that $f_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\}$, for $j = 1, \ldots, m$.*

Proof. It is very easy to see that all the functions in $\{\wedge^{(3)}, \vee, maj\}$ are 1-immune. Thus, we only need to prove the forward direction of the theorem.

To that end, let $f = (f_1, \ldots, f_m)$ be an n-ary StrongDem aggregator for D. Then, by Theorem 11, there exists a ternary aggregator $g = (g_1, \ldots, g_m)$ such that $g_j \in \{\wedge^{(3)}, \vee^{(3)}, maj, \oplus\}$ for $j = 1, \ldots, m$. Let $J = \{j \mid f_j = \oplus\}$. If $J = \emptyset$, then we have nothing to prove. Otherwise, consider the clones $\mathcal{C}_j$, for each $j \in J$.

Suppose now that there exists a $j \in J$, such that $\mathcal{C}_j$ contains neither $\wedge$, nor $\vee$, nor maj. By Post's classification of clones of Boolean functions (see [3, 27]) and since $\mathcal{C}_j$ contains $\oplus$ and only unanimous functions, $\mathcal{C}_j$ contains only linear unanimous functions. By Lemma 13, $\mathcal{C}_j$ does not contain any 1-immune aggregator. Contradiction.

Thus, for each $j \in J$, it holds that $\mathcal{C}_j$ contains either $\wedge$, or $\vee$ or maj. In the first two cases, $\mathcal{C}_j$ obviously contains $\wedge^{(3)}$ or $\vee^{(3)}$ too respectively. Then, it holds that for each

$j \in J$ there exists an aggregator $h^j = (h_1^j, \ldots, h_n^j)$, such that $h_j^j \in \{\wedge^{(3)}, \vee^{(3)}, maj\}$. Let $J := \{j_1, \ldots, j_t\}$.

We will now perform a series of iterative combinations between g and the various h^j's, using the $\diamond$ and $\star$ operators, in order to obtain the required aggregator.

First, let $g^j = g \diamond h^j$, for all $j \in J$. Lemma 14 now implies that

$$g_i^j \in \{\wedge^{(3)}, \vee^{(3)}, maj, \oplus\},$$

for all $i \in \{1, \ldots, m\}$ and $j \in J$. Furthermore,

$$g_{j_s}^{j_s} \in \{\wedge^{(3)}, \vee^{(3)}, maj\},$$

for $s = 1, \ldots, t$. Thus for the aggregator:

$$G := (\cdots((g \star g^{j_1}) \star g^{j_2}) \star \cdots \star g^{j_t}),$$

we have, by Lemma 15:

$$G_j \in \{\wedge^{(3)}, \vee^{(3)}, maj\},$$

for $j = 1, \ldots, m$, which concludes the proof. □

Definition 28. *A local possibility integrity constraint is $\oplus$-free, if the set V_2 of Definition 20 is the empty set, $V_2 = \emptyset$.*

So, by Theorems 12 and 16, we obtain the following syntactic characterization of domains admitting StrongDem aggregators.

Corollary 8 (Dìaz et al. [9]). *A Boolean domain $D \subseteq \{0,1\}^m$ admits an n-ary StrongDem aggregator if and only if there exists an $\oplus$-free local possibility integrity constraint whose set of satisfying assignments equals D.*

3.4 Systematic Aggregators

In this section, we look into systematic aggregators. Recall that $(f_1, \ldots, f_m)$ is systematic if $f_1 = f_2 = \ldots = f_m := f$, and is denoted by $\bar{f}$, i.e., an aggregator is systematic when it aggregates the votes over each issue with a common rule. This property has appeared also as *(issue-)neutrality* in the bibliography (see e.g. Grandi and Endriss [13] and Nehring and Puppe [25]). From a Social Choice point of view, systematicity is a natural requirement, provided that the issues that need to be decided are of the same nature. In what follows, we syntactically characterize domains admitting (non-dictatorial) systematic aggregators as models of specific types of local possibility integrity constraints. Then, we present analogous characterizations for domains that are closed under the various kinds of non-dictatorial voting schemes we examined, under the additional requirement that these aggregators are also systematic. To this purpose, we employ known results and tools from the field of Universal Algebra.

Definition 29. *Let $D \subseteq \{0,1\}^m$ be a Boolean domain and $f : \{0,1\}^n \mapsto \{0,1\}$ a n-ary Boolean operation. We say that f is a polymorphism for D (or f preserves D, or D is closed under f) if, for all $x^1, \ldots, x^n \in D$:*

$$(f(x_1), \ldots, f(x_m)) \in D,$$

where $x^i = (x_1^i, \ldots, x_m^i)$ and $x_j = (x_j^1, \ldots, x_j^n)$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.

As is well known, the notion of polymorphisms plays an important role in Universal Algebra (see e.g. Szendrei [34]). In fact, polymorphisms were fundamentally used to obtain Post's classification results [27]. One cannot help but notice the clear correlation between the notions of a polymorphism and a systematic aggregator. The following lemma is straightforward, by considering the definitions of an aggregator and a polymorphism.

Lemma 16. *Let $D \subseteq \{0,1\}^m$ be a Boolean domain and $\bar{f} = (f,\dots,f)$ a systematic m-tuple of n-ary Boolean functions. Then $\bar{f}$ is an aggregator for D if and only if f is a polymorphism for D.*

In order to obtain the syntactic characterization of domains that admit (non-dictatorial) systematic aggregators, we use a result that can be directly acquired by Post's Lattice[4].

Corollary 9. *Let $D \subseteq \{0,1\}^m$ be a Boolean domain. Then, either D admits only essentially unary functions, or it is closed under $\wedge$, $\vee$, maj or $\oplus$.*

This directly implies that domains admitting non-dictatorial systematic aggregators are either Horn, dual-Horn, bijunctive or affine. Since the only unanimous essentially unary functions are projections, we immediately obtain the following characterization.

Corollary 10. *A Boolean domain $D \subseteq \{0,1\}^m$ admits an n-ary non-dictatorial systematic aggregator if and only if there exists an integrity constraint which is either Horn, dual Horn, bijunctive or affine, whose set of satisfying assignments equals D.*

Remark 7. *Note that the majority aggregator $\overline{maj}$ appears in the characterization of domains admitting systematic aggregators, whereas aggregators with even one component being maj do not appear in the characterization of possibility domains in general.*

More specifically, Theorem 7 asserts that a domain D is a possibility domain if and only if it admits the ternary (and systematic) minority aggregator $\bar{\oplus}$, or a binary non-dictatorial aggregator $(f_1,\dots,f_m)$, where $f_j \in \{\wedge,\vee,\mathrm{pr}_1^2,\mathrm{pr}_2^2\}$ for all $j \in \{1,\dots,m\}$. Thus, the appearance of Horn, dual-Horn and affine formulas in the syntactic characterization of domains admitting systematic aggregators is completely justifiable and expected, as $\bar{\wedge}$ and $\bar{\vee}$ are the only binary non-dictatorial and systematic aggregators, and D is affine if and only if it admits $\bar{\oplus}$.

The reason why bijunctive formulas appear in Corollary 10 and not in Theorem 7 is that, in the Boolean case, a domain admitting the aggregator $\overline{maj}$, also admits a binary aggregator $f = (f_1,\dots,f_m)$, such that $f_j \in \{\wedge,\vee\}$, $j = 1,\dots,m$. For the proof, see Kirousis et al. [16, Theorem 3.7]. The problem is that this aggregator need not be systematic. In fact, the proof of the aforementioned theorem would produce a systematic aggregator only if $(0,\dots,0)$ or $(1,\dots,1) \in D$.

Now we proceed with the characterizations for domains that admit the various non-dictatorial aggregators we examined in the previous sections, under the additional assumption that they also satisfy systematicity.

The case of domains that admit locally non-dictatorial or anonymous aggregators that are systematic is easy. Indeed, all the aggregators of Corollary 9 are locally non-dictatorial and anonymous. Thus the syntactic characterization of Corollary 10 applies for domains admitting systematic locally non-dictatorial or anonymous aggregators, as well.

[4] For a direct algebraic approach, see also Szendrei [34, Proposition 1.12] (by noting that the only Boolean *semi-projections* of arity at least 3 are projections).

As it comes to the class of domains admitting aggregators that are not generalized dictatorships, by Theorem 13, we have that it coincides with that of possibility domains with at least three elements. Thus, Corollary 10 works for domains admitting systematic aggregators that are not generalized dictatorships too.

In order to obtain a characterization for domains admitting monotone or StrongDem systematic aggregators, we will again use the terminology of polymorphisms, as well as Lemma 13 and Post's Lattice.

Corollary 11. *A domain $D \subseteq \{0,1\}^m$ admits an n-ary systematic non-dictatorial monotone or StrongDem aggregator if and only if it is closed under $\wedge$, $\vee$ or maj.*

Proof. It is known (and straightforward to see) that the set of polymorphisms of a domain is a clone. Let $\mathcal{C}$ be the Boolean clone of polymorphisms of D. Since it admits a non-dictatorial aggregator, at least one operator from $\wedge, \vee, maj, \oplus$ is in $\mathcal{C}$. By Lemma 13, this cannot be only $\oplus$. □

Thus, we obtain the following result, which completes this section.

Corollary 12. *A Boolean domain $D \subseteq \{0,1\}^m$ admits an n-ary systematic non-dictatorial monotone or StrongDem aggregator if and only if there exists an integrity constraint which is either Horn, dual Horn or bijunctive, whose set of satisfying assignments equals D.*

Chapter 4

Algorithmic identification results

In this chapter, we approach the problem of the syntactic characterization of possibility domains and local possibility domains from the perspective of computational complexity. The results we present are of two types. First we show how to recognize a (local) possibility integrity constraint efficiently, that is, in polynomial time to its length. Then we proceed to show that given a domain D we can efficiently decide whether it is a (local) possibility domain and in case it is, produce a (local) possibility integrity constraint ϕ, such that $D = \text{Mod}(\phi)$. These results have so far appeared only in the work of Dìaz et al. [9]. Then we turn to the different notions of non-dictatorial aggregators we discussed in the previous chapter and show that the corresponding domains can be efficiently identified us well.

4.1 Identification of possibility integrity constraints & possibility domains

In section 2.3 we saw that a subset $D \subseteq \{0,1\}^m$ is a possibility domain if and only if there exists a possibility integrity constraint ϕ such that $D = \text{Mod}(\phi)$. In this section, we show that given a formula ϕ, we can decide in time linear in the length of the formula whether it is a possibility integrity constraint. Then we use this -among other known results- to construct an algorithm that on input a domain D halts in time polynomial in the size of D and either decides that D is not a possibility domain or otherwise returns a possibility integrity constraint that describes D. The utility of this result becomes clear, in view of Arrow's impossibility theorem, since given a domain explicitly, as a listing of its elements, we can determine in time polynomial in its the size whether non-dictatorial aggregation is possible.

Possibility integrity constraints

Recall that, according to Definition 17, a formula is a possibility integrity constraint if it is either *separable* or *partially renamable Horn* or *affine*. Thus, in order to show that possibility integrity constraints are easily recognizable, it suffices to do that for each of the aforementioned syntactic types.

A similar classic result, is that the identification problem for renamable Horn formulas is solvable in linear time (see e.g. del Val [8] for a relatively recent such algorithm and Lewis [21] for the original non-linear one). Observe also that for Horn, dual-Horn and affine formulas, the same result also holds.

Henceforth, we assume that we have a set of variables $V := \{x_1, \ldots, x_m\}$ and a formula ϕ defined on V that is a conjunction of r clauses $C_1, \ldots, C_r$, where $C_j =$

$(l_{j_1}, \ldots, l_{j_{k_j}})$, $j = 1, \ldots, r$, and l_{j_s} is a positive or negative literal of x_{j_s}, $s = 1, \ldots, k_j$. We denote the set of variables corresponding to the literals of a clause C_j by $\text{vbl}(C_j)$.

The fact that separable formulas can be recognized in linear time is relatively straightforward:

Proposition 5 (Dìaz et al. [9])**.** *There is an algorithm that, on input a formula ϕ, halts in time linear in the length of ϕ and either returns that the formula is not separable, or alternatively produces a* partition *of V in two non-empty and disjoint subsets $V_1, V_2 \subseteq V$, such that no clause of ϕ contains variables from both V_1 and V_2.*

Proof. Suppose the variables of each clause of a formula are ordered by the indices of their corresponding literals in the clause. Thus, we say that x_{j_s}, x_{j_t} are consecutive in C_j, if $t = s + 1$, $s = 1, \ldots, k_j - 1$.

Now, given a formula ϕ, construct an undirected graph $G = (V, E)$, where:

- V is the set of variables of ϕ, and
- two vertices are connected if they appear consecutively in a common clause of ϕ.

It is easy to see that each clause C_j, where $\text{vbl}(C_j) = \{x_{j_1}, \ldots, x_{j_{k_j}}\}$ induces the path $\{x_{j_1}, \ldots, x_{j_{k_j}}\}$ in G.

The result is then obtained by showing that ϕ is separable if and only if G is not connected. To establish the latter, simply observe that two connected vertices of G cannot be separated in ϕ. Indeed, consider a path $P := \{x_s, \ldots, x_t\}$ in G (this need not be a path induced by a single clause). Then, each couple x_i, x_{i+1} of vertices in P belongs in a common clause of ϕ, $i = r, \ldots, s - 1$. Thus, ϕ is separable if and only if G is not connected. For the sake of completeness, we also provide the pseudocode of this algorithm (see Algorithm 1).

For the proof of linearity, notice that the set of edges can be constructed in linear time with respect to the length of ϕ, since we simply need to read once each clause of ϕ and connect its consecutive vertices. Also, there are standard techniques to check connectivity in linear time in the number of edges (e.g. by a *depth-first search* algorithm). □

Algorithm 1 SEPARABILITY

Input: $\phi(x_1, \ldots, x_m) = C_1 \wedge \cdots \wedge C_r$.
Create a graph $G = (V = \{x_1, \ldots, x_m\}, E := \emptyset)$.
for $j = 1, \ldots, r$ **do**
    Add to E all $\{x_{j_s}, x_{j_{s+1}}\}$ such that $x_{j_s}, x_{j_{s+1}} \in \text{vbl}(C_j)$.
end for
if G is connected **then**
    return `fail` and **exit**
else
    Let $A_1, \ldots, A_p \subseteq V$ be the connected components of G in some arbitrary ordering.
    $V_1 := \{x_s \mid x_s \in A_1\}$, $V_2 := V \setminus V_1$.
    return (V_1, V_2).
end if

For the case of renamable partially Horn formulas, things are far more complicated. As stepping stones, we will use Lewis' technique [21] of constructing, for a given formula ϕ, a 2-CNF formula which is satisfiable if and only if ϕ is renamable Horn. This very idea is how the identification problem for renamable Horn formulas was originally solved by Lewis [21], since 2SAT instances can be solved in polynomial time. To deal with renamable partially Horn formulas though, one has to

(i) look for a renaming that might transform only some clauses into Horn and

(ii) take into account the inadmissible Horn clauses, as they may cause other Horn clauses to become inadmissible too.

Let us first introduce some notation. Let ϕ be a renamable partially Horn formula defined over V. Assume that after renaming some of the variables in V, we get the partially Horn formula ϕ^*, with V_0 being the admissible set of variables. Let $\mathcal{C}_0$ be an admissible set of clauses for ϕ^*. By Remark 3, we assume below that only a subset $V^* \subseteq V_0$ has been renamed and that all Horn clauses of ϕ^* with variables exclusively from V_0 belong to $\mathcal{C}_0$. The clauses of ϕ^*, which are in a one to one correspondence with those of ϕ, are denoted by $C_1^*, \dots, C_r^*$, where C_j^* corresponds to C_j, $j = 1, \dots, r$. We also denote by V_1 the set $V_1 := V \setminus V_0$.

The following proposition relates the property of a formula ϕ being partially renamable Horn with the satisfiability of a formula ϕ'. It should be noted here that, we do not actually need to construct this formula in order to obtain the identification result for partially renamable Horn formulas. It is only introduced as an expedient towards Theorem 17.

Proposition 6 (Dìaz et al. [9]). *For every formula ϕ, there is a formula ϕ' such that ϕ is renamable partially Horn if and only if ϕ' is satisfiable.*

Proof. For each variable $x \in V$, we introduce a new variable x'. Intuitively, setting $x = 1$ means that x is renamed (and therefore $x \in V^*$), whereas setting $x' = 1$ means that x is in V_0, but is not renamed. Finally we set both x and x' equal to 0 in case x is not in V_0. Obviously, we should not not allow the assignment $x = x' = 1$ (a variable in V_0 cannot be renamed and not renamed). Let $V' = V \cup \{x' \mid x \in V\}$.

Consider the formula ϕ' below, defined over the variable set V'. For each clause C of ϕ and for each $x \in \mathrm{vbl}(C)$: if x appears positively in C, introduce the literals x and $\neg x'$ and if it appears negatively, the literals $\neg x$ and x'. ϕ' is the conjunction of the following clauses: for each clause C of ϕ and for each two variables $x, y \in \mathrm{vbl}(C)$, ϕ' contains the disjunctions of the positive with the negative literals introduced above. Thus:

(i) if C contains the literals x, y, then ϕ' contains the clauses $(x \vee \neg y')$ and $(\neg x' \vee y)$,

(ii) if C contains the literals $x, \neg y$, then ϕ' contains the clauses $(x \vee \neg y)$ and $(\neg x' \vee y')$

(iii) if C contains the literals $\neg x, y$, then ϕ' contains the clauses $(\neg x \vee y)$ and $(x' \vee \neg y')$ and

(iv) if C contains the literals $\neg x, \neg y$, then ϕ' contains the clauses $(\neg x \vee y')$ and $(x' \vee \neg y)$.

Finally, we add the following clauses to ϕ':

(v) $(\neg x_i \vee \neg x_i')$, $i = 1, \dots, m$ and

(vi) $\bigvee_{x \in V'} x$.

The clauses of items (i)–(v) correspond to the intuition we explained in the beginning. For example, consider the case where a clause C_j of ϕ has the literals $x, \neg y$. If we add x to V_0 without renaming it, we should not rename y, since we would have two positive literals in a clause of $\mathcal{C}_0$. Also, we should not add the latter to V_1, since we would have a variable of V_0 appearing positively in a clause containing a variable of V_1. Thus, we have that $x' \rightarrow y'$, which is expressed by the equivalent clause $(\neg x' \vee y')$ of item (ii). The clauses of item (v) exclude the assignment $x = x' = 1$ for any $x \in V$. Finally, since we want V_0 to be non-empty, we need at least one variable of V' to be set to 1.

To complete the proof of Proposition 6, we now proceed as follows.

($\Rightarrow$) First, suppose ϕ is renamable partially Horn. Let V_0, V_1, V^* and V' as above. Suppose also that $V_0 \neq \emptyset$.

Set $a = (a_1, \ldots, a_{2m})$ to be the following assignment of values to the variables of V':

$$a(x) = \begin{cases} 1, & \text{if } x \in V^*, \\ 0, & \text{else,} \end{cases} \quad \text{and } a(x') = \begin{cases} 0, & \text{if } x \in V^* \cup V_1, \\ 1, & \text{else,} \end{cases}$$

for all $x \in V$. To obtain a contradiction, suppose a does not satisfy ϕ'.

Obviously, the clauses of items (v) and (vi) above are satisfied, by the definition of a and the fact that V_0 is not empty.

Now, consider the remaining clauses of items (i)–(iv) above and suppose for example that some $(\neg x \vee y')$ is not satisfied. By the definition of ϕ', there exists a clause C which, before the renaming takes place, contains the literals $\neg x, \neg y$ (see item (iv)). Since the clause is not satisfied, $a(x) = 1$ and $a(y') = 0$, which in turn means that $x \in V^*$ and $y \in V^* \cup V_1$. If $y \in V_1$, C^* contains, after the renaming, a variable in V_1 and a positive appearance of a variable in V_0. If $y \in V^*$, C^* contains two positive literals of variables in V_0. Contradiction. The remaining cases can be proven analogously and are left to the reader.

($\Leftarrow$) Suppose now that $a = (a_1, \ldots, a_{2m})$ is an assignment of values to the variables of V' that satisfies ϕ'. We define the following subsets of V':

- $V^* = \{x \mid a(x) = 1\}$,
- $V_0 = \{x \mid a(x) = 1 \text{ or } a(x') = 1\}$ and
- $V_1 = \{x \mid a(x) = a(x') = 0\}$.

Let ϕ^* be the formula obtained by ϕ, after renaming the variables of V^*.

Obviously, V_0 is not empty, since a satisfies the clause of item (v).

Suppose that a clause C^*, containing only variables from V_0, is not Horn. Then, C^* contains two positive literals x, y. If $x, y \in V_0 \setminus V^*$, then neither variable was renamed and thus C also contains the literals x, y. This means that, by item (i) above, ϕ' contains the clauses $(x \vee \neg y')$ and $(\neg x' \vee y)$. Now, since $x, y \in V_0 \setminus V^*$, it holds that $a(x) = a_(y) = 0$ and $a(x') = a(y') = 1$. Then, a does not satisfy these two clauses. Contradiction. In the same way, we obtain contradictions in cases that at least one of x and y is in V^*.

Finally, suppose that there is a variable $x \in V_0$ that appears positively in a clause $C^* \notin \mathcal{C}_0$. Let $y \in V_1$ be a variable in C^* (there is at least one such variable, lest $C^* \in \mathcal{C}_0$). Suppose also that y appears positively in C^*.

Assume $x \in V^*$. Then, C contains the literals $\neg x, y$. Thus, by item (ii), ϕ' contains the clause $(\neg x \vee y)$. Furthermore, since $x \in V^*$, $a(x) = 1$ and since $y \in V_1$, $a(y) = 0$.

Thus the above clause is not satisfied. Contradiction. In the same way, we obtain contradictions in all the remaining cases. □

Observe that in order to compute ϕ' from ϕ, one would need quadratic time in the length of ϕ. To obtain a *linear time* algorithm which given a formula ϕ decides whether or not it is renamable partially Horn, we will work as follows: First, based on ϕ, we construct a *directed bipartite* graph G, i.e. a directed graph whose set of vertices is partitioned in two sets such that no vertices belonging in the same part are adjacent. Then, without computing ϕ', we show that ϕ' is satisfiable *if and only if* at least one of the *strongly connected components (scc)* of G, i.e. its maximal sets of vertices such that every two of them are connected by a directed path, is not *"bad"* (this term is specified bellow). The result is then acquired as an immediate consequence of Proposition 6.

Let us first introduce some auxiliary notation and terminology. For a directed graph G, we will denote a directed edge from a vertex u to a vertex v by (u, v). A (directed) path from u to v, containing the vertices $u = u_0, \ldots, u_s = v$, will be denoted by $(u, u_1, \ldots, u_{s-1}, v)$ and its existence by $u \to v$. If both $u \to v$ and $v \to u$ exist, we will sometimes write $u \leftrightarrow v$.

Note also that there are several algorithms in the literature that, given a directed graph $G = (V, E)$, can compute the scc of G in time $O(|V| + |E|)$, where $|V|$ denotes the number of vertices of G and $|E|$ that of its edges (see e.g. [35, Theorem 12]). By identifying the vertices of each scc, we obtain a *directed acyclic graph (DAG)*. An ordering $(u_1, \ldots, u_n)$ of the vertices of a graph is called *topological* if there are no edges (u_i, u_j) such that $i \geq j$, for all $i, j \in \{1, \ldots, n\}$.

Theorem 17 (Dìaz et al. [9])**.** *There is an algorithm that, on input a formula ϕ, halts in time linear in the length of ϕ and either returns that ϕ is not renamable partially Horn or alternatively produces a subset $V^* \subseteq V$ such that the formula ϕ^* obtained from ϕ by renaming the literals of variables in V^* is partially Horn.*

Proof. Given ϕ defined on V, whose set of clauses is $\mathcal{C}$ and let again $V' = V \cup \{x' \mid x \in V\}$. We define the graph G, with vertex set $V' \cup \mathcal{C}$ and edge set E such that, if $C \in \mathcal{C}$ and $x \in \text{vbl}(C)$, then:

- if x appears *negatively* in C, E contains (x, C) and (C, x'),
- if x appears *positively* in C, E contains (x', C) and (C, x) and
- E contains no other edges.

Intuitively, if $x, y \in V'$, then a path (x, C, y) corresponds to the clause $x \to y$ which is logically equivalent to $(\neg x \vee y)$. The intuition behind x and x' is exactly the same as in Proposition 6. We will thus show that the bipartite graph G defined above, contains all the necessary information to decide if ϕ' is satisfiable, with the difference that G can obviously be constructed in time linear in the length of the input formula.

There is a slight technicality arising here since, by the construction above, G always contains either the path (x, C, x') or (x', C, x), for any clause C and $x \in \text{vbl}(C)$, whereas neither $(\neg x \vee x')$ nor $(x \vee \neg x')$ are ever clauses of ϕ'. Thus, from now on, we will assume that no path can contain the vertices x, C and x' or x', C and x *consecutively*, for any clause C and $x \in \text{vbl}(C)$.

Observe that by construction,

(i) (x, C) or (C, x) is an edge of G if and only if $x \in \text{vbl}(C)$, $x \in V'$ and

(ii) (x, C) (resp. (x', C)) is an edge of G if and only if (C, x') (resp.(C, x)) is one too.

We now prove several claims concerning the structure of G. To make notation less cumbersome, assume that for an $x \in V$, $x'' = x$. Consider the formula ϕ' of Proposition 6.

Claim 9. *Let $x, y \in V'$. For $z_1, \dots, z_k \in V'$ and $C_1, \dots, C_{k+1} \in \mathcal{C}$, it holds that $(x, C_1, z_1, C_2, \dots, z_k, C_{k+1}, y)$ is a path of G if and only if $(\neg x \vee z_1)$, $(\neg z_i \vee z_{i+1})$, $i = 1, \dots, k-1$ and $(\neg z_k \vee y)$ are all clauses of ϕ'.*

Proof of Claim. Can be easily proved inductively to the length of the path, by recalling that a path (u, C, v) corresponds to the clause $(\neg u \vee v)$, for all $u, v \in V'$ and $C \in \mathcal{C}$. □

Claim 10. *Let $x, y \in V'$. If $x \to y$, then $y' \to x'$.*

Proof of Claim. Since $x \to y$, there exist $z_1, \dots, z_k \in V'$ and $C_1, \dots, C_{k+1} \in \mathcal{C}$, such that $(x, C_1, z_1, C_2, \dots, z_k, C_{k+1}, y)$ is a path of G. By Claim 9, $(\neg x \vee z_1)$, $(\neg z_i \vee z_{i+1})$, $i = 1, \dots, k-1$ and $(\neg z_k \vee y)$ are all clauses of ϕ'. By Proposition 6, so do $(\neg y' \vee z'_k)$, $(\neg z'_{i+1} \vee z'_i)$, $i = 1, \dots, k-1$ and $(\neg z'_1 \vee x')$ and the result is obtained using Claim 9 again. □

We can obtain the scc's of G using a variation of a *depth-first search (DFS)* algorithm, that, whenever it goes from a vertex x (resp. x') to a vertex C, it cannot then go to x' (resp. x) at the next step. Since the algorithm runs in time linear in the number of the vertices and the edges of G, it is also linear in the length of the input formula ϕ.

Let S be a scc of G. We say that S is *bad*, if, for some $x \in V$, S contains both x and x'. We can decide if each of the scc's is bad or not again in time linear in the length of the input formula.

Claim 11. *Let S be a bad scc of G and $y \in V'$ be a vertex of S. Then, y' is in S.*

Proof of Claim. Since S is bad, there exist two vertices x, x' of V' in S. If $x = y$ we have nothing to prove, so we assume that $x \neq y$. Then, we have that $y \to x$, which, by Claim 10 implies that $x' \to y'$. Since $x \to x'$, we get that $y \to y'$. That $y' \to y$ can be proven analogously. □.

Recall that by identifying the vertices of each sccof a graph, we can obtain a DAG. There are known algorithms in the literature for constructing a topological ordering of any DAG in *linear time*.

So, let $S_1, \dots . S_t$ be the scc's of G, in *reverse topological order*. We describe a process of assigning values to the variables of V':

1. Set every variable that appears in a bad scc of G to 0.
2. For each $j = 1, \dots, t$ assign value 1 to every variable of S_j that has not already received one (if S_j is bad no such variable exists). If some $x \in V'$ of S_j takes value 1, then assign value 0 to x'.
3. Let a be the resulting assignment to the variables of V'.

Now, the last claim we prove is the following:

Claim 12. *There is at least one variable $z \in V'$ that does not appear in a bad scc of G if and only if ϕ' is satisfiable.*

Proof of Claim. (⇒) We prove that every clause of type (i)–(vi) is satisfied. First, by the construction of a, every clause $\neg x_i \vee \neg x_i'$, $i = 1, \ldots, m$, of type (v) is obviously satisfied. Also, since by the hypothesis, z is not in a bad scc, it holds, by step 2 above, that either z or z' are set to 1. Thus, the clause $\bigvee_{x \in V'} x$ of type (vi) is also satisfied.

Now, suppose some clause $(x \vee \neg y')$ (type (i)) of ϕ' is not satisfied. Then $a(x) = 0$ and $a(y') = 1$. Furthermore, there is a vertex C such that (y', C) and (C, x) are edges of G. By the construction of G, (x', C) and (C, y) are also edges of G.

Since $a(x) = 0$, it must hold either that x is in a bad scc of G, or that $a(x') = 1$. In the former case, we have that $x \to x'$, which, together with (y', C, x) and (x', C, y) gives us that $y' \to y$. Contradiction, since then $a(y')$ should be 0. In the latter case, we have that there are two scc's S_p, S_r of G such that $x \in S_p$, $x' \in S_r$ and $p < r$ in their topological order. But then, there is some $q : p \leq q \leq r$ such that C in S_q. Now, if $p = q$, we obtain a contradiction due to the existence of (x', C), else, due to (C, x).

The proof for the rest of the clauses of types (i)–(iv) are left to the reader.

(⇐) First, recall that for two propositional formulas ϕ, ψ, we say that ϕ *logically entails* ψ, and write $\phi \models \psi$, if any assignment that satisfies ϕ, satisfies ψ too.

Now observe that, if x, y are two vertices in V' such that $x \to y$, then $\phi' \models (\neg x \vee y)$. Indeed, suppose β is an assignment of values that satisfies ϕ'. If $\beta(y) = 1$, we have nothing to prove. Thus, assume that $\beta(y) = 0$. By Claim 9, if $(x, C_1, z_1, C_2, z_2, \ldots, z_k, C_{k+1}, y)$ is the path $x \to y$, then $(\neg x \vee z_1)$, $(\neg z_i \vee z_{i+1})$, $i = 1, \ldots, k-1$ and $(\neg z_k \vee y)$ are all clauses of ϕ' and are thus satisfied by β. Since $\beta(y) = 0$, we have $\beta(z_k) = 0$. Continuing in this way, $\beta(z_i) = 0$, $i = 1, \ldots, k$ and thus $\beta(x) = 0$ too, which implies that $\beta(\neg x \vee y) = 1$.

Now, for the proof of the claim, suppose again that ϕ' is satisfiable, and let β be an assignment (possibly different than α) that satisfies ϕ'. Since β satisfies ϕ', it satisfies $\bigvee_{x \in V'} x$. This means that there exists some $x \in V'$ such that $\beta(x) = 1$. But β also satisfies $(\neg x \vee \neg x')$, so we get that $\beta(x') = 0$. Thus $\beta((\neg x \vee x')) = 0$, which means that ϕ' does not logically entail $\neg x \vee x'$. By the discussion above, there exists no path from x to x', so x is not in a bad scc of G. □

By Proposition 6, we have seen that ϕ is renamable partially Horn if and only if ϕ' is satisfiable. Also, in case ϕ' is satisfiable, a variable $x \in V$ is renamed if and only if $a(x) = 1$.

Thus, by the above and Claim 12, ϕ is renamable partially Horn if and only if there is some variable x that does not appear in a bad scc of G. Furthermore, the process described in order to obtain assignment a is linear in the length of the input formula, and a provides the information about which variables to rename. □

So far, we have established that the identification problem for separable and renamable partially Horn formulas is solvable in linear time. Also, it is obvious that the same holds for affine formulas as well. What should be noted here, is that the satisfiability problem remains NP-complete for the case of formulas that are separable or renamable partially Horn. Indeed, any formula ϕ in CNF can be extended to a separable or a renamable partially Horn $\phi \wedge \neg x$, where x is a new variable. Despite this, in Computational Social Choice, domains are considered to be non-empty as a non-degeneracy condition. Actually, it is usually assumed that the projection of a domain to any one of the m issues is the set $\{0, 1\}$.

From the above we obtain the following theorem, which states that checking whether a formula is a possibility integrity constraint can be done in polynomial time in the size of the formula.

Theorem 18 (Dìaz et al. [9]). *There is an algorithm that, on input a formula ϕ, halts in linear time in the length of ϕ and either returns that ϕ is not a possibility integrity constraint,* *or alternatively, (i) either it returns that ϕ is affine or (ii) in case ϕ is separable, it produces two non-empty and disjoint subsets $V_1, V_2 \subseteq V$ such that no clause of ϕ contains variables from both V_1 and V_2 and (iii) in case ϕ is renamable partially Horn, it produces a subset $V^* \subseteq V$ such that the formula ϕ^* obtained from ϕ by renaming the literals of variables in V^* is partially Horn.*

Remark 8. *Regarding the non-degeneracy assumptions, we ought to note here that the algorithms of Theorem 18 cannot distinguish formulas with non-degenerate sets of satisfying truth assignments from others with degenerate ones. An algorithm that could efficiently decide that, would effectively be (due e.g. to the syntactic form of separable formulas) an algorithm that could decide on input any given formula, which variables are satisfied by exactly one Boolean value and which admit both. It is quite plausible that no such efficient algorithm exists, as it could be used to solve known computationally hard problems, such like the* unique satisfiability *problem.*

Possibility domains

We proceed with the identification problem for possibility domains and show that given a domain $D \subseteq \{0,1\}^m$, we can efficiently decide whether or not D is a possibility domain. To this purpose, apart from Theorem 18, we substantially use Zanuttini and Hébrard's "unified framework" [36], that employs the notions of *prime implicants* and *prime formulas* to produce polynomial time algorithms for several structure identification problems.

Given a clause C of a formula ϕ, we say that a *sub-clause* of C is any non-empty clause created by deleting at least one literal of C.

Definition 30. *A clause C of a formula ϕ is a prime implicate of ϕ if no sub-clause of C is logically implied by ϕ. Furthermore, ϕ is prime if all its clauses are prime implicates of it.*

Given a non-empty subset $D \subseteq \{0,1\}^m$, Zanuttini and Hébrard [36, Proposition 5] provide an algorithm that produces a prime formula with $O(|D|m)$ clauses that describes D, in time $O(|D|^2m^2)$. They also provide an algorithm [36, Proposition 8], decides whether or not D is affine in time $O(|D|^2m^2)$ and in case it is produces an affine formula ϕ with $O(|D|m)$ clauses, that describes it.

The following proposition is the last necessary ingredient towards the identification of possibility domains, as well as the efficient construction of formulas that describe them.

Proposition 7 (Dìaz et al. [9]). *Let ϕ_P be a prime formula and ϕ be a formula logically equivalent to ϕ_P. Then:*

1. *if ϕ is separable, ϕ_P is also separable and*
2. *if ϕ is renamable partially Horn, ϕ_P is also renamable partially Horn.*

Proof. Let ϕ_P be a prime formula. Quine [28] showed that the prime implicates of ϕ_P can be obtained from any formula ϕ logically equivalent to ϕ_P, by repeated (i) resolution and (ii) omission of the clauses that have sub-clauses already created. Thus, using the procedures (i) and (ii) on ϕ, we can obtain every clause of ϕ_P.

If ϕ is separable, where $(V', V \setminus V')$ is the partition of its vertex set such that no clause contains variables from both V' and $V \setminus V'$, it is obvious that neither resolution or omission can create a clause that destroys that property. Thus, ϕ_P is separable.

Now, let ϕ be a renamable partially Horn formula where, by renaming the variables of $V^* \subseteq V$, we obtain the partially Horn formula ϕ^*, whose admissible set of variables is V_0. Let also ϕ^*_P be the formula obtained by renaming the variables of V^* in ϕ_P. Easily, ϕ^*_P is prime too.

Observe that the prime implicates of a partially Horn formula, are also partially Horn. Indeed, it is not difficult to observe that neither resolution, nor omission can cause a variable to seize being admissible: suppose $x \in V_0$. Then, the only way that it can appear in an inadmissible clause due to resolution is if there is an admissible Horn clause C containing $\neg x, y$, where $y \in V_0$ too and an inadmissible clause C' containing $\neg y$. But then, after using resolution, x appears negatively to the newly obtained clause. Thus, ϕ^*_P is partially Horn, which means that ϕ_P is renamable partially Horn. □

We are now ready to prove the following:

Theorem 19. *There is an algorithm that, on input $D \subseteq \{0,1\}^m$, halts in time $O(|D|^2 m^2)$ and either returns that D is not a possibility domain, or alternatively outputs a possibility integrity constraint ϕ, containing $O(|D|m)$ clauses, whose set of satisfying truth assignments is D.*

Proof. Given a domain D, we first use Zanuttini and Hébrard's algorithm to check if it is affine [36, Proposition 8], and if it is, produce, in time $O(|D|^2 m^2)$ an affine formula ϕ with $O(|D|m)$ clauses, such that $\text{Mod}(\phi) = D$. If it isn't, we use again Zanuttini and Hébrard's algorithm [36] to produce, in time $O(|D|^2 m^2)$, a prime formula ϕ with $O(|D|m)$ clauses, such that $\text{Mod}(\phi) = D$. Then, we use the linear algorithms of Proposition 5 and Theorem 17 to check if ϕ is separable or renamable partially Horn. If it is either of the two, then ϕ is a possibility integrity constraint and, by Theorem 10, D is a possibility domain. Else, by Proposition 7, D is not a possibility domain. □

4.2 Local possibility integrity constraints & Local possibility domains

In this section, we examine the identification problem for local possibility integrity constraints (lpic's) and local possibility domains (lpd's). For the former, we provide a linear time algorithm in the length of the formula and for the latter an algorithm polynomial in the size of the domain. In addition, we show that in case a domain D is an lpd we can efficiently construct an lpic that describes it.

Local possibility integrity constraints

As we have already mentioned (Corollary 1), every lpic is a possibility integrity constraint. Keeping this and Definition 20 in mind, we show that we can recognize lpic's efficiently.

Theorem 20 (Dìaz et al. [9]). *There is an algorithm that, on input a formula ϕ, halts in linear time in the length of ϕ and either returns that ϕ is not a local possibility constraint, or alternatively, produces the sets V_0, V_1, V_2 described in Definition 20.*

Proof. First, we check if ϕ is bijunctive or affine (this can be trivially done in linear time). If it is, then ϕ is an lpic. Else, we use the algorithm of Theorem 17 to obtain V_0.

Note that, by the construction of G and the way we obtain V_0, there is no variable in $V \setminus V_0$ that can belong in an admissible set.

If $V_0 = \emptyset$, then either ϕ is not an lpic, or there is a partition (V_1, V_2) of V such that no clause of ϕ contains variables from both V_1 and V_2. Thus, we use the algorithm of Proposition 5 to check if ϕ is separable. If it is not, then ϕ is not an lpic. If it is, we obtain two sub-formulas ϕ_1, ϕ_2 such that $\phi = \phi_1 \wedge \phi_2$. We can then trivially check, in linear time to their lengths, if ϕ_1 and ϕ_2 are bijunctive and affine respectively, or vice-versa. If they are, then ϕ is an lpic. Else, it is not.

Obviously, if $V_0 = V$, then ϕ is (renamable) Horn and thus an lpic. Now, suppose that $(V_0, V \setminus V_0)$ is a partition of V. Add all the variables of $V \setminus V_0$ that appear in an $(V, V \setminus V_0)$-generalized clause to V_2, and set $V_1 = V \setminus (V_0 \cup V_2)$. Now, if any clause of ϕ contains more that two variables from V_1, or variables from both V_1 and V_2, then ϕ is not an lpic. Else, it is. □

It should be noted that the issue about the non-degeneracy assumptions discussed in Remark 8 applies here too.

Local possibility domains

We end this section by showing that, given a domain D, we can efficiently determine whether it is an lpd and construct an lpic ϕ such that $\text{Mod}(\phi) = D$ in case it is.

Again, we employ results of Zanuttini and Hébrard's "unified framework". The first fact we use is that a prime formula that is logically equivalent to a bijunctive one, is also bijunctive [36, Proposition 3]. The second is that given a prime formula that describes an affine domain D we can construct, in linear time in the size of the input formula, an affine formula that describes D. In particular, for a clause $C = l_1 \vee \ldots \vee l_t$, where l_j are literals, $j = 1, \ldots, t$, let $E(C) := l_1 \oplus \ldots \oplus l_t$. For a CNF formula $\phi = \bigwedge_{j=1}^{m} C_j$, let $A(\phi) := \bigwedge_{j=1}^{m} E(C)$. In [36, Proposition 8] Zanuttini and Hébrard show that if ϕ is prime, $\text{Mod}(\phi) = D$ and D is affine, then $\text{Mod}(A(\phi)) = D$.

Theorem 21 (Dìaz et al. [9])**.** *There is an algorithm that, on input $D \subseteq \{0,1\}^m$, halts in time $O(|D|^2 m^2)$ and either returns that D is not a local possibility domain, or alternatively outputs a local possibility integrity constraint ϕ, containing $O(|D|m)$ clauses, whose set of satisfying truth assignments is D.*

Proof. Given a domain D, we first use Zanuttini and Hébrard's algorithm [36] to produce, in time $O(|D|^2 m^2)$, a prime formula ϕ with $O(|D|m)$ clauses, such that $\text{Mod}(\phi) = D$. Note that at this point, ϕ does not contain any generalized clauses. We then use the linear algorithm of Theorem 17 to check if ϕ is renamable partially Horn. If not, by Proposition 7, D is not an lpic. Otherwise, the algorithm produces a set V_0 such that ϕ is renamable partially Horn with admissible set V_0.

If $V_0 = V$ we have nothing to prove. Thus, suppose that $\phi = \phi_0 \wedge \phi_1$, where ϕ_0 contains only variables from V_0. Let ϕ_1' be the sub-formula of ϕ_1, obtained by deleting all variables of V_0 from ϕ. We use the algorithm of Proposition 5 to check if ϕ_1' is separable.

Suppose that ϕ_1' is not separable. We then check, with Zanuttini and Hébrard's algorithm, if ϕ_1' is either bijunctive or affine. If it is neither, then D is not an lpd. If it is bijunctive, then ϕ is a lpic. If it is affine, we construct the formula $A^*(\phi_1)$ as follows. For each clause $C = (l_1 \vee \cdots \vee l_s \vee (l_{s+1} \vee \cdots \vee l_t))$, where $l_1, \ldots, l_s$ are literals of variables in V_0, let:

$$E^*(C) = (l_1 \vee \cdots \vee l_s \vee (l_{s+1} \oplus \cdots \oplus l_t))$$

and $A^*(\phi_1) = \bigwedge_{j=1}^{m} E^*(C_j)$. Then, the lpic that describes D is $\phi_0 \wedge A^*(\phi_1)$.

In case ϕ_1' is separable, assume that $\phi_1' = \phi_2' \wedge \phi_3'$, where no variable appears in both ϕ_2' and ϕ_3'. Let also ϕ_2 be ϕ_2' with the variables from V_0 and respectively for ϕ_3. We now proceed exactly as with ϕ_1', but separately for ϕ_2' and ϕ_3'. If either one is neither bijunctive nor affine, D is not an lpd. Else, we produce the corresponding lpic as above. □

4.3 Domains closed under other forms of non-dictatorial aggregators

In Sections 3.2, 3.3 and 3.4 we examined several forms of non-dictatorial aggregators, namely aggregators that are not generalized dictatorships, anonymous, monotone, StrongDem and systematic aggregators. We also provided syntactic characterizations for the class of domains closed under each of these types of aggregation procedures. In this final section, we show that given a domain $D \subseteq \{0,1\}^m$ we can efficiently decide whether or not it belongs to one of these classes.

Aggregators that are not generalized dictatorships

By Corollary 2, we have that a domain $D \subseteq \{0,1\}^m$ with at least three elements, admits an aggregator that is not a generalized dictatorship if and only if there exists a possibility integrity constraint whose set of models equals D. Thus, given $D \subseteq \{0,1\}^m$ with at least three elements, using the algorithm of Theorem 19 we can determine in time $O(|D|^2m^2)$ whether or not D admits an aggregator that is not a generalized dictatorship, and produce a possibility integrity constraint with $O(|D|m)$ clauses that describes it, in case it is.

Remark 9. *Recall that in order to decide whether a given possibility integrity constraint really describes a domain that admits an aggregator that is not a generalized dictatorship, we additionally have to ensure that its set of models is comprised of at least three elements.*

It is easy to see that (non-degenerate) possibility domains with at least tree elements can only arise as the truth sets of possibility integrity constraints that are Horn, renamable Horn or affine. In all these cases, Creignou and Hébrard [6] have devised polynomial-delay *algorithms that generate all the solutions of such formulas, which can easily be implemented to terminate if they find more than two solutions.*

Anonymous aggregators

By Corollary 6, we have that D admits an n-ary anonymous aggregator if and only if there exists a local possibility integrity constraint whose set of models equals D. Therefore, given $D \subseteq \{0,1\}^m$, the algorithm of Theorem 4.3 provides, in time $O(|D|^2m^2)$, an answer to the identification problem for these domains, as well.

Non-dictatorial Monotone aggregators

Recall that by Corollary 6 a domain D admits an n-ary non-dictatorial monotone aggregator if and only if there exists a separable or renamable partially Horn integrity constraint whose set of models equals D.

Corollary 13. *There is an algorithm that, on input $D \subseteq \{0,1\}^m$, halts in time $O(|D|^2m^2)$ and either returns that D does not admit a non-dictatorial monotone aggregator, or alternatively outputs a separable or renamable partially Horn formula ϕ, containing $O(|D|m)$ clauses, whose set of satisfying truth assignments is D.*

Proof. Given a domain D, we use Zanuttini and Hébrard's algorithm [36] to produce, in time $O(|D|^2m^2)$, a prime formula ϕ with $O(|D|m)$ clauses, such that $\text{Mod}(\phi) = D$. Then, we use the linear algorithms of Proposition 5 and Theorem 17 to check if ϕ is separable or renamable partially Horn. If it is either of the two, D admits a non-dictatorial monotone aggregator. Otherwise, by Proposition 7, it does not. □

StrongDem aggregators

According to Corollary 8, a Boolean domain $D \subseteq \{0,1\}^m$ admits an n-ary StrongDem aggregator if and only if there exists an $\oplus$-free local possibility integrity constraint whose set of satisfying assignments equals D.

Corollary 14. *There is an algorithm that, on input $D \subseteq \{0,1\}^m$, halts in time $O(|D|^2m^2)$ and either returns that D does not admit a StrongDem aggregator, or alternatively outputs a $\oplus$-free local possibility integrity constraint ϕ, containing $O(|D|m)$ clauses, whose set of satisfying truth assignments is D.*

Proof. Given a domain D, we first use Zanuttini and Hébrard's algorithm [36] to produce, in time $O(|D|^2m^2)$, a prime formula ϕ with $O(|D|m)$ clauses, such that $\text{Mod}(\phi) = D$. Note that at this point, ϕ does not contain any generalized clauses. We then use the linear algorithm of Theorem 17 to check if ϕ is renamable partially Horn. If not, by Proposition 7, D is not an lpic. Otherwise, the algorithm produces a set V_0 such that ϕ is renamable partially Horn with admissible set V_0.

If $V_0 = V$ we have nothing to prove. Thus, suppose that $\phi = \phi_0 \wedge \phi_1$, where ϕ_0 contains only variables from V_0. Let ϕ_1' be the sub-formula of ϕ_1, obtained by deleting all variables of V_0 from ϕ.

We then check, with Zanuttini and Hébrard's algorithm, if ϕ_1' is bijunctive. If it is bijunctive, then ϕ is a $\oplus$-free lpic, thus D admits a StrongDem aggregator. Else, it does not. □

Non-dictatorial systematic aggregators

By Corollary 10, we have that a Boolean domain $D \subseteq \{0,1\}^m$ admits an n-ary non-dictatorial systematic aggregator if and only if there exists an integrity constraint which is either Horn, dual Horn, bijunctive or affine, whose set of satisfying assignments equals D.

Given $D \subseteq \{0,1\}^m$, we use use Zanuttini and Hébrard's algorithm [36] to produce, in time $O(|D|^2m^2)$, a prime formula ϕ with $O(|D|m)$ clauses, such that $\text{Mod}(\phi) = D$. To efficiently decide whether D admits a systematic non-dictatorial aggregator, we use two facts. First, that each the syntactic types above is known to be easily recognisable and second, that a prime formula logically equivalent to a Horn, dual Horn, bijunctive or affine is itself Horn, dual Horn, bijunctive or affine, respectively (again, see Zanuttini and Hébrard [36]).

Concluding remarks

The cornerstone of every decision making process is the urgency to aggregate various individual positions over a set of issues, into a single collective one. J.K. Arrow proved that, non-dictatorial aggregation is not always an option. In this work, we presented the classical frameworks used to formalize aggregation problems and focused on the integrity constraint based approach, where a domain of m issues is represented by a single Boolean formula of m variables, called the integrity constraint. We gave necessary and sufficient conditions, regarding the syntactic form of a formula, to describe a possibility domain, i.e. a domain where non-dictatorial aggregation is possible. We call these formulas possibility integrity constraints. Furthermore, we examined other forms of non-dictatorial aggregators that have appeared in the literature, and presented syntactic characterizations of the integrity constraints that describe the corresponding domains. In particular, we showed that domains that admit locally non-dictatorial aggregators, called local possibility domains, and domains admitting anonymous aggregators coincide and are described by local possibility integrity constraints. Domains admitting aggregators that are not generalized dictatorships are characterized as models of possibility integrity constraints, while domains that admit monotone non-dictatorial aggregators are described by a subclass of such formulas, the separable and renamable partially Horn formulas. Domains admitting StrongDem aggregators are described by a subclass of local possibility integrity constraints, called $\oplus$-free. Then, we discussed how these results are effected if we further require that these aggregators are systematic. Additionally, we showed that every integrity constraint of the above types, is easily (in linear time in the length of the input formula) recognisable. We also provided algorithms which, given a domain $D \subseteq \{0,1\}^m$ can determine in time polynomial in the size of the domain whether or not D admits a non-dictatorial aggregator of each of the types mentioned above, and in case it does, construct an integrity constraint that describes it, whose number of clauses is linear in the size of the domain. Apart from, knowing whether non-dictatorial aggregation is possible for a given domain, these algorithms also provide specific information on how to construct such an aggregator.

Bibliography

[1] Kenneth J Arrow. A difficulty in the concept of social welfare. *Journal of political economy*, 58(4):328–346, 1950.

[2] Duncan Black, Robert Albert Newing, Iain McLean, Alistair McMillan, and Burt L Monroe. The theory of committees and elections. 1958.

[3] Elmar Böhler, Nadia Creignou, Steffen Reith, and Heribert Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. In *SIGACT News*. Citeseer, 2003.

[4] Andrei A Bulatov. Conservative constraint satisfaction re-revisited. *Journal of Computer and System Sciences*, 82(2):347–356, 2016.

[5] Fabrizio Cariani, Marc Pauly, and Josh Snyder. Decision framing in judgment aggregation. *Synthese*, 163(1):1–24, 2008.

[6] Nadia Creignou and J-J Hébrard. On generating all solutions of generalized satisfiability problems. *RAIRO-Theoretical Informatics and Applications*, 31(6):499–511, 1997.

[7] Rina Dechter and Judea Pearl. Structure identification in relational data. *Artificial Intelligence*, 58(1-3):237–270, 1992.

[8] Alvaro del Val. On 2-sat and renamable Horn. In *Proceedings of the National Conference on Artificial Intelligence*, pages 279–284. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000.

[9] Josep Díaz, Lefteris Kirousis, Sofia Kokonezi, and John Livieratos. Algorithmically efficient syntactic characterization of possibility domains. *arXiv preprint arXiv:1901.00138*, 2019.

[10] Elad Dokow and Ron Holzman. Aggregation of binary evaluations for truth-functional agendas. *Social Choice and Welfare*, 32(2):221–241, 2009.

[11] Elad Dokow and Ron Holzman. Aggregation of binary evaluations. *Journal of Economic Theory*, 145:495–511, 2010.

[12] Herbert Enderton and Herbert B Enderton. *A mathematical introduction to logic*. Elsevier, 2001.

[13] Umberto Grandi and Ulle Endriss. Lifting integrity constraints in binary aggregation. *Artificial Intelligence*, 199:45–66, 2013.

[14] Umberto Grandi, Ulle Endriss, et al. Lifting rationality assumptions in binary aggregation. In *AAAI*, 2010.

[15] Davide Grossi and Gabriella Pigozzi. Judgment aggregation: a primer. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(2):1–151, 2014.

[16] Lefteris Kirousis, Phokion G Kolaitis, and John Livieratos. Aggregation of votes with multiple positions on each issue. In *Proceedings 16th International Conference on Relational and Algebraic Methods in Computer Science*, pages 209–225. Springer, 2017. Expanded version to appear in *ACM Transactions on Economics and Computation*.

[17] Lefteris Kirousis, Phokion G Kolaitis, and John Livieratos. On the computational complexity of non-dictatorial aggregation. In *International Conference on Relational and Algebraic Methods in Computer Science*, pages 350–365. Springer, 2018.

[18] Lefteris Kirousis, Phokion G Kolaitis, and John Livieratos. Aggregation of votes with multiple positions on each issue. *ACM Transactions on Economics and Computation (TEAC)*, 7(1):1, 2019.

[19] Lewis A Kornhauser and Lawrence G Sager. The one and the many: Adjudication in collegial courts. *Cal. L. Rev.*, 81:1, 1993.

[20] Gábor Kun and Mario Szegedy. A new line of attack on the dichotomy conjecture. *European Journal of Combinatorics*, 52:338–367, 2016.

[21] Harry R Lewis. Renaming a set of clauses as a Horn set. *Journal of the ACM (JACM)*, 25(1):134–135, 1978.

[22] Christian List and Philip Pettit. Aggregating sets of judgments: Two impossibility results compared. *Synthese*, 140(1-2):207–235, 2004.

[23] Christian List and Clemens Puppe. Judgment aggregation: A survey. In Christian List and Clemens Puppe, editors, *Handbook of Rational and Social Choice*. Oxford University Press, 2009.

[24] Philippe Mongin. Judgment aggregation. *Cahier DRI*, page 02, 2011.

[25] Klaus Nehring and Clemens Puppe. Abstract arrowian aggregation. *Journal of Economic Theory*, 145(2):467–494, 2010.

[26] Philip Pettit. Deliberative democracy and the discursive dilemma. *Noûs*, 35:268–299, 2001.

[27] Emil Leon Post. *The two-valued iterative systems of mathematical logic*. Number 5 in Annals of Mathematics Studies. Princeton University Press, 1941.

[28] Willard V Quine. On cores and prime implicants of truth functions. *The American Mathematical Monthly*, 66(9):755–760, 1959.

[29] Lionel Robbins. Interpersonal comparisons of utility: A comment. *The Economic Journal*, 48(192):635–641, 1938.

[30] Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226. ACM, 1978.

[31] Amartya Sen. Chapter 22 social choice theory. volume 3 of *Handbook of Mathematical Economics*, pages 1073 – 1181. Elsevier, 1986.

[32] Amartya Sen. The possibility of social choice. *American Economic Review*, 89(3):349–378, 1999.

[33] Mario Szegedy and Yixin Xu. Impossibility theorems and the universal algebraic toolkit. *arXiv preprint arXiv:1506.01315*, 2015.

[34] Ágnes Szendrei. *Clones in universal algebra*, volume 99. Presses de l'Université de Montréal, 1986.

[35] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[36] Bruno Zanuttini and Jean-Jacques Hébrard. A unified framework for structure identification. *Information Processing Letters*, 81(6):335–339, 2002.